Doctoral Thesis: Robust Video Trackers against In-capture and Post-capture Distortions using Video Quality Assessment based on Natural Scene Statistics and Deep Learning.

Roger Gomez Nieto, M.Sc

Advisor: Hernan Dario Benitez Restrepo, Ph.D

Co-Advisor: Alan Bovik, Ph.D.

Pontificia Universidad Javeriana Seccional Cali
School of Engineering and Sciences - Doctoral Program in Engineering and Applied Sciences
Cali, Colombia
2021-II

Content

List of Figures

List of Tables

Abstract

The current work in Video Object Tracking (VOT) has studied various image factors that affect VOT algorithm performance. For instance, factors such as occlusion, clutter, confusion, object shape, varying speed, and zooming, which influence video quality, affect tracking performance. Nonetheless, there is no clear distinction between scene-dependent challenges such as occlusion and clutter and the challenges imposed by traditional notions of “quality impairments” inherited from capture, compression, processing, and transmission. 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.

VOT is a challenging task because the videos analyzed can be distorted by impairments such as bounding box initialization error, sensor noise, latency by video transmission, illumination changes, and data loss caused by compression algorithms. An exciting direction of research is the interaction between visual quality and tracking task. Taking into account that Surveillance videos are fraught with numerous sources of distortions, including blur, noise, and artifacts arising from processes such as compression, scaling, and format conversion. Multiple interacting distortions are often present, which significantly complicates the tracking task. Although effective tracking algorithms have been implemented, the study of their performance with respect to a wide variety of generally nonlinear, often commingled, poorly understood distortions, in practice, is a complex problem. Despite numerous algorithms proposed in the past few decades, video tracking methods that explicitly include strategies to make them robust to in-capture and post-capture video distortions have not been widely explored. Hence, there is a lack of data sets designed for object tracking that present in-capture distortion and systematic evaluation of state-of-the-art video tracking algorithms under in-capture and post-capture video distortions.

Because a similar resource was not already available, we created an Authentically Distorted Surveillance Videos Dataset (AD-SVD) acquired by four different surveillance cameras 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, including a variety of activities. Furthermore, we provide benchmarking results for evaluating 11 state-of-the-art visual object trackers (from VOT 2017-2018 challenges, among others) based

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

on the proposed dataset. To the best of our knowledge, AD-SVD is the largest, densely annotated, and authentically distorted video object tracking benchmark for STT. Because the configuration of the distortion levels in the four video cameras is not identical, we used the reliable No-Reference (NR) VQA metric V-BLIINDS to test the consistency of the parameter settings of the cameras used to record the AD-SVD videos. We observe that perceptual quality decreases (V-BLIINDS scores increase) in the following order: pristine, exposure, defocus, and combined exposure-defocus distortions, as expected. Videos affected by exposure distortions exhibit considerable variability in the V-BLIINDS scores for AD-SVD. By contrast, V-BLIINDS scores of videos affected by defocus and commingled defocus and exposure impairments showed minor variance.

Moreover, we developed an approach for performance prediction and quality-aware feature selection for single-object tracking in authentically distorted surveillance videos. We designed a model-agnostic (independent of the tracker model) framework that predicts performance without running the corresponding tracking algorithm to facilitate comparisons across different approaches. To this end, we learn a mapping between the input video and the area under the curve (AUC) of the success plot. This process is carried out in two stages, i) extraction of a fixed-size set of features and ii) AUC estimation using a support vector machine regression model. The proposed method predicts the performance of a VOT algorithm with high accuracy in such a way that the probability of obtaining the reference output is maximized without executing the tracking algorithms. With a high level of accuracy, this framework predicts the performance of the VOT algorithm on several outdoor and indoor locations, different visual contents, and under diverse types and levels of authentic distortion.

We also propose a framework to reduce the video tracker computation resources (time and disk space required for storage). We achieve this by predicting the VOT performance on authentically distorted surveillance videos to determine the optimal frame resolution scale for processing the video, which reduces the execution time of the video tracker and preserves its performance. We obtained the best results on scales $1/4$ and $1/10$ because of the requirements imposed by the video tracker algorithms on the input video resolution. A scale smaller than $1/4$ or $1/10$, depending on the tracker, does not imply a more significant 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 with lower scales. This approach can reduce the execution time by up to 34% with a slight decrease in performance of 3%.

In addition, we conducted experiments to explore the effects of reducing the number of frames processed by the 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, without significantly decreasing the performance. Further studies to determine the usefulness of frame reduction in VOT might validate these encouraging results by incorporating more tracking algorithms and data scenes.

This research is concerned with the interpretation of video quality, as it affects video tracking performance and how it affects the performance of trackers in real surveillance videos. Hence, we also present a novel method for No-Reference VQA. This framework is fast and does not require the extraction of handcrafted features. We extracted convolutional features of the 3-D C3D Convolutional Neural Network and trained a Support Vector Regressor to obtain a VQA score. We performed transformations into different color spaces to generate better discriminant deep features. We extracted features from several layers and found the best configuration to improve the VQA score with and without overlap. We tested the proposed approach using the LIVE-Qualcomm dataset. We evaluated the perceptual quality prediction model extensively, obtaining a final Pearson correlation of 0.7749 ± 0.0884 with Mean Opinion Scores. We also showed that it could achieve good video quality prediction, outperforming other state-of-the-art VQA leading models.

Overall, limited research has been conducted in the area of video quality applied to VOT tasks. This thesis deepens the understanding of particular topics and issues related to the interaction between perceptual and machine vision quality assessment by providing an experimental perspective. For example, VOT investigations have predominately been done by scientists who do not necessarily account for the effects of video quality attributes in the task. The same applies to computer vision scientists who produce No-Reference video quality systems based on subjective human scores and do not consider specific task performance. For instance, most NR-VQA datasets created over the years from the computer vision community do not contain in-capture or post-capture distortions in videos with typical video surveillance activities. Similarly, standard distorted datasets are created without controlled level variations of such distortions. Hence, this thesis advances further research on the performance of VOT algorithms concerning in-capture distortions.

Keywords

Video Quality Assessment, Visual Object Tracking, Convolutional Neural Network, Video Tracking Performance Prediction, In-capture distortions, Robust Object Tracking, Authentically Distorted Video Dataset.

Resumen

El trabajo investigativo realizado hasta el momento en Seguimiento de Objetos en Video (VOT) ha estudiado diversos factores de la imagen que afectan el rendimiento de VOT. Por ejemplo, factores como oclusión, aglomeración de objetos, confusión, la forma del objeto, velocidad variable, acercamiento, entre otros, influencian la calidad del video y afectan la precisión del seguidor. Sin embargo, hasta el momento, no se ha definido una distinción clara entre los desafíos originados por la escena, tales como oclusión y aglomeración de objetos, con los desafíos impuestos directamente por la calidad del video. Estas distorsiones que afectan la calidad del video pueden generarse por etapas o fases presentes en la captura, compresión, procesamiento y transmisión del video. A pesar de la abundancia de métodos VOT en la literatura, aún se presenta una ausencia de estudios detallados que analicen el rendimiento de los VOT en videos que contengan distorsiones en captura y post-captura.

El seguimiento de objetos en video es una tarea desafiante debido a la necesidad de trabajar con videos que tienen múltiples imperfecciones y distorsiones. Entre estas se encuentran rectangulos de inicialización del objeto mal ubicados, ruido en el sensor, latencia por transmisión de video, cambios de iluminación, y pérdida de datos por algoritmos de compresión. Un importante y actual campo de investigación es la interacción entre la calidad de video y el desempeño en la tarea. Esto al tener en cuenta que los videos usados en video-vigilancia están plagados con numerosas fuentes de distorsión, incluyendo borrosidad, ruido y artefactos que surgen de procesos como compresión, escalamiento, conversión de formato, entre otros. A menudo en un mismo video se encuentran múltiples distorsiones, las cuales interactúan, lo cual complica significativamente la tarea del seguidor de objetos. Aunque en el estado del arte se proponen numerosos algoritmos seguidores de objeto cada año, hacerlos robustos contra la amplia variedad de distorsiones no lineales, a menudo contenidas de forma simultánea, y además, poco entendidas, es un problema altamente complejo. A pesar de la buena precisión de los algoritmos seguidores recientes, estos no han demostrado ser lo suficientemente robustos a distorsiones de video en captura y postcaptura. Algo que no ha permitido el avance en la mejora de dicha robustez, es la ausencia de bases de datos de videos que presenten distorsiones en captura. Similarmente, no se reporta una evaluación sistemática de los seguidores del estado del arte en videos que adquieran distorsiones durante la captura y postcaptura.

Debido a la ausencia de un recurso similar, creamos una base de datos de videos auténticamente distorsionas, a la que llamamos AD-SVD. Esta base de datos está compuesta por videos adquiridos por cuatro cámaras de vigilancia especiales, las cuales permitían tener un control fino de los parámetros de la captura, con el fin de generar las distorsiones auténticas y sus tres niveles de intensidad. Hacemos público para la comunidad científica y para fines de investigación esta base de datos, en IEEE DataPort. AD-SVD contiene 4476 videos, grabados en 3 escenarios a la intemperie, y 4 escenarios en entornos cerrados, incluyendo una variedad de actividades de interés para video-vigilancia. Adicional a la base de datos, realizamos un estudio comparativo para once seguidores de objetos del estado-del arte (algunos de los retos VOT del 2017 y 2018), evaluados en los videos de AD-SVD. A la extensión de nuestro conocimiento, AD-SVD es la base de datos de videos auténticamente distorsionados más grande, densamente etiquetada (todos los cuadros/imágenes del video tienen etiquetas de la posición real del objeto), pensada para seguimiento de objetos en videos cortos, en la actualidad. Debido a que la configuración de los parámetros de captura de las cámaras usadas no es uniforme, empleamos la métrica V-BLIINDS para evaluar la calidad de los mismos. Con esta evaluación de calidad, podemos encontrar conjuntos de parámetros que generen niveles de distorsiones auténticas en las cuatro cámaras. Este estudio concluyó que la calidad perceptual de los videos presenta el siguiente orden, de mayor a menor (a menor calidad, el puntaje V-BLIINDS es mayor): original, distorsionado por tiempo de exposición (exposición), pérdida de foco, y pérdida de foco simultáneamente con exposición. Estos resultados de alguna manera fueron los esperados. Sin embargo, también se encontró que los videos afectados por exposición exhiben una variabilidad mayor en los puntajes V-BLIINDS. Por el contrario, los videos afectados por distorsión por pérdida de foco y foco unido con exposición, presentan una variabilidad de puntaje menor.

Desarrollamos también una aproximación para la predicción del rendimiento y la selección de características de calidad, para seguimiento de objetos en videos de vigilancia auténticamente distorsionados. Diseñamos un modelo que es independiente del tracker en el que se usará. Este modelo predice el rendimiento sin necesidad de ejecutar el seguidor como tal en el video. Esto permite predecir el rendimiento en diversos seguidores, y realizar comparaciones del rendimiento, sin necesidad de ejecutar el seguidor en todo el video. Para hacer esto, diseñamos una correspondencia entre el video de entrada y el área bajo la curva (AUC) del succes plot. Este proceso se realiza en dos etapas: i) extracción de un conjunto de características, con tamaño fijo, ii) estimación del AUC usando un modelo de máquinas de soporte para regresión.

El método propuesto logra una alta precisión en la predicción del rendimiento de un algoritmo VOT. De esta manera, se maximiza la probabilidad de obtener la salida de referencia sin tener que ejecutar el algoritmo seguidor como tal. Este método demostró una alta precisión cuando se ejecuta en videos tanto de escenas al exterior como en ambientes interiores, contenidos visuales diversos y diferentes niveles de distorsiones auténticas.

De igual manera, proponemos una metodología para reducir los recursos de cómputo necesarios para ejecutar un algoritmo seguidor (tiempo computaciones y uso de disco para almacenamiento). Para esto, predecimos el rendimiento del seguidor en videos auténticamente distorsionados, con el fin de determinar la resolución espacial óptima para procesar el video, de tal manera que se reduzca el tiempo de ejecución pero no su precisión. Los mejores resultados se obtuvieron con las escalas $1/4$ y $1/10$. Las escalas más pequeñas no necesariamente mejoran el tiempo de ejecución, pero si degradan de manera considerable el rendimiento del seguidor. Los resultados finales demuestran que la aproximación propuesta permite disminuir hasta 34% el tiempo de ejecución, con una disminución en el rendimiento de tan solo 3%.

Adicionalmente, realizamos experimentos con el fin de explorar los efectos de reducir los cuadros de un video en el rendimiento de un algoritmo seguidor. La metodología propuesta permite preservar el rendimiento del algoritmo, mientras se reduce la tasa de cuadros por segundo (FPS). Hicimos reducciones de la FPS en un nivel que llegó hasta un orden de magnitud menor que los cuadros originales del video, sin reducir notablemente el rendimiento del algoritmo. Para estos experimentos, usamos un conjunto de 1000 videos extraídos de AD-SVD, los cuales contienen diversos contenidos de escena y auténticas distorsiones. Pese a estos buenos resultados obtenidos, se propone como trabajo futuro replicar estos resultados con más algoritmos de seguimientos y diversidad de video de prueba.

Esta investigación se centra en la Evaluación de Calidad de Video, y en como esta afecta el rendimiento de algoritmos de seguimiento de objetos, cuando se tienen como entrada videos de video-vigilancia reales. Para esto, proponemos también un nuevo método No Referenciado de Evaluación de Calidad de Video (NR VQA). Este método es rápido y no requiere la extracción de características predeterminadas. Para hacerlo, extraemos las características de una de las capas más profundas de una red neuronal convolucional 3-D. Con estas características, entrenamos un regresor de soporte vectorial para obtener un puntaje de calidad de video. Antes de ingresar a la CNN, le aplicamos una transformación del espacio de color al video, para incrementar el poder de discriminación de las características perceptuales

profundas. Experimentamos extrayendo las características de las diferentes capas que tiene la 3-D CNN, y con diferentes valores de traslape entre cuadros del video, encontrando la configuración que brinda la mejor correlación con los puntajes de calidad humanos. Para validar nuestros resultados, probamos también en la base de datos LIVE-QUALCOMM, obteniendo un coeficiente de Correlación de Pearson con los puntajes de calidad subjetivos de 0.7749 ± 0.0884. De esta manera, demostramos que el método NR-VQA propuesto presenta una buena capacidad de predicción de VQA, superando incluso algunos métodos recientes del estado del arte.

Se concluye que en el estado del arte existe escasa información en el área de calidad de video aplicada a tareas de seguimiento de objetos. Esta tesis pretende ampliar el conocimiento que se tiene en este tema en particular, desde una perspectiva experimental. A modo de ejemplo, la mayoría de investigaciones y propuestas de algoritmos seguidores de objetos no tienen en cuenta la calidad de los videos, ni el impacto que dicha calidad tiene en la precisión del seguidor aplicado en videos reales. El mismo dilema aplica a los investigadores que desarrollan algoritmos NR VQA, principalmente centrados en lograr predicción basada en puntajes de calidad dado por humanos, y no centrados en el rendimiento de las diversas tareas de visión por computador. Como muestra de ello, recientemente no se han propuesto bases de datos de videos que contengan distorsiones auténticas y postcaptura, y que al mismo tiempo contengan acciones y escenas de interés para video-vigilancia y seguimiento de objetos. De manera similar, las bases de datos de videos que contienen distorsiones auténticas, carecen de un control preciso del nivel de incidencia de la distorsión en el video. Por tanto, esta tesis doctoral es un aporte al avance de la investigación de algoritmos de seguimiento de objetos que puedan funcionar correctamente en videos que contentan distorsiones auténticas.

Palabras clave

Evaluación de Calidad de Video, Seguimiento de Objetos en Video, Red Neuronal Convolucional, Predicción de Rendimiento de Seguidor de Objetos, Distorsiones en el momento de la captura, Seguimiento de objetos en Video Robusto, Base de Datos de Videos auténticamente distorsionados.

1 Introduction

1.1 Problem Statement

Tracking is a challenging task because of the need to work in videos with high contents of distortions and impairments such as bounding box initialization error, sensor noise, latency by video transmission, illumination changes, and data lost by compression algorithms [20]. An exciting direction of research is the interaction between visual quality and tracking tasks. Take into account that Surveillance videos are fraught with numerous sources of distortions, including blur, noise, and artifacts arising from processes such as compression, scaling, and format conversion. Multiple interacting distortions are often present, which significantly complicates the tracking task. Although tracking algorithms have been primarily implemented, mapping them against a wide variety of generally nonlinear, often commingled, poorly understood distortions in practice is complex.

Despite numerous algorithms proposed in the past few decades [3, 21, 22], video tracking methods that explicitly include strategies to make them robust to in-capture and post-capture video distortions have not been widely explored. Hence, there is a lack of (i) ad-hoc datasets designed for object tracking that present in-capture distortion and (ii) systematic evaluation of state-of-the-art video tracking algorithms under in-capture and post-capture video distortions. The existing body of work has studied various image factors that affect tracking performance. For instance, [23] specifies a list of factors such as occlusion, clutter, confusion, object shape, varying speed, and zooming, etc., that affect 'video quality. While these conditions affect tracking performance, there is no clear distinction between the scene–dependent challenges such as occlusion and clutter and the challenges imposed by traditional notions of "quality impairments" from capture, compression, processing, transmission, etc. This research is concerned with the latter interpretation of "quality" as it affects video tracking performance. A significant challenge is the complex interaction between video content and quality assessment in object tracking tasks, which raises the following research questions:

- How to create an ad-hoc dataset of pristine and distorted videos with in-capture and post-capture distortions to test video tracking algorithms?
- How usually deployed algorithms in video-analytics (video trackers) are affected by

in-capture (i.e., artifacts, color, exposure, focus, sharpness, stabilization) and post-capture (i.e., distortions generated by compression, transmission, and storage) visual distortions?

- How to design and test video object tracking algorithms explicitly robust in terms of performance with respect to in-capture and post-capture distortions?.

Diversely, another part of this research has to do with video quality assessment (VQA), an essential topic in several industries ranging from video streaming to camera manufacturing. Natural videos often contain in-capture distortions that affect the video quality perceived by humans. Video streaming and camera manufacturers are keen to understand the influence and presence of these distortions in natural videos. This quality prediction can be performed automatically using VQA algorithms. Nonetheless, one of the main challenges in VQA is video content dependency, which makes it difficult to generalize from a unique dataset. This issue motivated the following research question:

- How to develop fast and hand-crafted features free no-reference VQA algorithm to evaluate authentically distorted videos that contain a variety of visual content?

1.2 State of the art

This section presents the state-of-the-art of Video Quality Assessment (VQA) and Video Object Tracking (VOT) with an emphasis on the interaction and gaps between VQA and object tracking tasks.

Video Quality Assessment

Video Quality Assessment (VQA) is the set of techniques that attempt to quantify the quality of a video signal as seen by a human observer. VQA plays a crucial role in almost every aspect of video processing. VQA applications can be categorized as i)*Quality monitoring* Video quality is affected by numerous factors in video communication such as compression, noise, errors, congestion, and latency in networks [24]. ii)*Performance evaluation* Systematic evaluations of both hardware and software video processing systems that target human users are greatly facilitated by reliable means of VQA. The perceptual quality of images and video

generated by video devices can be automatically measured for competitive evaluation using VQA systems [24]. iii)*Optimization of video processing systems* Several video processing systems are designed by either specifying a maximum distortion of the video signal or alternately minimizing the video distortion for a detailed system configuration [25].

The only reliable means of evaluating the quality of a video as seen by a particular observer is to ask the human subject for their opinion of the visual quality of the video on a numerical or qualitative scale. This process is known as subjective VQA. Subjective VQA is expensive and tedious and is often challenging to perform in real-time applications. On the other hand, an objective Image Quality Assessment (IQA) refers to the automatic-computational model-based prediction of image quality as perceived by a human being [24].

The model is called a full-reference (FR) if the VQA prediction compares the original (i.e., undistorted) video and distorted video. When the metric uses only a small fraction of reference information (in the form of features extracted from the original video), the model is called reduced-reference (RR). Nonetheless, in many applications such as image denoising, deblurring, and enhancement where the reference video is absent, the VQA must be conducted without it. In these cases, a no-reference (NR) model permits VQA without access to the reference image. It is essential to highlight that it remains too much to be learned regarding full reference and reduced reference VQA, especially regarding the human visual perception of quality before generic blind algorithms become feasible. Next, we describe the current state of development of the FR, RR, and NR VQA metrics.

Full-reference VQA metrics The Mean Square Error (MSE), or equivalently, the Peak-Signal-to-Noise Ratio (PSNR), is often used to measure quality because of its simplicity and mathematical convenience. However, it is well known that MSE correlates poorly with visual quality [24]. Most VQA systems include a preprocessing or calibration stage at the front end. This phase accounts for registration (spatial and temporal alignment of reference and test video patches), modeling of the display device (gamma correction, eccentricity), viewing distance calibration, determination of the valid region of the video, and gain and offset calibration [26]. Next, we present three FR VQA algorithms that measure video quality using these calibrated videos [27].

Human visual system modeling based methods A substantial body of methods in the literature has focused on using models based on Human Visual System (HVS) in the design of VQA. The

HVS derives information from the environment from light that is either emitted, transmitted, or reflected from different objects within the environment [28]. The premise behind HVS-based metrics is to process visual data by simulating the optical pathway of the eye-brain system [24]. The reference and distorted videos are passed through computational models of various stages of processing that occur in the HVS, and visual quality is defined as the error measure between these signals computed from the output of the visual perception model [29, 30].

Feature based methods Feature-based models are based on statistical and visual features. The statistical models use statistical measures, such as mean, variance, covariance, and distributions, to model their respective quality metrics [31]. Visual feature-based models employ measurements of blurring and blocking in a video as well as image segmentation to extract significant visual information [26, 32, 33].

Motion modeling-based methods Accurate representation of motion in video sequences, as well as temporal distortions, have great potential for advancing video quality prediction [34]. A framework for evaluating the spatial and temporal (spatiotemporal) aspects of distortions in a video called Motion-based Video Integrity Evaluation (MOVIE) index was proposed by [35]. In this framework, video quality is evaluated in space, time, and space-time by estimating the motion quality along computed motion trajectories. The model's performance on the VQEG FRTV Phase 1 dataset was Pearson Linear Correlation Coefficient (PLCC)=0.821, Spearman's Rank Correlation Coefficient (SROCC) = 0.833.

Reduced reference VQA metrics: RR-I/VQA refers to I/VQA (Image/Video Quality Assessment) models that require partial information from the reference signal to predict the quality of a test signal. The outstanding RR-I/VQA algorithms include the wavelet-based RR-IQA algorithm in [36], the divisive normalization transform-based RR-IQA algorithm in [37], the information-theoretic RRED index in [38], and wavelet-based RR-VQA method in [39], among others [40]. In the most comprehensive comparison of VQA methods up to 2011 [27] the authors found that the natural visual statistics based MultiScale-Structural SIMilarity index (MS-SSIM), the natural visual feature-based Video Quality Metric (VQM), and the perceptual Spatio-temporal frequency-domain based Motion-based Video Integrity Evaluation (MOVIE) index deliver the best performance for the LIVE Video Quality Database.

No reference VQA metrics

Non-Reference Video Quality Assessment (NR-VQA) is challenging because of the lack of relevant statistical and perceptual models. Indeed, accurate modeling of motion and temporal change statistics in natural videos would be valuable because these attributes play an essential role in the perception of videos. Diverse approximations have been made to try to solve this challenge, in several fields [41–49]. In [50], an H.264-specific algorithm was proposed that extracts transformed coefficients from encoded bitstreams. The PSNR value is estimated between the quantized transformed coefficients and the predicted Non-quantized coefficients before encoding. The estimated PSNR was weighted using perceptual models in [51]. However, the algorithm requires knowledge of the quantization step used by the encoder for each macroblock in the video and is not applicable when this information is not available. Additionally, approaches have been developed that involve machine learning to learn human responses to distortion, using NSS features in the Wavelet domain [52], the Discrete Cosine Transform domain [53], or using Natural Scene Statistics (NSS) features as image edges [54] and perceptual characteristics [55].

Most of the related work in No Reference (NR) VQA models has focused on compression and transmission artifacts [56–69], and the most used applications of these NR-VQA models are quality monitoring in video storage, streaming, gaming, and broadcasting [70]. Mittal *et al.*. [71] created a No Reference Image Quality Assessment (IQA) model, called NIQE, that uses the statistical regularities observed in natural images without training on Mean Opinion Scores (MOS). They based this model on constructing a set of statistical features based on a Natural Scene Statistics (NSS) model. The IQA is given by the distance between the extracted NSS features and a multivariate Gaussian model of the quality-aware features obtained from the collection of images. In [72] Bampis *et al.*, applied NIQE to VQA and tested it on the LIVE-NFLX database [73], with little success, probably because it is a frame-based NR model (intended initially to IQA).

In [74], the authors proposed an NR IQA method that uses spatial features. The model, called Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE), uses scene statistics of locally normalized luminance coefficients to quantify losses of “naturalness” in an image due to distortions. BRISQUE is not limited by the type of distortion, which is advantageous compared to other approaches to NR IQA, which are distortion-specific [75]. Sogaard *et al.* [69] proposed

an NR VQA method; they built features by estimating selected video codec parameters along with the BRISQUE features applied to each frame. They exploited a changed version of the IQA BRISQUE and applied it to videos, along with features from the video codec analysis, and used these as input for an SVR machine learning method.

Saad *et al.* proposed in [53, 76, 77] an NR VQA method dubbed video BLind Image Integrity Notator using DCT Statistics (V-BLIINDS) that is built on a spatiotemporal model of videos converted to the discrete cosine transform domain and on a model that characterizes the motion occurring in the scenes to predict video quality. V-BLIINDS uses features extracted under the spatio-temporal Natural Video Scenes model to feed a support vector regressor (SVR) trained to predict the visual quality of videos. In this model, the spatial and temporal dimensions of videos were assessed collectively. They tested V-BLIINDS on the LIVE VQA database [35], and demonstrated good performance in terms of compression and packet loss [78].

The authors of [79] propose a distortion-specific approach based on a saliency map of detected faces. However, this approach is both semantic and distortion-dependent. Recently, Saad and Bovik [76] proposed an NR-VQA algorithm that has been shown to consistently correlates well with human judgments of temporal visual quality. The approach relies on a spatio-temporal model of video scenes in the discrete cosine transform domain and a model that characterizes the type of motion occurring in the scenes to predict video quality. This framework uses models to define video statistics and perceptual features that are the basis of a video quality assessment (VQA) algorithm that does not require a pristine video to compare against to predict a perceptual quality score. The proposed algorithm, called video BLIINDS (V-BLIINDS), was tested on the LIVE VQA database and the EPFL-PoliMi [80] video database and shown to achieve close to the level of top-performing reduced and full reference VQA algorithms. V-BLIINDS was used to characterize the quality of a set of videos of the proposed AD-SVD dataset. As the newly developed NR VQA algorithm, ST-BLIINDS [77] predominates, which extends from a DCT NR IQA domain to a temporal domain by training an SVR (Support Vector Regressor) over an NVS (Natural Video Statistics) model of image differences.

In [65] Mittal et al. validated VIIDEO using LIVE DATABASE [81]. VIIDEO is a non-reference VQA measure, which means that it does not require previous training or human judgment on pristine or distorted videos. VIIDEO is based on perceptually relevant temporal statistical video models of video frame difference signals to obtain a VQA score. To select appropriate features for the Asymmetric Generalized Gaussian Distribution model, three criteria were used:

(i) regularity over pristine videos, (ii) regularity should be destroyed in distorted videos, (iii) the loss of regularity varies with the degree of perceived distortion. The results of this study suggest that temporal characteristics work better than spatial characteristics for VQA. A comparison of VIIDEO with five FR indices (MSE, MS-SSIM [82], MOVIE [35], VQM [26] and ST-MAD [83]) and one RR index (STRRED [84]) showed that VIIDEO predicted, with higher correlation coefficients, human evaluations of video quality than MSE. Nonetheless, compared with other VQA measures, its performance is lower. Despite these results, it is essential to note that VIIDEO represents good progress in designing robust no-reference VQA measures because it does not require previous training or a piece of preliminary information. However, the authors did not test VIIDEO on a dataset with naturally distorted videos, such as LIVE-Qualcomm [15].

No reference VQA measures are convenient to support VQA concepts for computer vision tasks, such as object tracking in videos. Similar NR approaches have been successfully deployed for IQA and assignments, such as face detection and lesion analysis on dermoscopy images [85, 86]. Unlike traditional IQA algorithms, which use the human subjective score as ground truth, the ground truth is driven by the application and generated according to the degree of influence of the distortions on face detection or lesion analysis.

Kim et al. [87] analyzed the use of deep CNNs on the image-quality prediction problem. They tested different IQA NR and FR algorithms in five datasets, using two performance metrics to benchmark the models: Spearman's rank-order correlation coefficient (SRCC) and Pearson's linear correlation coefficient (PLCC). They found that, despite new datasets such as [88], the size of the database for picture-quality assessment remains an open question because the available dataset is a tiny size compared to the databases for image recognition. They also concluded that the pre-trained deep models adapted quickly to authentic distortions. The IQA algorithms that performed best in these IQA datasets were DIQA [89], ImageWise CNN, and ResNEt50 + fine-tuning [87].

Ghadiyaram *et al.* in [90] proposed a No-Reference IQA method, called Feature Maps–Based Referenceless Image Quality Evaluation Engine (FRIQUEE), based on a large set of perceptually relevant feature maps, along with NSS models. These features train an SVR learning model with radial basis kernel functions. They deployed these models in several color spaces (HSI, LMS, LAB, CIELAB [91]), and tested the proposed method, mainly on the LIVE In the Wild Image Quality Challenge Database (1162 natural images, obtained using mobile camera devices, and evaluated by 8100 human observers) [92, 93], achieving good quality

prediction on authentically distorted images.

In [94] the authors proposed a VQA method that considers video content. The algorithm calculates four measures: 1. picture resolution, 2. bitrates, 3. spatial Information (SI), and 4. temporal information (TI) to represent visual quality in four dimensions. Thus, they created a dataset with ten videos represented by MOSs provided by 20 subjects. However, they did not test VQA performance on other datasets. Furthermore, the proposed dataset is small and does not allow for adequate generalization.

Ghadiyaram *et al.* [15] proposed a dataset called LIVE-Qualcomm Mobile In-Capture Video Quality Database, comprising 208 videos obtained from eight different smartphones, modeling six common in-capture distortions. They conducted a subjective quality assessment study on this dataset, where 39 subjects assessed each video. Likewise, they tested several state-of-the-art No-Reference IQA and VQA algorithms [65, 71, 74, 76, 90] on this dataset and reported FRIQUEE [90] to be the best performing VQA algorithm on LIVE Qualcomm in terms of PLCC and SROCC, which were 0.7349 and 0.6795 respectively, with all distortions commingled.

In [95] Zhang *et al.* demonstrated the effectiveness of deep features as a perceptual metric. They found that deep features outperformed previous metrics by large margins in perceptual tasks. The authors remarked that this performance was independent of the CNN architecture and levels of supervision. Zhang demonstrated that even a CNN trained for common computer vision tasks achieved good performance on semantic and perceptual tasks [96]. However, this performance can be improved by training a simple linear scaling of layer activation using perceptual VQA or IQA datasets. Similarly, other studies determined that a CNN trained for object and video recognition can be helpful in determining human perceptual characteristics [97, 98].

Göring *et al.* [63] trained two no-reference models for classical video quality up to 4K resolution. The first is a BRISQUE+NIQE baseline model trained on per-frame VMAF scores, using a random forest regression model without feature selection. The second approach uses a pre-trained classification Deep Neural Network (DNN) combined with hierarchical sub-image creation. They introduced a hierarchical patching approach to train the DNN, dividing a frame into sub-images of equal size (1/2, 1/4, and 1/8 of each dimension). To generate deep features, the authors used the inception-v3 network [99], implemented in Keras, with re-scaling to 299$\times$

299 pixels. However, they tested the DeViQ method on a dataset containing only 12 videos, which led to low generalization capabilities. Zhang et al. [95] demonstrated that SSIM, FSIM, and PSNR metrics fail to account for many nuances of human perception. Moreover, they introduced a new dataset of human perceptual similarity judgments. In addition, they found that deep features outperform all previous metrics by large margins on that dataset in supervised and unsupervised levels. Finally, they concluded that perceptual similarity is an emergent property shared across deep visual representations.

VQA-NR Datasets: A number of IQA and VQA datasets have been designed in the last decade, including: LIVE [100], CSIQ [101], LIVE-Avvasi Mobile Video [102], CID 2013 [103], LIVE In the Wild Challenge [92], CVD 2014 [104], LIVE Mobile Stall Video [105], and LIVE Netflix Video Quality of Experience [106]. Most of the existing tracking and VQA datasets do not contain simultaneous in-capture and post-capture distortions or suffer from a single distortion type. Nonetheless, in [15] Ghadiyaram et al. captured and analyzed a dataset of 208 videos containing six types of in-capture distortions for VQA. The videos were recorded with eight smartphones to study how real-world in-capture distortions challenge both human viewers and automatic perceptual quality prediction models. Tsifouti et al. [107] generated degraded datasets that allow testing how video compression and frame rate reduction affect the performance of analytics systems. They concluded that the performance of the systems depends on the specific implementation of the software used for the compression, target bit rate, and frame rate. Furthermore, they reported that compression methods increased the number of false positives. In the future, they proposed the analysis of properties such as low contrast ratio and low brightness/dark events that affect the performance of video analytics systems.

In most picture-quality databases, distorted images are afflicted by only a single type of synthetically introduced distortion, such as JPEG compression, simulated sensor noise, or simulated blur. According to this and the survey of video datasets for tracking and VQA, it becomes evident that there is a need to have sets of videos that contain scenes of interest for video tracking applications, affected by different types of in-capture and post-capture authentically distortions. The intended solution to solve this issue is to create video sets with varied indoor and outdoor scenes of interest to test tracking algorithms that contain authentic in-capture distortions. This AD-SVD dataset is shared with the scientific community by creating an open-access repository in the IEEE Dataport website.

Video Object Tracking

To analyze tracking algorithms under different distortion conditions, we deployed 14 outstanding state-of-the-art tracker algorithms highly ranked by top conferences and VOT challenges [2, 108, 109]. These trackers are concisely defined in the **Appendix B**. We used the tracking performance measures proposed in [110] to describe the accuracy and robustness. In [110], the authors concluded that the average overlap measure is the most appropriate for use in tracker comparison, as it is simple to compute, it is scale and threshold invariant, exploits the entire sequence, and is easy to interpret. According to the correlation, the least correlated measures are the failure rate and average overlap of re-initialized trajectories. The average overlap measure is the best choice for measuring the accuracy of a tracker because it considers the size of the object and does not require a threshold parameter.

A measure that addresses the problem of the tracking length measure is the failure rate measure [110]. The failure rate measure casts the tracking problem as a supervised system in which a human operator reinitializes the tracker once it fails. The number of required manual interventions per frame was recorded and used as a comparative score. We consider a failure when the bounding box overlap is zero because we are interested only in the most apparent failure with no overlap between regions. Robustness is defined as an exponential failure distribution, $R_s = e^{SM}$. The value of M denotes meantime-between-failures, that is, $M = \frac{F_0}{N}$, where N is the length of the sequence. The reliability of a tracker can be interpreted as the probability that the tracker will still successfully track the object up to S frames since the last failure, assuming a uniform failure distribution that does not depend on previous failures [110].

Many of these trackers deploy discriminative classification based on the distinction between the target foreground and the background. This approach is a convenient framework to include VQA features such as Natural Scene Statistics (NSS) [111, 112] to enrich the foreground and background representation space and tracker performance even in the presence of high levels of distortion. The NSS has commonly been used to extract statistical features in universal image quality assessment [113]. The features extracted from the NSS statistics are altered in the presence of distortions [114]. Using NSS, it is possible to identify the distortion afflicting the image and perform no-reference (NR) IQA [52]. The goal of an NR VQA algorithm based on NSS is to capture “unnatural” statistics in the distorted image and relate it to image quality.

Video Object Tracking Datasets: Currently, there are more than 68 different datasets used for tracking and action recognition [115]. Some of the most used for test the state-of-the-art tracking algorithms are [116]: CAVIAR [117], KTH actions [118], UCF101 Sport Actions [119], VOT-2014 [1], VOT-2020ST [2], VOT-2020LT [2], OTB50 [3], OTB100 [3], UAV20L [4], UAV123 [4], TC128 [5], NUS-PRO [6], LaSOT [7], VQUAD [8], TrackingNet [9], Got-10k [10], ALOV [21], NfS [120], UAVDT [121], VisDrone2019L [122], Small-90/Small-112 [123], and Youtube action dataset [124].

With more than 100 citations, CAVIAR and KTH are highly used datasets [115]. CAVIAR dataset [117] contains two sets of data filmed in two different indoor scenarios. The sequences recorded in INRIA consist of four clips with fights and 24 clips with other activities, such as group walking and meeting. The KTH dataset contains six types of human actions (walking, jogging, running, boxing, hand waving, and hand clapping) performed several times by 25 people in four different scenarios. On the other hand, CAVIAR [117] was the first dataset recorded in more complex environments, where background and illumination were not controlled. ActivityNet [125] is a large-scale video benchmark for understanding human activity. This benchmark aims to cover a wide range of complex human activities that interest people in their daily lives. In its current version, ActivityNet provides samples from 203 activity classes with an average of 137 untrimmed videos per class for a total of 849 video hours.

Video Quality in the Object Tracking Task

Tracking is the analysis of video sequences for establishing the location of a target over a sequence of frames (time) [21]. Currently, it has applications ranging from security and surveillance [126, 127], intelligent systems [128], to medical imaging and augmented reality [129–131]. Previous studies on the impact of quality on video tracking have analyzed post-capture and in-capture distortions. Post-capture distortions refer to aberrations that are synthetically introduced into videos after capture. In contrast, in-capture distortions are naturally occurring impairments such as texture distortions, artifacts due to exposure, lens limitations, focus, and color aberrations. For instance, [3, 132, 133] studied the performance of trackers under distortions synthetically introduced to a video, such as 'salt and pepper', and additive Gaussian noise. Moreover, a low frame rate (≤ 25) can affect tracking performance [134, 135]. However, it was reported that the performance decreased with an increased bit rate from 400 to 1600 kbps [107].

A challenging aspect of the design of robust trackers concerning distortions is that high visual quality may not equate to high application value in this application. For example, in video analysis for tracking, the quality of frames should be expressed in terms of benefit in the sense that image analysis tasks should be more efficient on images having high “task quality”. NSS features are widely used in many state-of-the-art IQA algorithms [3, 21]. Natural scene statistic models seek to capture the statistical properties of natural scenes that hold across different contents [136]. VQA can be carried out by analyzing the local statistics of frame differences F^t of videos that have been debiased and normalized using a procedure called mean subtraction and contrast normalization (MSCN). Different image distortions alter the regularity of the MSCN coefficients in a particular manner. Authors define the frame difference ΔF^t between consecutive frames F^{2t+1} and F^{2t} of spatial dimensions $M \times N$ and *T* as the total number of frames, as follows [65]:

$$\Delta F^t = F^{2t+1} - F^{2t} \quad \forall\, t \in \{0, 1, 2...\frac{T-1}{2}\} \tag{1}$$

In [84], it was shown that the band-pass filter coefficients of frame differences capture temporal statistical regularities arising from structures such as moving edges. These frame differences are operated on via the MSCN following the NSS model [65, 137]:

$$\Delta\hat{F}(i,j) = \frac{\Delta F^t(i,j) - \mu^t(i,j)}{\sigma^t(i,j) + C} \tag{2}$$

over spatial indices $i \in \{1, 2...M\}, j \in \{1, 2...N\}$, over a set of consecutive frame time samples $t \in \{0, 1, 2...\frac{T-1}{2}\}$ where

$$\mu^t(i,j) = \sum_{k=-K}^{K} \sum_{l=-L}^{L} \omega_{k,l} \Delta F^t(i+k, j+l) \tag{3}$$

and

$$\sigma^t(i,j) = \sqrt{\sum_{k=-K}^{K} \sum_{l=-L}^{L} \omega_{k,l} [\Delta F^t(i+k, j+l) - \mu(i,j)]^2} \tag{4}$$

estimate the local time-differenced luminance averages and contrasts. Each frame has a spatial

dimension of $M \times N$. $\omega_{k,l}$ is a 2D weighted isotropic symmetric Gaussian function. To estimate the shape parameter γ of PDF, Sharifi in [138] proposed a method that given the mean, variance and the mean absolute value of the image X ($E[|X|]$), allows to find γ, as follows:

$$\gamma = \frac{\sigma_X^2}{E^2[|X|]} \tag{5}$$

In [65], Mittal et al. showed that when natural video frame differences are subjected to commonly encountered unnatural distortions, their processed coefficients no longer tend toward Gaussianity. Similarly, the authors demonstrated that the signs of adjacent coefficients also exhibit a regular structure, which is disturbed by distortion. The products of neighboring coefficients have been shown to be well-modeled as follows a zero-mode asymmetric generalized Gaussian distribution (AGGD) [139]:

$$f(x;\gamma,\beta_l,\beta r) = \begin{cases} \frac{\gamma}{(\beta_l+\beta_r)\Gamma(\frac{1}{\gamma})}\, exp((\frac{-x}{\beta_l})^\gamma) & \forall x \leq 0 \\ \frac{\gamma}{(\beta_l+\beta_r)\Gamma(\frac{1}{\gamma})}\, exp((\frac{x}{\beta_l})^\gamma) & \forall x \leq 0 \end{cases} \tag{6}$$

The parameters of the AGGD $(\gamma, \beta_l, \beta_r)$ can be efficiently estimated using Eq.(7) and Eq.(12):

$$\hat{\gamma} = \frac{\sqrt{\frac{1}{N_l-1}\sum_{k=1,x_k\leq 0}^{N_l} x_k^2}}{\sqrt{\frac{1}{N_r-1}\sum_{k=1,x_k\leq 0}^{N_r} x_k^2}} \tag{7}$$

Where $N_l(N_r)$ is the number of x_k samples (≥ 0). An unbiased estimate of r is [139]:

$$\hat{r} = \frac{(\sum |x_k|^2)}{\sum x_k^2} \tag{8}$$

$\hat{R}$ is calculated using Eq.(9):

$$\hat{R} = \hat{r}\frac{(\hat{\gamma^3}+1)(\hat{\gamma}+1)}{(\hat{\gamma^2}+1)^2} \tag{9}$$

According to $\hat{R}$ value, α is estimated using the approximation of the inverse generalized Gaussian ratio [140].

$$\hat{\alpha} = \hat{\rho^{-1}}(\hat{R}) \tag{10}$$

The parameters of left and right scale are estimated with [139]:

$$\hat{\beta}_l = \sqrt{\frac{1}{N_l - 1} \sum_{k=1, x_k \leq 0}^{N_l} x_k^2} \times \sqrt{\frac{\Gamma(\frac{3}{\hat{\alpha}})}{\Gamma(\frac{1}{\hat{\alpha}})}} \tag{11}$$

$$\hat{\beta}_r = \sqrt{\frac{1}{N_r - 1} \sum_{k=1, x_k \leq 0}^{N_r} x_k^2} \times \sqrt{\frac{\Gamma(\frac{3}{\hat{\alpha}})}{\Gamma(\frac{1}{\hat{\alpha}})}} \tag{12}$$

With the gamma function $\Gamma(\cdot)$ given by [141]:

$$\Gamma(x) = \int_0^\infty t^{z-1} e^{-t} dt \tag{13}$$

As in the BRISQUE model [74], paired product coefficients are also extracted at each coordinate by multiplying neighboring MSCN coefficients along with four directions: horizontal(H), vertical(V), main-diagonal(D1), and secondary-diagonal(D2). These coefficients capture the directional correlation behavior of images, which are also perturbed by the presence of distortions [142]. One option would be to use the NSS features BRISQUE f and pp, which generated the best results in the study made in [142]. Other examples of NSS models that are potentially applicable are: i) BRISQUE [74], which proposes an NR IQA approach that utilizes an asymmetric generalized Gaussian distribution (AGGD) to model images in the spatial domain. The modeled image features here are the differences of spatially neighbored, mean subtracted, and contrast normalized image samples. ii) NIQE [71] extracts features based on a multivariate Gaussian model and relates them to perceived quality in an unsupervised manner [143]. iii) FRIQUEE [102, 144, 145] is based on the hypothesis that different existing statistical image models capture distinctive aspects of the loss of the perceived quality of a given image. FRIQUEE embodies 564 statistical features that have been observed to contribute meaningful information regarding image distortion visibility and perceived image quality [92]. Similarly, can also be used the 1/f model of the amplitude spectrum of visible light (VL) images, sparse coding characteristics of cortical-like filters [146], and the underlying gaussianity of perceptually-processed band-pass images [113].

Nawaz and Cavallaro [20] distinguish between physical world conditions from a scene that

challenge tracking performance, such as illumination changes and video quality impairments from conditions in capture, compression, processing, and transmission such as sensor noise, compression of the video data, initialization errors caused by a detector, and latency due to the transmission of video data over a channel or due to the delayed generation of results by the tracker. Minyoung et al. [147] developed a face tracker and recognizer with visual constraints. Their approach was tested on a large set of noisy real-world videos, which had low resolution and were recorded at high compression rates. Korshunov et al. [148] analyzed the maximum amount of video distortions (regarding critical video quality measured by blockiness and mutual information) that a face tracker can tolerate before the accuracy of the face tracking and recognition algorithm drops significantly. Because of the conditions in the capture, they proposed two alternative video quality metrics to estimate the critical video quality for face analysis algorithms. A typical surveillance camera produces a video with a resolution of not less than 320×240, and 10-30 fps, the minimum quality desirable for the human visual system. Post-capture distortions such as JPEG compression, scaling, a combination of compression and scaling, and frame drop affected the videos analyzed in this study. This work suggests that it is impractical and inefficient to treat video analysis algorithms in the same manner as a human video observer.

Nieto et al. [149] evaluated two trackers (STRUCK and TLD) in surveillance videos affected by in-capture distortions, such as under-exposure and defocus. Even though STRUCK and TLD have ranked high in video tracking surveys [3, 21], this study concludes that in-capture distortions severely affect the performance of these trackers. Xu et al. [150] guided the quality assessment of surveillance videos when they found the critical video quality that can be used to reduce the bitrate of video transmission without significantly affecting the accuracy of the surveillance tasks (tracking). The essential factors influencing surveillance tasks are the differences in spatial and temporal activities and target character size. They concluded that an average bitrate of 1000 kbit/s results in 80% of detection probability in surveillance tasks. Nonetheless, owing to the relatively high diversity of scenarios, critical bitrate strongly depends on the scene content.

1.3 Research Hypothesis

Learned deep spatial-time features are useful for predicting VOT algorithm performance and perceptual quality in authentically distorted videos.

1.4 Research Scope

This thesis aims to propose new methodologies based on deep perceptual features for the intersected fields of machine vision (VOT algorithm prediction performance) and human perceptual quality assessment of authentically distorted videos.

1.5 Document Outline

In detail, the rest of the thesis is structured as follows:

- Chapter 1 provides a brief introduction to the presented problem. We briefly presented briefly state-of-the-art NR-VQA methods, Video Object Tracking, and VQA applied to VOT tasks.
- Chapter 2 presents the general and specific objectives accomplished with this research.
- Chapter 3 recapitulates the proposed AD-SVD dataset, along with a benchmark of 11 state-of-the-art trackers on this dataset. AD-SVD comprises more than 4476 videos affected by in-capture distortions, acquired by four different surveillance cameras and recorded at three outdoor and four indoor locations.
- Chapter 4 defines a novel method for Non-Reference VQA. This framework is fast and does not require the extraction of handcrafted features. We extracted the convolutional features of the 3-D C3D Convolutional Neural Network and fed one trained Support Vector Regressor to obtain a VQA score. To generate discriminant and perceptually relevant deep features, we carried out transformations to different color spaces. We extracted features from several layers, with and without overlap, to find the best configuration for improving the VQA score. This chapter was published in [151].
- Chapter 5 provides a framework to improve tracker robustness against post-capture distortions based on quality-aware feature selection for VOT. We defined the best features HOG and NSS that generate the most significant area under the curve in the success plots, yielding an improvement in the video tracker performance in videos affected by post-capture distortions. We proposed an approach to integrate NSS perceptual quality features into a video object tracker scheme and demonstrated its performance in several videos affected by post-capture distortions.

- Chapter 6 introduces a model-agnostic (independent of the tracker model) framework that predicts performance without running the corresponding tracking algorithm. The method predicts the performance of a VOT algorithm with high accuracy in such a way that the probability of obtaining the reference output is maximized without executing the tracking algorithms. To this end, we learn a mapping between the input video and the area under the curve (AUC) of the success plot. 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.
- Chapter 7 proposes a framework to reduce video tracker computation resources (time and video storage space). This time reduction 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 tracker's video storage demands and execution time, thereby preserving its performance.
- Chapter 8 provides an in-depth discussion of the results obtained from the five investigations, along with the proposed conclusions.
- Chapter 9 presents recommendations for further work, based on the findings and results.
- Chapter 10 resume the publications produced with the development of this thesis.

1.6 Associated Software

The software originally developed and the state-of-the-art algorithms implemented from open-source papers with code alongside the modifications made are publicly available under a unified repository at https://github.com/roger-26/Phd_Code.

2 Objectives

2.1 General Objective

To create one robust video tracker against in-capture and post-capture distortions using Video Quality Assessment based on Natural Scene Statistics and Deep Learning.

2.2 Specific Objectives

The research questions (RQ) that this doctoral proposal propose solution strategies are expressed as follows:

SO.1.

To create an ad-hoc dataset of pristine and distorted videos with in-capture and post-capture distortions to test video tracking algorithms.

SO.2

To study how usually deployed algorithms in video-analytics (video trackers) are affected by in-capture (i.e artifacts, color, exposure, focus, sharpness, stabilization) and post-capture (i.e distortions generated by compression, transmission, and storage) visual distortions.

SO.3

To design and test video object tracking algorithms explicitly robust in terms of performance with respect to in-capture and post-capture distortions.

3 Authentically Distorted Surveillance Video: Proposed Dataset

In this chapter, we present the Authentically Distorted Surveillance Videos Dataset (AD-SVD) proposed. This dataset was proposed to test Video Trackers algorithms in videos with authentic distortions at quantified levels. This dataset is available to the scientific community at https://ieee-dataport.org/open-access/authentically-distorted-surveillance-videos-dataset. AD-SVD comprises more than 4476 videos affected by in-capture distortions, acquired by four different surveillance cameras [152]. The videos in this dataset have an equal rate of I/P frames of 10 fps. This frame rate is typical for commercial video surveillance applications. Minimization of storage costs also motivates the frame rate selection. The frame size is FHD (1920$\times$ 1080), and the color space is three RGB channels, and the exposure variation range is $\left\{\frac{1}{480}, \frac{1}{120}\right\}$ seconds. The video dataset also contains H.264/AVC compression post-capture distortions at three different bitrates, resulting in three mirrored video sequences that change only in the level of compression. We chose three different bitrates (4700, 1800, and 1200 kbps) to generate degradation over the distortion scale (from imperceptible to very annoying). Similarly, based on the AD-SVD proposed dataset, we assessed 11 state-of-the-art trackers using the A-R and success plot performance measures. We demonstrated that in-capture distortions severely hamper the performance of VOT methods in a non-intuitive manner.

VOT is a well-studied and rapidly advancing field. VOT remains a challenging task because only the initial state of the target is available. Despite the plethora of VOT methods existing in the literature, there is a lack of detailed studies analyzing the performance of videos with authentic in-capture and post-capture distortions. Such an analysis requires a database with videos containing the distortions mentioned above in a controlled and quantifiable manner. In [108], the authors proposed a standard set of evaluation measures for VOT 2017 [1, 153]. However, this dataset lacks video sequences, including in-capture and post-capture distortions in outdoor or indoor environments. A significant number of video quality databases have been designed in recent years [92, 102, 104, 106, 154]. These databases were generated by systematically distorting a small set of high-quality videos in a controlled manner. Most of the existing VOT and Video Quality Assessment -VQA- datasets do not contain simultaneously in-capture and post-capture distortions or only have a single distortion type [100]. Furthermore, these databases do not include authentic in-capture distortions [118].

Deepti *et al* presented in [15] a video database containing in-capture distortions for VQA. It comprises 208 videos captured using eight different smartphones. The videos in this database contain six common in-capture distortions: artifacts, color, exposure, out-of-focus, sharpness, and stabilization. For instance, Tsifouti *et al.*. [107] generated degraded datasets that allow testing of how video compression and frame rate reduction affect the performance of video-analytic systems. In this way, they reported an increased false-positive ratio due to compression methods. They concluded that the performance depends on the specific implementation of the compression software used, the target bit rate, and the frame rate. Despite these advances, very little work has been done on construction databases affected by in-capture distortions for video surveillance applications to the best of our knowledge.

3.1 AD-SVD characteristics

Since a similar resource was not already 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 the in-capture distortions [11]. The hardware characteristics for these cameras are detailed in Appendix C. AD-SVD is publicly available at the IEEE DataPort[2]. 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.

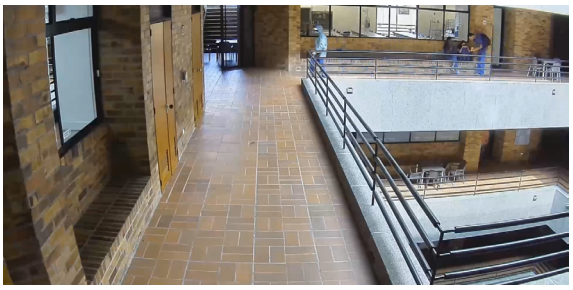
Table 1 shows the number of videos grouped according to condition, location, and activity. AD-SVD contains the following activities [11]:

- **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 circuit.

[2]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.

(a) Parking lot 1

(b) Parking lot 2

(c) Theater

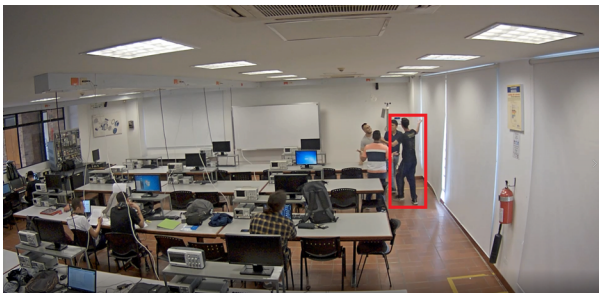
Figure 2: Outdoor locations.

- **Prowl (PW):** A person makes suspicious movements when searching for 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 circuit.

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

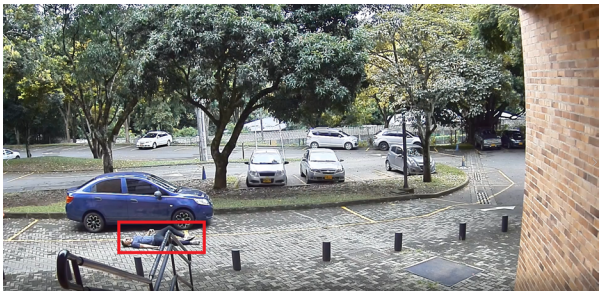
Condition	**# Videos**
Pristine	160
Exposure	1457
Defocus	1409
Mixed	1450
Total Videos	**4476**
Location	**# Videos**
1. Theater	856
2. Parking Lot 2	41
3. Parking Lot 1	801
4. Media Room	851
5. Industrial Lab	889
6. Guayacanes Hall	152
7. Electronics Lab	886
Total Videos	**4476**
Activity	**# Videos**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

Different datasets are currently used to evaluate the VOT algorithms. Figure 4 summarizes the parameters 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* [7], *VOT2020-LT* [2] and *UAV20L* [4] were used for long-term tracking (LTT) evaluation. By contrast, AD-SVD was created for short-term tracker (STT) assessment. AD-SVD, *LaSOT* [7] and TrackingNet [9] 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



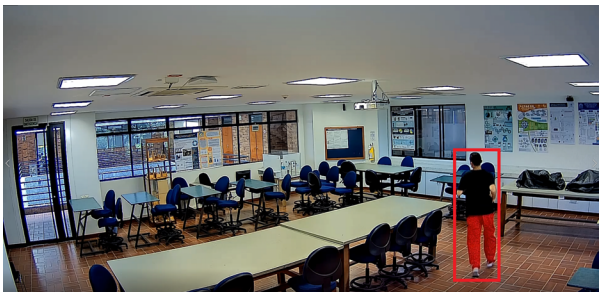
(a) Fighting in group

(b) Leaving package in a public space



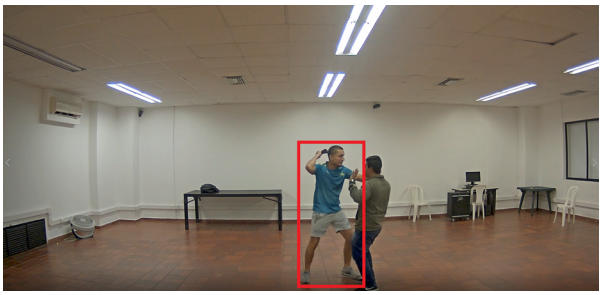
(c) Person passing out

(d) Person pushing people

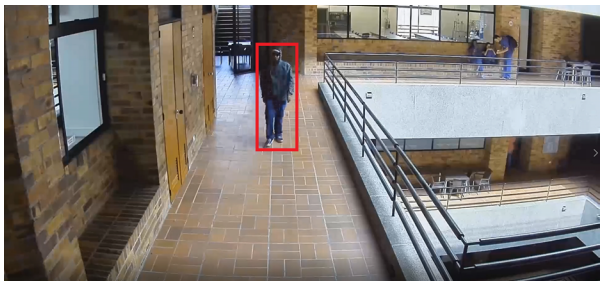


(e) Person running

(f) Prowling



(g) Robbing with a knife



(h) Person walking

Figure 3: Activities recorded in AD-SVD dataset.

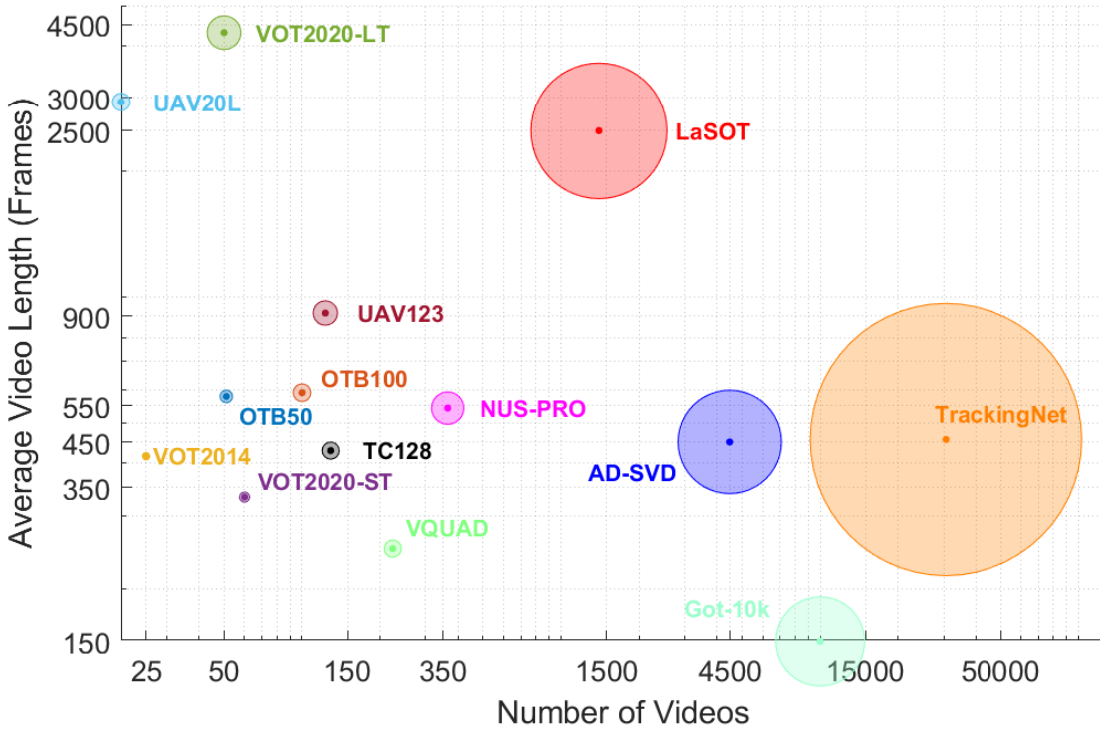


Figure 4: VOT benchmarks with high quality dense (per frame) annotations, including VOT-2014 [1], VOT-2020ST [2], VOT-2020LT [2], OTB50 [3], OTB100 [3], UAV20L [4], UAV123 [4], TC128 [5], NUS-PRO [6], LaSOT [7], VQUAD [8], TrackingNet [9], Got-10k [10] and AD-SVD. The circle diameter is proportional 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.

distorted video object tracking benchmark for STT.

Bounding Box Annotations

In AD-SVD, each video had 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 [155] annotating every five frames. An interpolator algorithm was used to obtain all the labels for each frame in the video, based on the five-frame annotations. Because AD-SVD evaluates Short Term Tracking (STT) algorithms, the region of interest (ROI) is present throughout the entire sequence, and each frame is labeled.

Table 2: Tracking Benchmarks Summary.

Benchmark	Av. Video Length	# Videos
TrackingNet [9]	456	30643
Got-10k [10]	150	10000
AD-SVD	**450**	**4476**
LaSOT [7]	2500	1400
NUS-PRO [6]	542	365
TC128 [5]	429	129
UAV123 [4]	915	123
OTB100 [3]	590	100
VOT2020-ST [2]	332	60
OTB50 [3]	578	51
VOT2020-LT [2]	4306	50
VOT2014 [1]	416	25
UAV20L [4]	2934	20

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 the 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 has levels low (1), medium (2), and high (3) set accordingly with each video camera configuration.

Because the configuration of the distortion levels in the four video cameras is not identical, and they are different brands and models, we used the reliable No-Reference Video Quality Assessment (VQA) metric V-BLIINDS [76]. Using V-BLIINDS, we tested the consistency of the parameter settings of the cameras used to record the AD-SVD videos. The parameters used to obtain the in-capture distortions with the four surveillance cameras are detailed in Appendix D. Figure 8 shows the box plots of the 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 decided on this reduced set because V-BLIINDS is computationally expensive. Defocus, exposure, and Defocus+Exposure distortions and

(a) Low exposure level

(b) Medium exposure level

(c) High exposure level

Figure 5: Examples of frames with different exposure levels.

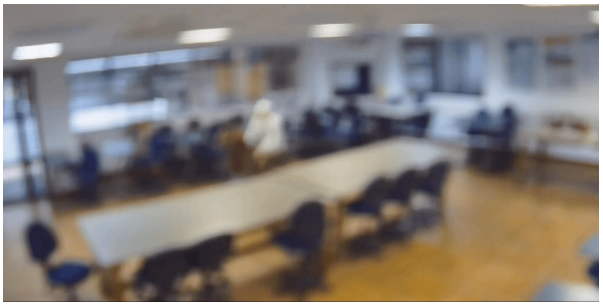
pristine videos were represented by 283, 292, 284, and 33 videos, respectively. The higher the V-BLIINDS values, the worse the perceptual visual quality of the video. We observe that perceptual quality decreases (V-BLIINDS scores increase) in the following order: pristine, exposure, defocus, and combined exposure, and defocus distortions, as expected. Videos affected by exposure distortions exhibited more significant variability in V-BLIINDS scores for AD-SVD. In contrast, V-BLIINDS [76] scores of videos affected by defocus and commingled defocus and exposure impairments showed minor variance.

3.2 Benchmarking of Video Object Trackers

Despite the plethora of competitive VOT methods presented in contests such as VOT 2017 [108] and 2018 [109], there is a lack of detailed studies analyzing the performance of videos with authentic in-capture and post-capture distortions. To conduct 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

(a) Low defocus level

(b) Medium defocus level



(c) High defocus level

Figure 6: Defocus levels.

additional tracker evaluated: scale DL-SSVM [156]. Table 3 lists the algorithms chosen along with the VOT rankings and features.

Table 3: Evaluated Trackers.

Tracker	**Contest**	**VOT Ranking**	**Features**[3]
CFWCR [157]	VOT 2017	2	CNN/ECO
Gnet [158]	VOT 2017	5	CNN/DCF
MCCT [159]	VOT 2017	6	DCF
LADCF [14]	VOT 2018	1	DCF/HOG
MFT [160]	VOT 2018	2	DCF/ECO
RCO [109]	VOT 2018	5	DCF/CNN
DeepSTRCF [161]	VOT 2018	7	DCF/CNN
CPT [162]	VOT 2018	8	DCF/VGG
SRCT [163]	VOT 2018	12	SRB/ECO
CPT_fast [162]	VOT 2018	14	DCF/VGG
Scale DLSSVM [156]	-	-	MSE/LK

[3] **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.

(a) Low defocus+exposure

(b) Medium defocus+exposure

(c) High defocus+exposure level

Figure 7: Defocus+Exposure levels.

Evaluation measures

Tracker parameters were set to their respective default values and kept constant during the experiment. Each tracker was executed 30 times for each sequence, considering the stochastic processes. This number of executions is sufficient for the statistical evaluation of the correlation across the measures. We used the performance measures proposed by [110], to analyze the accuracy and robustness. These are described as follows.

Average Overlap The average overlap measure is the most appropriate for use in a VOT algorithm comparison. It offers several advantages: simple computation, scale, and threshold invariance that exploit the entire sequence and a clear and concise interpretation. According to the correlation, the minor correlated measures are the failure rate and average overlap of re-initialized trajectories. The average overlap measure can be considered the best choice for measuring the accuracy of a tracker because it contemplates the size of the object and does not require a threshold parameter.

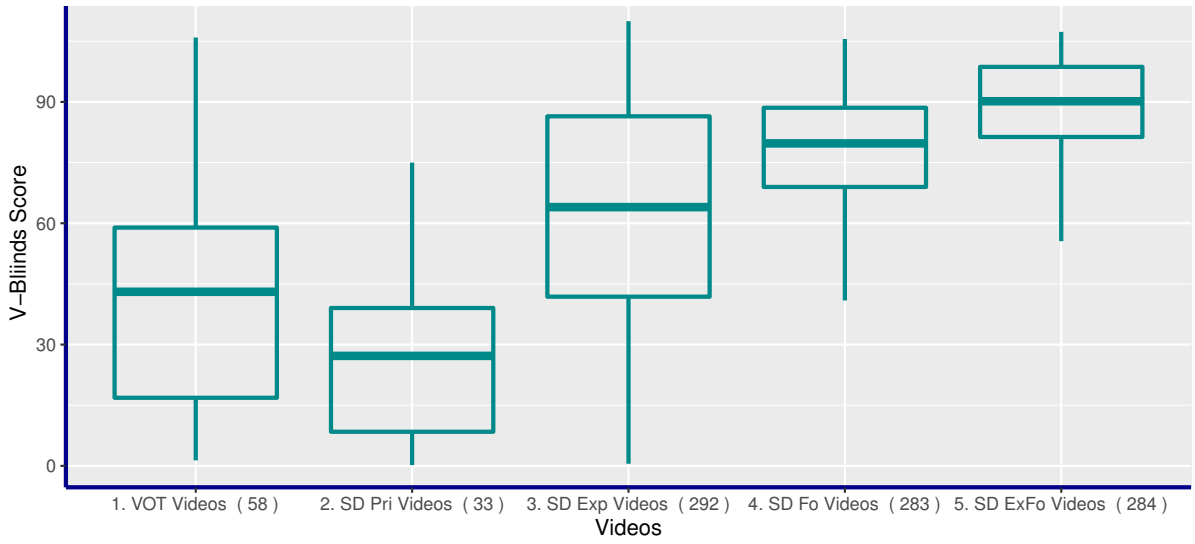


Figure 8: V-BLIINDS distributions on AD-SVD and VOT 2018 datasets, where the V-BLIINDS score is inversely proportional to the video quality.

Failure Rate The failure rate measure addresses the problem of the VOT length measure. It casts the VOT problem as a supervised system in which a human operator reinitializes the tracker once it fails. The number of required manual interventions per frame was recorded and used as a comparative score. We declare a failure when the bounding box overlap is zero because we are only interested in the most apparent failure without overlap between regions. Robustness is defined as an exponential failure distribution, $R_s = e^{SM}$. The value of M denotes meantime-between-failures, that is, $M = \frac{F_0}{N}$, where N is the length of the sequence. The reliability of a tracker can be interpreted as a probability that the tracker will still successfully track the object up to S frames since the last failure, assuming a uniform failure distribution that does not depend on previous failures. In this study, we assumed that the S=30.

Benchmarking Results

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 higher 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 9.

The area under the curve (AUC) (i.e., area under the success plot) of each tracker allowed 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

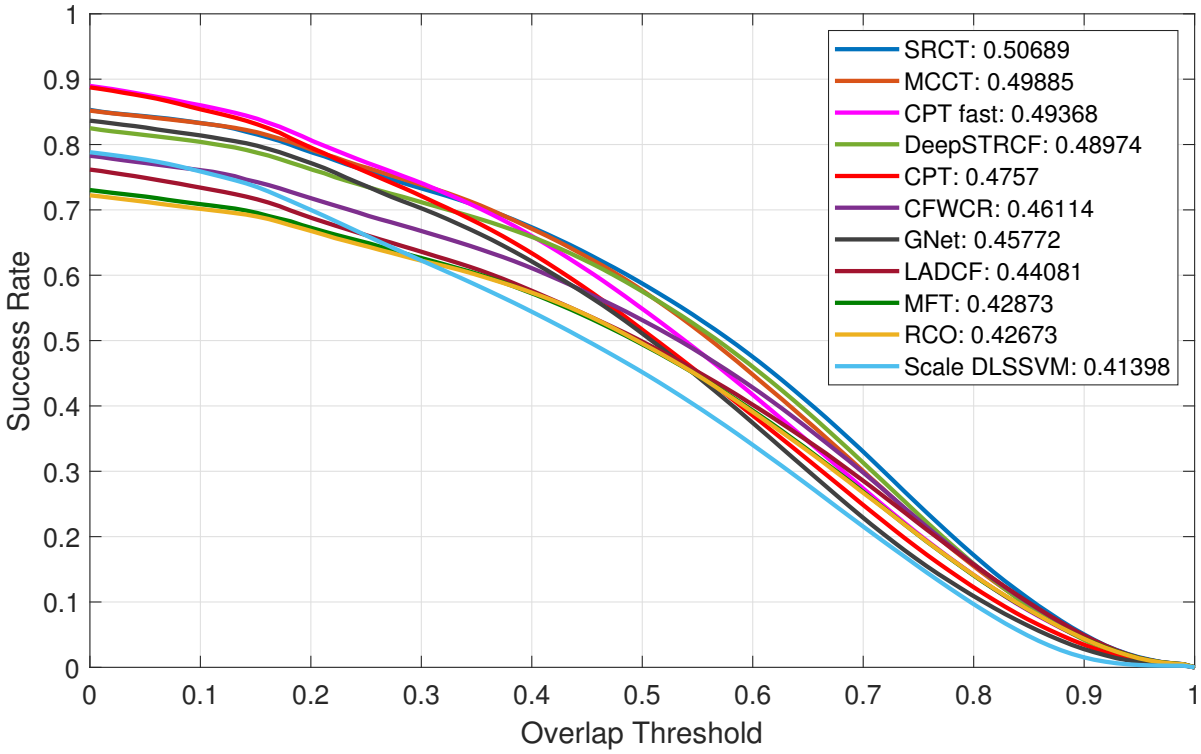


Figure 9: Success plot for each tracker after evaluating in the AD-SVD. Tracker:AUC

organize the trackers according to their AUCs. In line with the AUC of each algorithm, the SRCT [163] tracker achieved the best performance, while the Scale DLSSVM [156] tracker yielded the worst performance. The first-place winners in both contests (VOT 2017 [108] and VOT 2018 [109]) 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 [163]) uses a combination of Salient Region-Based (**SRB**) and Efficient Convolution Operators (**ECO** [164]) techniques, which have also been the basis of other state-of-the-art video trackers. In addition, the correlational filter (CF) technique has been used successfully in several VOT algorithms, such as DeepSTRCF [161], owing to its low computational cost and speed.

In the trackers using CF, from the first frame, the object is tracked by correlating the filter in regions close to the detected object across the following frames [165]. 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 [166]: categorized features [167], space weight factors [168–170], scale factors [171], and expert strategies [172].

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

Tracker	**AUC**	**AD-SVD Ranking**	**VOT Ranking**
SRCT [163]	0.5069	1	12 (2018)
MCCT [159]	0.4988	2	6 (2017)
CPT_fast [162]	0.4936	3	14 (2018)
DeepSTRCF [161]	0.4897	4	7 (2018)
CPT [162]	0.4757	5	8 (2018)
CFWCR [157]	0.4611	6	2 (2017)
Gnet [158]	0.4577	7	5 (2017)
LADCF [14]	0.4408	8	1 (2018)
MFT [160]	0.4287	9	2 (2018)
RCO [109]	0.4267	10	5 (2018)
S. DLSSVM [156]	0.4139	11	-

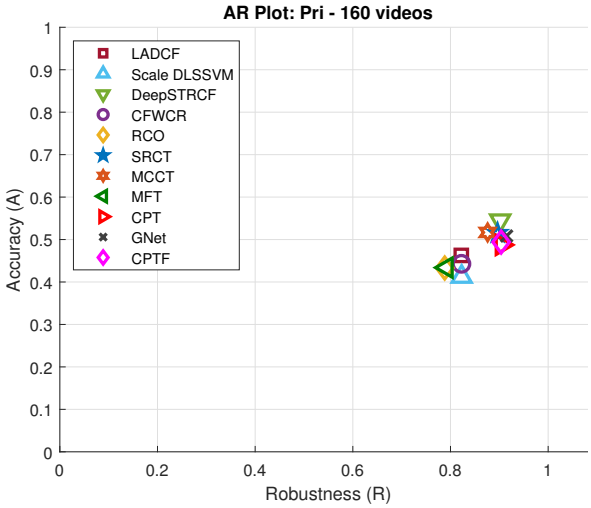
We used the metrics robustness and accuracy to evaluate the tracker performance per distortion. Robustness R is the number of times the tracker fails and must be reinitialized. A video tracker fails (and reinitialization is triggered) when the overlap ϕ_i (Eq. 14) drops to 0. A_t^G and A_t^T are the 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. 15) is calculated, assuming a uniform failure distribution that does not depend on previous failures [110]. Accuracy A in Eq. 16 is the average overlap over all the frames in a video sequence [109, 110], where the number of frames is N_{frames}.

$$\phi = \frac{|A_t^G \cap A_t^T|}{|A_t^G \cup A_t^T|} \tag{14}$$

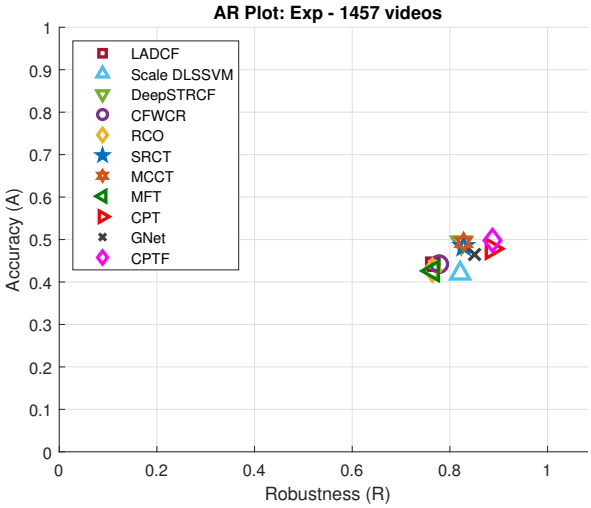
$$R_k = e^{(\frac{-SF_k}{N_{frames}})} \tag{15}$$

$$A = \frac{1}{N_{frames}} \sum_{i=1}^{N_{frames}} \phi_i \tag{16}$$

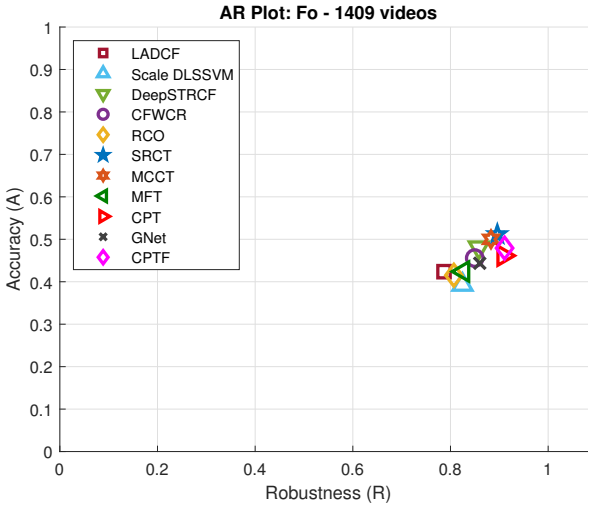
Table 5 and Figure 10 present the most accurate and robust trackers for distortion. Concerning robustness, trackers CPTfast [162] and CPT [162] performed very well, which means they were good at tracking an object without losing it as the video progresses. The fact that the best-ranked



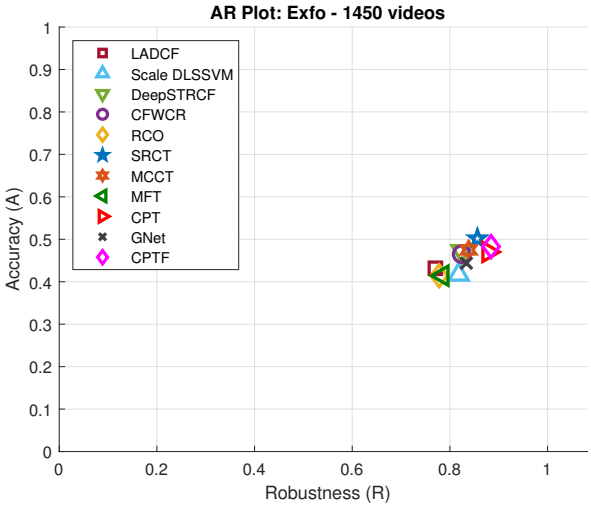
(a) Pristine videos



(b) Videos with exposure distortion.



(c) Videos with defocus distortion.



(d) Videos with defocus and exposure commingled distortions.

Figure 10: AR Plots per Distortion ($S = 100$, **Pri:** pristine, **Exp:** exposure, **Fo:** focus, **Exfo:** focus + exposure) [11].

trackers in the VOT 2018 and 2017 contests do not demonstrate the best performance in the AD-SVD dataset implies that a new benchmark baseline is necessary where the authors can evaluate the trackers on authentically distorted videos. This benchmark is important because the actual videos coming from the cities' surveillance cameras are affected by in-capture distortions, so it is necessary to guarantee the applicability of the tracking algorithms to test them in similar videos.

Table 5: Best performing trackers for each considered distortion.

Distortion condition	**Most Accurate**	**Most Robust**
Pristine	DeepSTRCF [161]	Gnet [158]
Exposure	CPT_fast [162]	CPT_fast [162]
Defocus	SRCT [163]	CPT_fast [162]
Defocus + Exposure	SRCT [163]	CPT_fast [162]

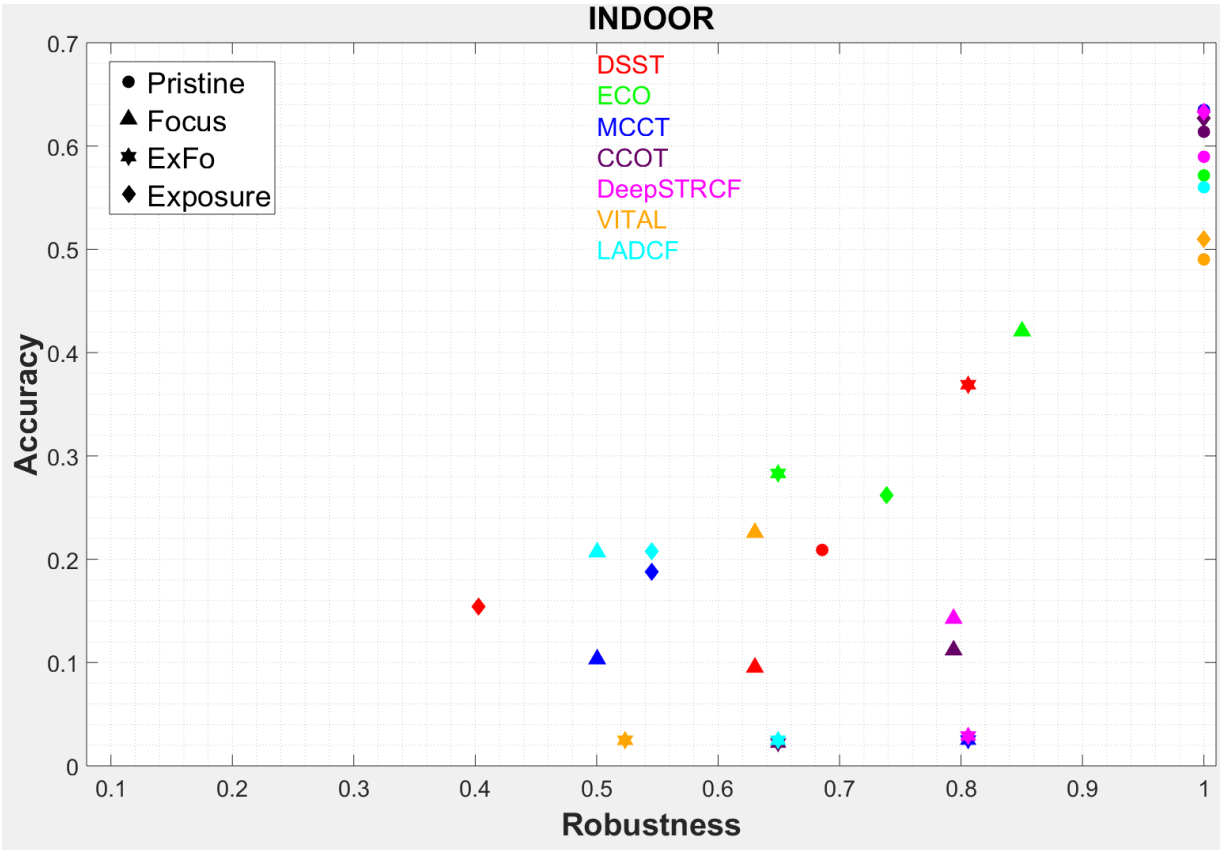


Figure 11: A-R plot for VOT in an indoor environment with pristine and distorted videos with the same activity.

We carried out one benchmark of the trackers previously presented in the AD-SVD dataset. Figures 5, 6, and 7 present examples of these test sequences, where the difference in image

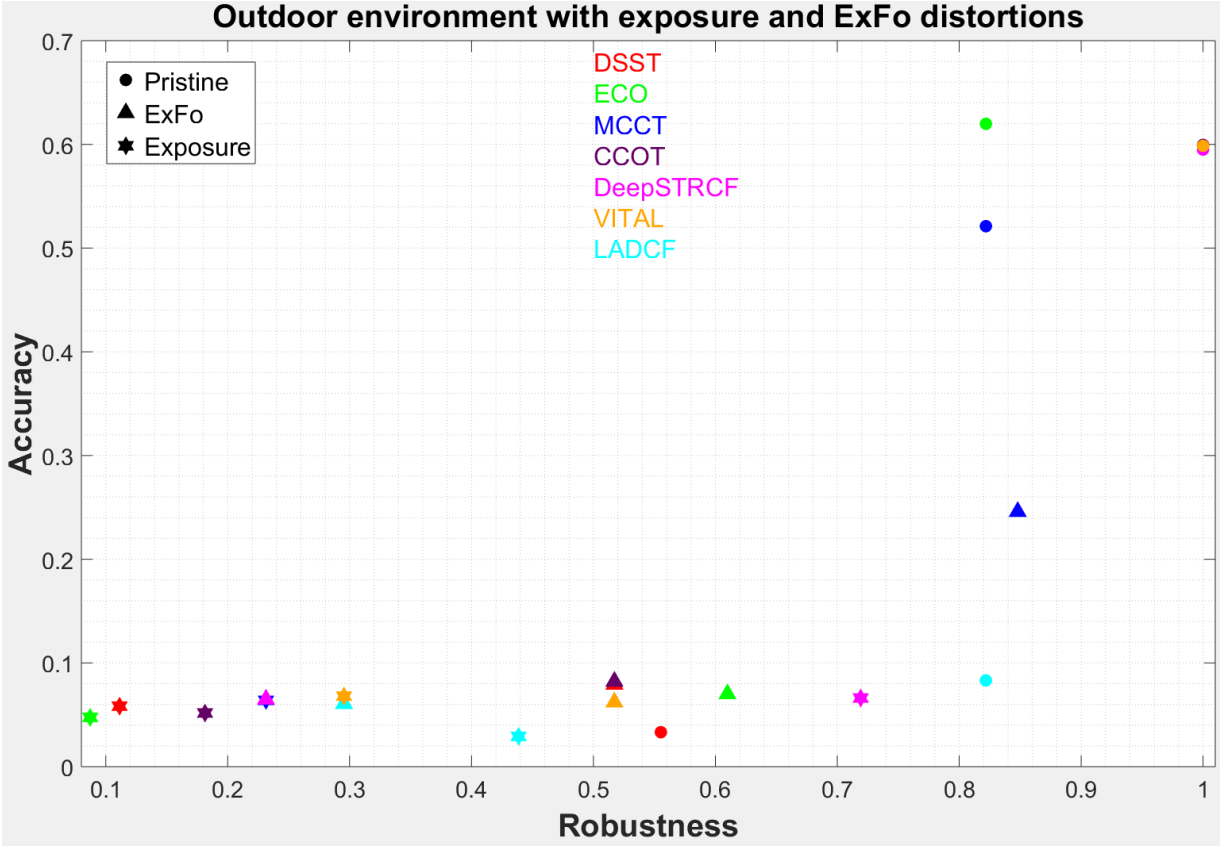


Figure 12: A-R plot for VOT in and outdoor environment with pristine and distorted videos with the same activity.

quality between the pristine and distorted image can be identified. It can be noticed that, even for the human observer, it is difficult to distinguish the people or objects that intervene in the scene in some distorted videos.

Figures 11 and 12 show the results of seven selected trackers in indoor and outdoor environments. All videos contained the same activity and were recorded using the same cameras in the same physical space. The only changing aspect is the distortion type. We can see that the best result is achieved with pristine videos in both indoor and outdoor environments. Nonetheless, the outdoor environment is more challenging for trackers, possibly because of changes in illumination and scene depth. In both environments, the distortion that most severely affected the trackers was the exposure time. Furthermore, the accuracy decreased severely and consistently for all distortions and pristine videos in the outdoor environment. The tracker that obtained the best results in the indoor environment was DeepSTRCF. In the outdoor environment, the most accurate tracker was the MCCT, and the most robust tracker

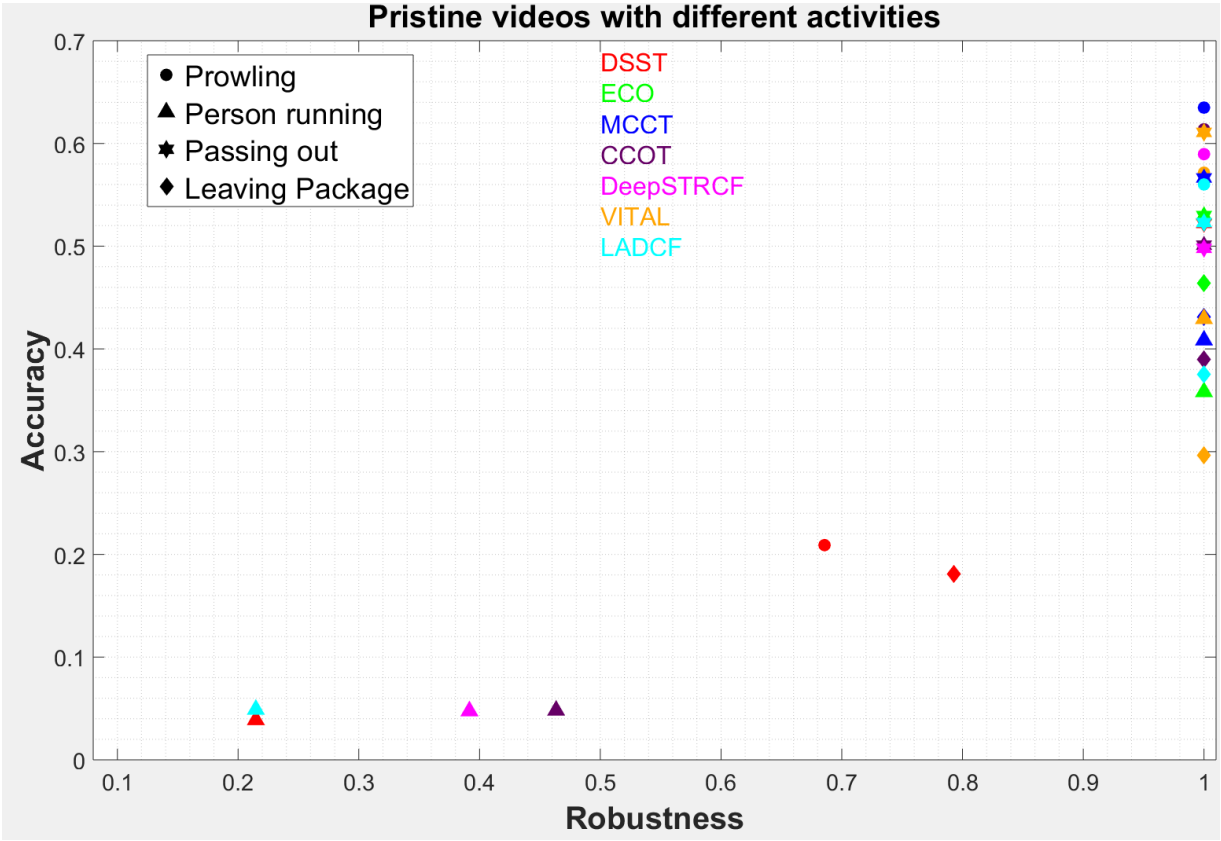


Figure 13: A-R plot for VOT within an indoor environment in pristine videos (same conditions for all) and different activity.

was DeepSTRCF. We tested the trackers with pristine and distorted videos that contain different activities to evaluate VOT in realistic scenarios.

In the first scenario, we tested similar activities with pristine videos, as shown in Figure 13. These results demonstrate that the visual content does not severely affect the tracker's robustness in pristine videos. In contrast, in videos with distortions, the tracker performance (accuracy and robustness) changes significantly and is highly dependent on the visual content. In the second scenario, we tested four activities (prowling, leaving package person, a person running, and passing out) on videos from the same camera, with exposure distortion, as shown in Figure 14. In general, the most challenging video for the trackers was a running person, possibly due to fast changes in the object position and the low FPS used (10 FPS). Considering the overall performance in all 15 scenes containing all distortions, environments, and activity, the most accurate tracker was DeepSTRCF, and the most robust was VITAL. However, these performances are far from the accuracy of the pristine videos. These results

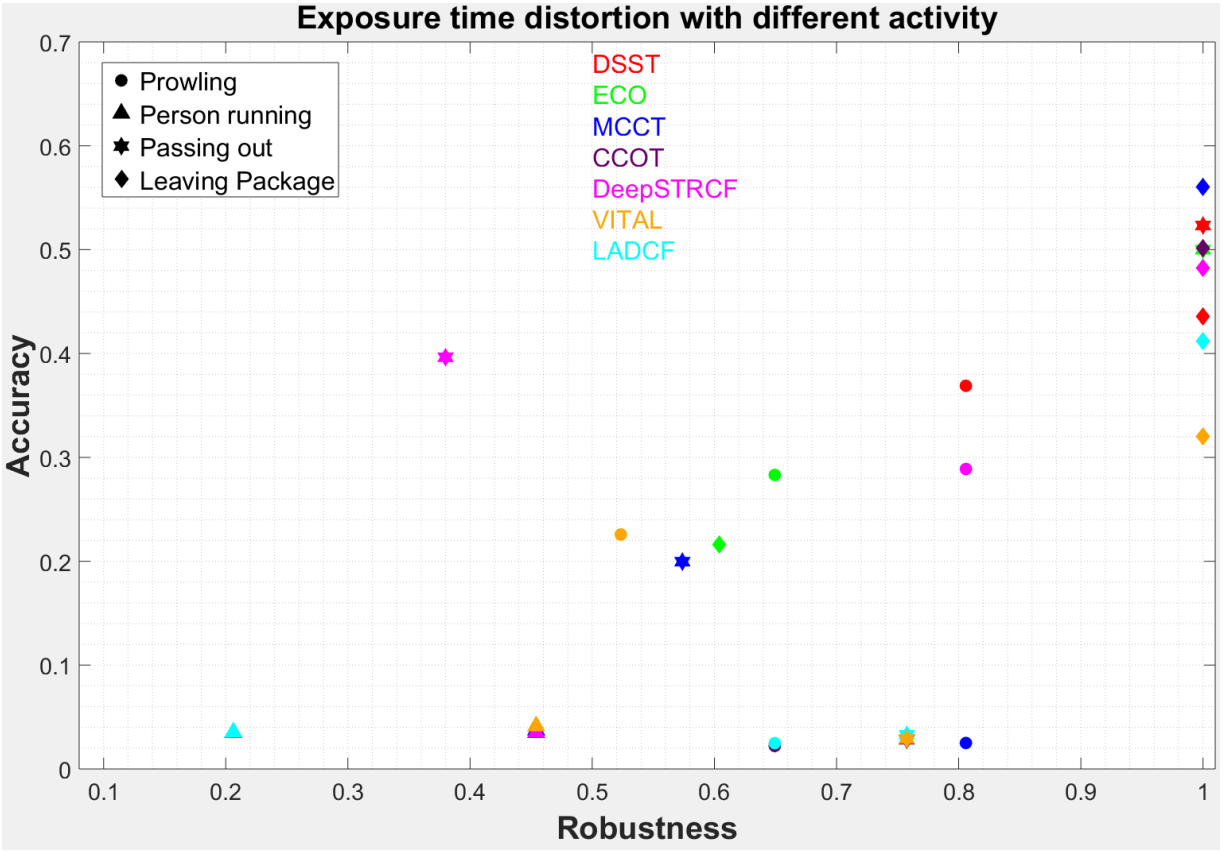
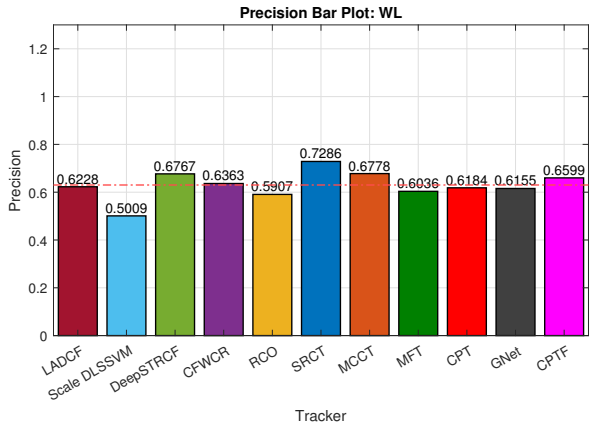


Figure 14: A-R plot for VOT with exposure time distortion in the same level and different activity in video.

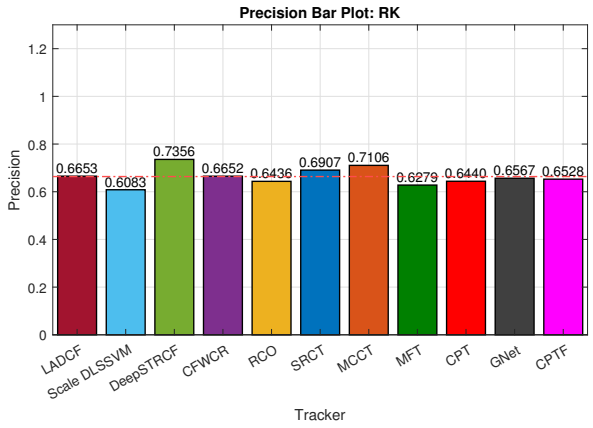
highlight the necessity of future VOT methods to improve the performance of videos with authentic distortions.

Impact of Video Surveillance Activities on VOT Performance

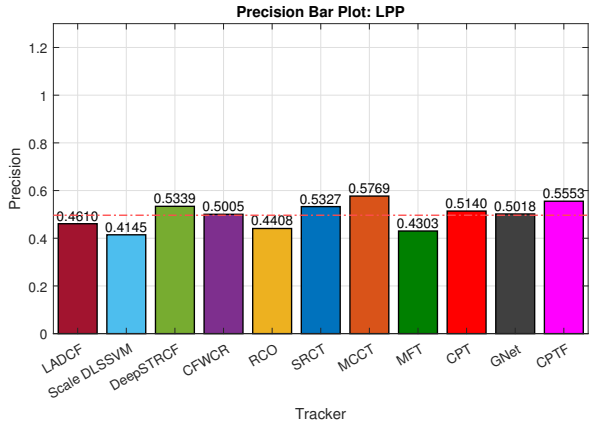
Figures 15 and 16 show the precision of each tracker with some surveillance activities [11]. A higher precision indicates that a tracker can follow an object with better estimates ($overlap > 50\%$). The least complex activity was RK, and the most complex activity was PO. The three better algorithms are as follows: 1) **DeepSTRCF** (RK, PPP), 2) **SRCT** (WL, PR, FG), and **MCCT** (LPP, PW, PO). The results show that the characteristics of surveillance activities affect the accurate tracking of an object.



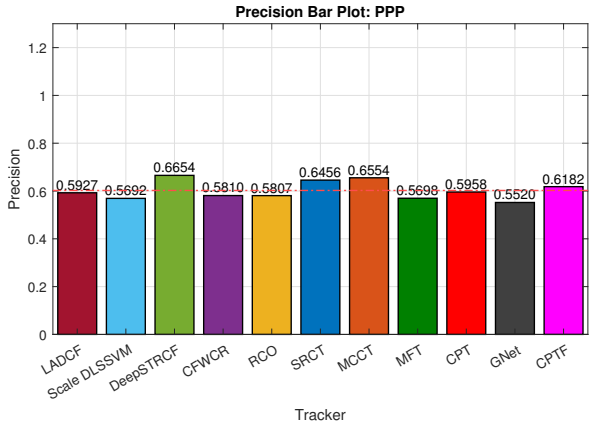
(a) Walking (WL)



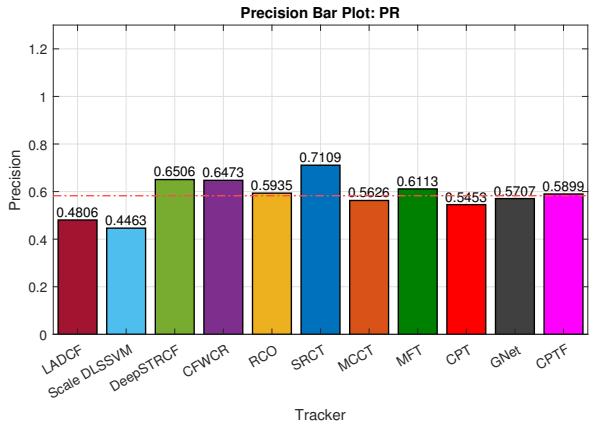
(b) Robbery with Knife (RK)



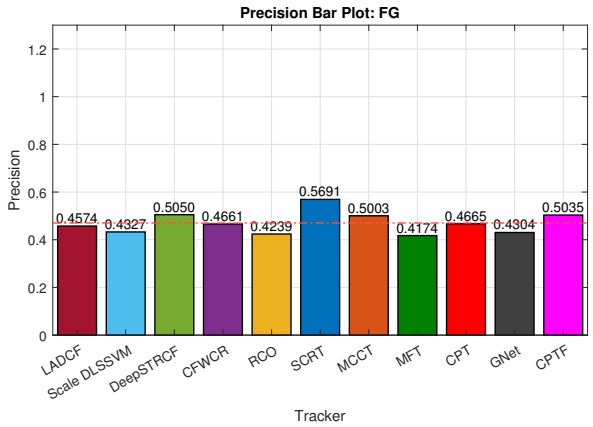
(c) Leaving Package in Public Place (LPP)



(d) Person pushing people

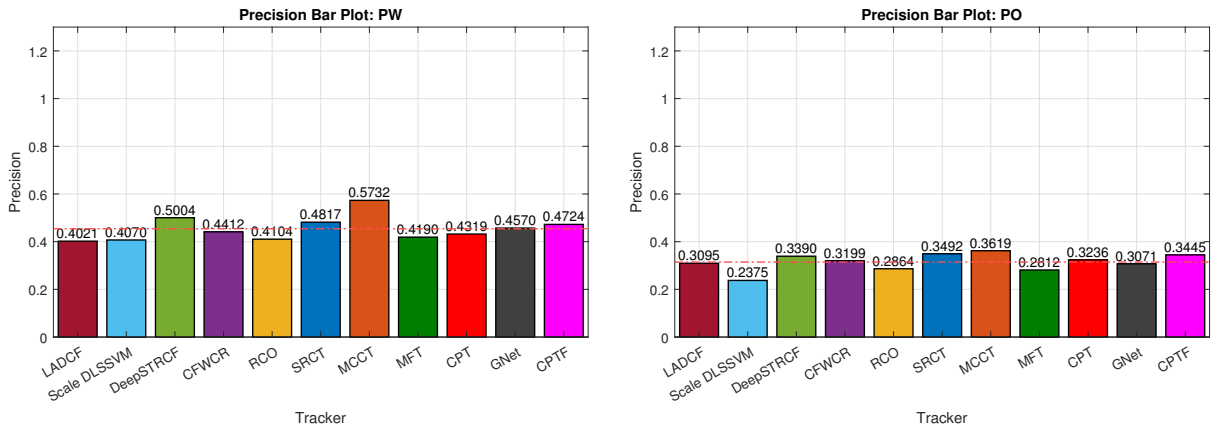


(e) Person running



(f) Fighting in Group (FG)

Figure 15: Trackers Precision with each Surveillance Activity.



(a) Prowling (PL)

(b) Passing Out (PO)

Figure 16: Trackers Precision with each Surveillance Activity.

Conclusions

We carried out an analysis of 11 state-of-the-art trackers highly ranked in the 2017 and the 2018 VOT challenges [108, 109]. The most innovative aspect of the present study is based on the database used. This chapter introduces AD-SVD, which models three levels of severity in 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. It involves 4476 videos that are affected by in-capture distortions produced by exposure time and defocus variations in challenging indoor and outdoor scenarios. AD-SVD contains real-world surveillance scenes such as people walking alone, meeting, fighting, passing out, leaving a package in a public place, prowling, and being robbed. In this way, AD-SVD can be seen as a solid starting point for studying the influence of distortions on video tracker performance.

This chapter concludes that in-capture distortions severely affect the performance of state-of-the-art trackers. As expected, the trackers exhibited the best performance for the pristine videos. Therefore, the results reflect the poor performance of the trackers owing to distortions such as underexposure and defocus. In practice, no specific type of distortion consistently generated the worst performance in all scenes and did not affect all trackers in the same way. Hence, the design and construction of a robust tracker for these distortions will be addressed in Chapter 5, proposing algorithms relying on perceptual features to compensate

for the impairments produced by these distortions.

4 Non-Reference Video Quality Assessment using 3-D Deep Features

Every day, millions of videos are shared and spread on platforms such as YouTube, Netflix, and Hulu. Cisco estimates that video traffic will be 82 percent of all Internet traffic (business and consumer) by 2022, up from 75 percent in 2017. Because of the high availability of smartphones, many of these videos are recorded by regular users who distort these videos with impairments such as artifacts, color, exposure, focus, sharpness, and stabilization caused by hardware limitations. Users do this because of a lack of knowledge about the generation of professional-quality videos. Natural videos often contain in-capture distortions that affect the video quality perceived by humans. Video streaming and camera manufacturers are keen to understand the influence and presence of these distortions in natural videos. This quality prediction can be performed automatically using VQA algorithms. Nonetheless, one of the main challenges in VQA is video content dependency, which makes it difficult to generalize from a unique dataset. In this chapter, we present an NR-VQA method. The proposed method is based on 3-D perceptual deep features extracted from the C3D Deep Network. We extensively tested NR-VQA methods on four recent authentically distorted datasets to benchmark and compared our proposed methodology. Similarly, we designed a neural network to classify an image as containing or no exposure distortion, allowing the determination of the exposure distortion presence in videos.

4.1 Convolutional Neural Network to identify videos with exposure distortion

We implemented and tested one CNN applied to VQA in the tracking task. To design and train the network, We built a computer architecture from the ground up. We used the GPU Titan XP, which contains 3840 NVIDIA® CUDA® cores running at 1.6 GHz and packs 12 TFLOPS of processing. The Titan XP has 12 GB of GDDR5X memory running at over 11 Gbps. Similarly, we deploy a CPU I7 8700K, with 14 nm technology and a clock frequency of 3.7 GHz, with six cores and 12 threads. We implemented 40 GB of RAM DDR4-2666 in this server. The motherboard used was a Z-370 with a capacity of two GPU Titan XP. For the power supply unit (PSU), we used an EVGA PLATINUM, 1200 W ECO. We installed a driver 390.25 of NVIDIA, Cuda 9.0, TensorFlow 1.7, and Keras 2.2.2. The interested reader can find one complete

installation guide for these libraries and frameworks at https://tinyurl.com/yd6cj6wj. We used the images/frames of the proposed AD-SVD dataset. We extracted 8600 images with exposure distortion and 8600 pristine images. An example of a pristine and distorted frame is shown in Figure 28. The videos used to train the CNN contain the same scene and activity, only differentiating the distortion level. The frames were randomly assigned to the test or training sets. In this manner, location or activity does not have a significant impact on the results. The inputs to the CNN are images of FHD resolution (1920 $\times$ 1080).

(a) Pristine frame of person Prowling

(b) Exposure distorted frame of person prowling in the same scene

Figure 17: Examples of pristine and exposure distorted frame used to train the CNN to classify video.

Before CNN training, we conducted the following pre-processing steps:

- To convert to floating-point arithmetic, the computational complexity of the NN is reduced.
- To scale data to a range [0,1].
- To replace label data with one-hot encoded versions.
- To scale samples to a 2D grid, one line per image.
- To scale sample data to 4D tensor using the channel last convention. It puts the data into the structure expected by the convolution layer that is located in the shallow layer of our convnet.

Before inputting the network, the images were reduced to 128 $\times$ 128 pixels. Additionally, the name was changed to the pictures, following this convention: the first three letters of the class (pri or exp), next to a dot, and then the consecutive number of the image (for example,

Table 6: Results of CNN classification for exposure distortion in images, for various size training and test datasets.

Train-Test %	Epoch 1 loss - acc	Epoch 3 loss - acc	Epoch 5 loss - acc	**Testing set** loss - acc
0.1- 99.9	0.8484-0.5294	3.2954 - 0.6471	0.5868 - 0.6471	0.8193 - 65.9330
1 - 99	2.0625-0.5200	1.7992 - 0.5429	0.3747 - 0.8857	0.3085 - 86.8178
5 - 95	3.0105-0.5463	0.2925 - 0.8629	0.2028 - 0.9246	0.1627 - 97.2466
10 - 90	2.3549-0.6543	0.1579 - 0.9611	0.1061 - 0.9783	0.0891 - 100.00
20 - 80	1.0635-0.8058	0.1016 - 0.9703	0.0430 - 0.9971	0.0359 - 99.8429
50 -50	0.3849-0.9074	0.0445 - 0.9921	0.0124 - 1.0000	0.0121 - 100.000
80 - 20	0.2660-0.9380	0.0135 - 1.0000	0.0057 - 1.0000	0.0053 - 100.000

pri.234). In the last layer, we used the softmax function as the activation unit. We used an online algorithm, the stochastic gradient descent (SGD), as the optimizer because it does not require storing samples. Given that we have just two categories, we use one output to decide between them (setting it to a value near zero for one class and a value near one for the other). The batch size was fixed at 128. We used the following CNN architecture for the direct training approach: Conv-768, Conv-384 with stride 2, and FC-2. All the convolutional layers were configured to use 33 filters, using zero-padding to preserve the spatial size.

The number of parameters in the proposed Convolutional Neural Network (CNN) architecture is approximately 40 million, much lower than those of AlexNet and ResNet50. This baseline model was trained using the binary cross-entropy loss. We iterated the training over five epochs. When we tried to increase the input size of the image, the GPU collapsed and returned an error, probably due to memory restrictions. Hence, 128×128 was selected as the ideal size for image input to the CNN. The trained CNN can generate optimal results, as shown in Table 6. This CNN has a high performance, even when the training set size is severely reduced. This high performance indicates that the features extracted by the CNN are representative of the distortions presented in the videos. This result is a positive outcome because it is well known that the success of a deep neural network depends on the training set size.

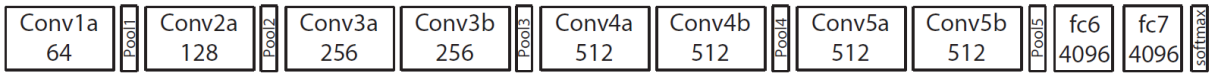


Figure 18: C3D Architecture [12].

4.2 NR-VQA Proposed Method

Convolutional Neural Network

In all the experiments, we used the C3D network [12] to extract spatio-temporal features. In [12] the authors showed that these 3-D spatio-temporal features, along with a simple linear classifier, can yield good performance on some video analysis tasks, such as action recognition [173], action similarity labeling, scene classification, and object recognition. C3D uses a $3 \times 3 \times 3$ convolution kernel for all layers and takes full video frames as input, with no pre-processing stage. C3D performs 3D convolutions and 3D pooling to propagate temporal information across all layers, allowing access to the model temporal information.

C3D has five convolution layers and five pooling layers (a pooling layer immediately follows each convolution layer), two fully connected layers, and a softmax loss function to predict action labels. The number of filters for convolution layers from 1 to 5 are 64, 128, 256, 256, 256, respectively, as shown in Fig. 18. C3D resizes all video frames to 128×171 pixels [12]. In one experiment, we modified the C3D original architecture to input the convolutional layers videos split into 16 frames clips with an 8-frame overlap between two consecutive clips, obtaining more feature vectors per video and improving the features/data ratio. C3D has 17.5 million parameters in the fully connected layers. The authors tested four distinct architectures by varying the size and depth of the kernel in each architecture. The number of parameters in the convolutional layers was different for each architecture. Still, the variation is minimal compared with the 17.5 million parameters in the fully connected layers, which is the same for all architectures [12]. We extracted feature vectors with dimensions of 50175 from the fifth convolutional layer (conv5b) and 4096 from the fully connected layer (fc6).

Dataset

Several VQA databases have been created in recent years [102, 104]. We used the LIVE-Qualcomm Mobile In-Capture Video Quality database proposed by [15] because it

contains videos with authentic distortions. The original videos from LIVE-Qualcomm were in the YUV420 format. We converted all videos to the AVI uncompressed format. This new format resulted in videos with an average size of 2.8 Gigabytes, duplicating its original size. This type of conversion minimizes the information lost by compression, thus avoiding the addition of different post-capture distortions to the videos. However, a drawback is the enormous size of these videos, which increases the processing time to extract the features from the C3D layers. Nonetheless, we believe that this additional work does not need to introduce further post-capture compression distortion, affecting the final results. The videos in the LIVE-Qualcomm dataset had an average duration of 15 seconds and a rate of 30 Frames per Second (FPS). Each video had approximately 450 frames, but not all videos had the same duration; some had fewer than 400 frames. We discarded one video as an outlier (only 360 frames). Therefore, we used 207 of the 208 videos from the LIVE-Qualcomm dataset.

Pre-processing

We pre-processed each video using a transformation to the YCbCr color space because of the rough correspondence between the YCbCr components and visual attributes. YCbCr is one of the two primary color spaces to represent digital videos (along with RGB). The distinction between YCbCr and RGB is that YCbCr represents color as brightness (Y) and two color difference signals (CB, Cr), while RGB means color as red, green, and blue [174,175]. Likewise, YCbCr is less redundant than RGB, supporting the CNN encoding capabilities. In the YCbCr color space, lightness changes that affect image contrast (but not color) are easily accessed. In addition, YCbCr is intended to take advantage of the human color-response characteristics. We believe that this can facilitate the detection of certain distortions. We also pre-processed each video using a statistical image model based on Mean Subtracted Contrast Normalized (MSCN) coefficients. MSCN coefficients have characteristic statistical properties that are changed by distortions. It is known that quantifying these changes makes it possible to predict the distortion affecting an image and its perceptual quality [74].

SVR training

SVR has been successfully applied to VQA problems [67, 76, 176–181]. In some VQA frameworks, one usual way to obtain a quality score is to train the SVR with the proposed

features. SVR can handle high-dimensional data comparable to the length of the feature vector in the output of C3D convolutional and fully connected layers. We utilized the MATLAB Machine Learning Toolbox to implement SVR with a radial basis function (RBF) and Gaussian kernel. We determined the optimal model parameters of SVR via 10-fold cross-validation. The aim of minimizing the error in the validation data guided our selection of the model. We used a random, non-overlapping train and test split with 80% of the sequences for training and 20% for testing in each test case. To avoid any bias due to data division, we randomly split the dataset 100 times. The median PLCC, SROCC results, and their standard deviations for each test case are reported in Tables 18-19. A higher value of each metric shows a better correlation between the MOS and the proposed VQA method scores.

To evaluate the performance of the VQA methods, we adopted two criteria, namely the Pearson Linear Correlation Coefficient (PLCC) and the Spearman Rank Ordered Correlation Coefficient (SROCC) between ground-truth MOS and predicted MOS. The PLCC is a measure of the strength of the linear dependence between two variables. The SROCC measures the prediction monotonicity because it operates only on the ranking of the data points and ignores the relative distance between them. The absolute value of SROCC describes the intensity of the monotonic relationship [182]. The feature vectors have a size of 4096×1 for the fully connected layer fc6 and 50176×1 for the fifth convolutional layer conv5b. These vectors compose the matrix F_{input}, which is the input matrix for training and testing the SVR machine. One of the deployed approaches uses Average Pooling (AP) by averaging all columns from the matrix F_{input} (m is the number of features in the output of the CNN layer), where the matrix of size $m \times n$ is converted in $m \times 1$, to represent each video as a single feature vector.

4.3 Benchmarking of VQA State-of-the-art algorithms on VQA datasets

To evaluate the performance of the proposed VQA method compared with the state-of-the-art NR-VQA methods, we performed an extensive benchmark evaluation. In this benchmark, we evaluated the algorithms BRISQUE [74], FRIQUEE [90], TLVQM [70], QAWV [183, 184], VQPooling NIQE [71], Average Pooled NIQE [71], VIIDEO [65], and 3D-MSCN [45], on the VQA NR datasets LIVE-Qualcomm [15], LIVE-VQC [185], KonVid-1K [186] and CVD2014 [104]. We should note that these datasets are all important for this research because they contain authentically distorted videos.

We used the VQPooling method along with the NIQE algorithm. VQPooling was proposed based on the hypothesis that the worst local quality scores (both spatial and temporal) affect the overall subjective perception of quality more significantly than do the regions of good quality [187, 188]. In this method, the quality scores of the frames in the video were classified into two groups [188]. These groups are the higher and lower quality scores, obtained using k-means clustering [189]. These two groups are combined to get an overall quality prediction for the entire video sequence [190, 191]. VQPooling adaptively emphasizes "worst" scores along the spatial and temporal dimensions of the video sequence and considers the perceptual effect of large-area cohesive motion flow [188].

Table 7: Properties of the 4 selected videos to measure execution time in the VQA Metrics

Video Name	Number of Frames	Resolution	Duration of Video (s)	Data rate Kbps
CVD-Test02-City-D01.avi	460	640x480	15	14747
KoNViD-1k-5927908371	240	960x540	8	9670
LIVEVQC-ABP-002-A069	300	1920x1080	10	9519
Focus-Qualcomm-1006-ComplexTrain	443	1920x1080	14	1493034

Table 8: Execution times for the VQA methods selected tested in the 4 selected videos. All the times are in seconds, as the summation of all the frames in the corresponding video. The fastest times per video are in bold.

Videos	BRISQUE	FRIQUEE	TLVQM	QAWV	NIQE	VIIDEO
CVD-Test02-City-D01.avi	130.82	12513.88	**102.32**	317.40	109.14	333.32
KoNViD-1k-5927908371	**80.35**	9917.54	84.82	237.68	91.20	209.14
LIVEVQC-ABP-002-A069	**261.33**	57952.82	411.99	1423.54	499.99	1015.37
Focus-Qualcomm-1006-ComplexTrain	**411.45**	82129.32	765.70	2086.84	691.284	1380.22

We executed these on one randomly elected video per dataset to measure the time performance of the analyzed VQA methods. Table 7 presents some of the main features of the four videos selected. The times of each VQA metric to process the entire video are shown in Table 8. These times were measured using a laptop with a processor I5-7200U, 500 Gb SSD disk, and 16GB DDR4 RAM. We note that the FRIQUEE [90] method was by far the one with the most extensive execution times. This longer execution time is due to the extraction of video characteristics through four layers and the prediction based on an SVR. The dimensions of the feature map for each frame were 560. In contrast, QAWV [183, 184], TLVQM [70], BRISQUE [74], and NIQE [71] metrics reduce the sizes of the feature maps and achieve a faster calculation between vectors. Overall, BRISQUE is the fastest VQA method analyzed in this study; times for BRISQUE is

reduced probably because it does not require transformation to another coordinate frame (DCT, wavelet, etc.) [74]. In general, faster methods also use elements from parallel frameworks such as CUDA, MPI, or MATLAB Parallel Computing Toolbox [191].

Tables 9, 10, and 11 show the median PLCC, SROCC and RMSE, as well as the standard deviation of the implemented VQA methods in the state-of-the-art VQA datasets [191]. We executed the algorithms on all the videos of the four VQA datasets. We note that the FRIQUEE method outperforms the other VQA algorithms in LIVE-Qualcomm. However, low speed is not the best option for conditions where a high-processing FPS is required. In other VQA datasets, QAWV and TLVQM obtained the best scores. QAWV performs better in KonVid-1K and CVD2014, whereas TLVQM obtains the best score in LIVE-VQC. Overall, the best method on the four datasets, measured as the sum of individual PLCC scores, was TLVQM, which was 2.9.

Table 9: Median PLCC ± Standard Deviation of state-of-the-art methods, evaluated on the four NR-VQA authentic in-capture distortions datasets. The 3D-MSCN method was not evaluated in LIVE-Qualcomm, due to computational resources limitations. The best scores in each dataset is marked in bold.

VQA Method	LIVE-Qualcomm	LIVE-VQC	KoNViD-1k	CVD2014
BRISQUE	0.5060 ± 0.1190	0.5640 ± 0.0750	0.5930 ± 0.0410	0.4580 ± 0.1120
FRIQUEE	**0.6108 ± 0.1229**	0.6500 ± 0.1545	0.6465 ± 0.1226	0.8105 ± 0.0663
TLVQM	0.5804 ± 0.1876	**0.7793 ± 0.0439**	0.7327 ± 0.0495	0.7240 ± 0.0883
QAWV	0.5400 ± 0.2215	0.7500 ± 0.0755	**0.7823 ± 0.0828**	**0.8303 ± 0.2084**
VQPooling NIQE	0.3799 ± 0.0314	-0.0308 ± 0.0179	-0.2078 ± 0.0175	0.1641 ± 0.0363
Average-pooled NIQE	0.4606 ± 0.0284	-0.0031 ± 0.1026	-0.5276 ± 0.0114	0.312 ± 0.0299
VIIDEO	0.1102 ± 0.0321	0.1038 ± 0.0233	0.3048 ± 0.0138	0.1163 ± 0.0310
3D-MSCN	X	0.2807 ± 0.1591	0.3205 ± 0.3786	0.4502 ± 0.2916

Additionally, we performed tests separating the videos of the LIVE-Qualcomm dataset by distortion type. In this way, we can evaluate the six in-capture distortions contained in LIVE-Qualcomm: artifacts, color, exposure, focus, sharpness, and stabilization. We organized the data into two sets: 80% for training and 20% for the test set. Two videos of the same scene would not exist in these sets, even if recorded with a different device. In this manner, we ensured that the test set did not contain information already incorporated in the training set. The results of these experiments are presented in Tables 12, 13, and 14. We note that the TLVQM method performs well in terms of focus, sharpness, and stabilization distortions. In contrast, the FRIQUEE method [90] was superior in videos with exposure distortion, together with experiments containing all videos (the six distortions) of the LIVE-Qualcomm dataset [191].

Table 10: Median SROCC ± Standard Deviation of state-of-the-art methods, evaluated on the four NR-VQA authentic in-capture distortions datasets. The 3D-MSCN method was not evaluated in LIVE-Qualcomm, due to computational resources limitations. The best scores in each dataset is marked in bold.

VQA Method	LIVE-Qualcomm	LIVE-VQC	KoNViD-1k	CVD2014
BRISQUE	0.4940 ± 0.1030	0.5790 ± 0.0560	0.6260 ± 0.0380	0.5110 ± 0.1060
FRIQUEE	**0.5789 ± 0.1067**	0.6492 ± 0.0485	0.6575 ± 0.0389	0.7879 ± 0.0628
TLVQM	0.5597 ± 0.1446	**0.7843 ± 0.0376**	0.7469 ± 0.0314	0.7299 ± 0.090
QAWV	0.4760 ± 0.2066	0.6996 ± 0.0845	**0.7757 ± 0.0932**	**0.8614 ± 0.2237**
VQPooling NIQE	0.3756 ± 0.0318	0.2155 ± 0.0204	-0.3064 ± 0.0122	0.3451 ± 0.0329
Average-pooled NIQE	0.4313 ± 0.0289	0.2864 ± 0.020	-0.5355 ± 0.0125	0.4872 ± 0.0321
VIIDEO	0.1381 ± 0.0359	0.0164 ± 0.0201	0.3068 ± 0.0141	0.0911 ± 0.0342
3D-MSCN	X	0.2026 ± 0.1519	0.3096 ± 0.1402	0.4045 ± 0.3243

Table 11: Median RMSE ± Standard Deviation of state-of-the-art methods, evaluated on the four NR-VQA authentic in-capture distortions datasets. The 3D-MSCN method was not evaluated in LIVE-Qualcomm, due to computational resources limitations. The best scores in each dataset is marked in bold.

VQA Method	LIVE-Qualcomm	LIVE-VQC	KoNViD-1k	CVD2014
BRISQUE	10.4450 ± 1.1350	14.0050 ± 0.8470	10.4500 ± 0.5010	19.5380 ± 3.9790
FRIQUEE	**9.7150 ± 1.4730**	12.9500 ± 4.4700	12.5015 ± 1.1811	12.7620 ± 1.8710
TLVQM	10.1660 ± 1.7120	**10.6828 ± 0.9583**	10.9580 ± 0.9238	15.4255 ± 2.1393
QAWV	11.6197 ± 2.2747	12.1337 ± 1.4093	**0.3990 ± 0.0578**	**12.3669 ± 3.7055**
VQPooling NIQE	19.5398 ± 0.3799	22.924 ± 0.3929	0.7404 ± 0.0085	27.614 ± 0.8151
Average-pooled NIQE	21.7613 ± 0.3903	72.1562 ± 23.0766	1.0673 ± 0.0099	51.0431 ± 0.7084
VIIDEO	12.0650 ± 0.2591	17.0255 ± 0.2279	0.6139 ± 0.0063	21.3166 ± 0.4247
3D-MSCN	X	17.9263 ± 2.1769	0.6558 ± 0.1885	18.3050 ± 3.5347

Finally, we performed experiments testing the VQA state-of-the-art algorithms on videos of the LIVE-Qualcomm dataset, separated according to the capture smartphone. Eight devices are used in capturing videos: Samsung S5, Samsung S6, Samsung Note 4, HTC One VX, iPhone 5S, LG G2, Nokia Lumia 1020, and OppoFind 7. To avoid the redundancy of information in the performance of the SVR, we eliminated duplicate videos of the same scene. Therefore, the dataset was organized in such a way that each device was represented by the same number of videos, resulting in balanced class sets (Galaxy S5=8, Galaxy S6=8, HTC One VX=8, iPhone 5S=8, LG G2=8, Lumia 1020 = 3, Note 4 =4, OppoFind7 =7). This division allowed us to obtain a total of 54 videos on which we executed the VQA algorithms. The results are presented in Tables 15, 16, and 17. Overall, the best VQA method for the different devices was TLVQM.

4.4 Performance of the Proposed VQA Method

We obtained 73 pristine videos from several sources [35, 101, 192] and extracted the C3D deep feature vectors from the fully connected layer fc6. Thereafter, we studied these feature vectors with the equivalent feature vectors of distorted videos from the LIVE-Qualcomm dataset. Figure 19 shows the results of Principal Component Analysis projection (PCA). Diverse clusters occur because of the pristine and various types of distorted videos. The videos were converted to the YCbCr color space and processed using the C3D CNN. The values in Tables 18 and 19 justify our choice of the YCbCr color space, which plays a significant role in enhancing the video quality predictions. Therefore, we take the feature vector from the output of the sixth fully connected layer fc6, using the advancement of eight frames in each block of frames to feed the CNN.

The LIVE-Qualcomm dataset contains videos acquired from four different smartphones per unique scene. We evaluated the impact of this information redundancy in the SVR performance. To do this, we used the method with higher overall performance (Fc6_YCbCr_8Frames_AP) and deleted duplicated videos for the same scene. The dataset was organized such that each smartphone device was represented by approximately the same number of videos, with a total of 54 videos (Galaxy GS5=8, Galaxy GS6=8, HTC One VX=8, iPhone 5S=8, LG G2=8, Lumia 1020 = 3, Note 4 =4, Oppo Find 7 =7). The results of this test are summarized in the Fc6_YCbCr_8Frames_AP* method, as shown in Tables 18 and 19. It can be observed that the overall performance varied by only 0.0099. These results support the conclusion that the redundancy in information per scene is not a critical factor in the overall performance of the

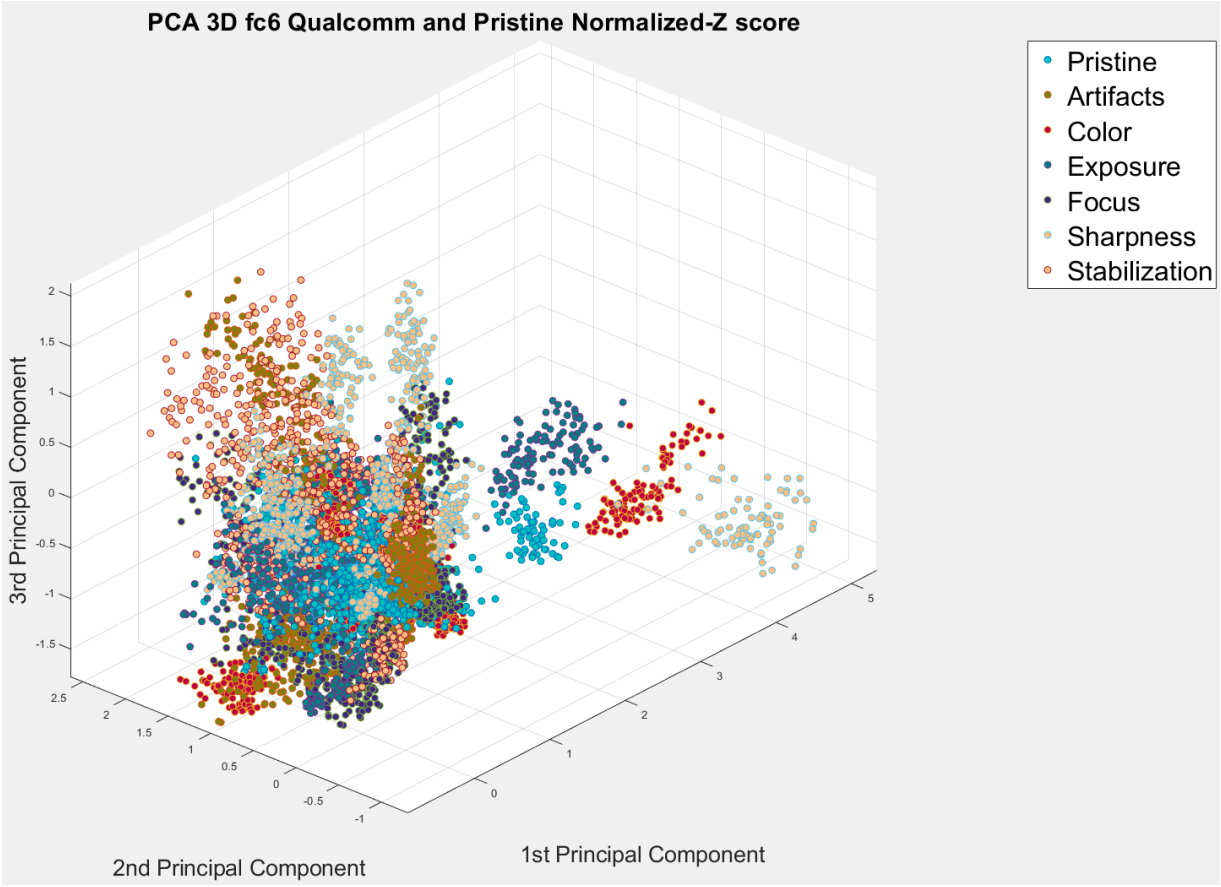


Figure 19: 3-D PCA of pristine and distorted videos from LIVE-Qualcomm dataset, with No Average Pooling NA (25 points represents one single video).

proposed VQA method when tested on the LIVE-Qualcomm dataset. In contrast, the results reported in [15] were tested using all videos, including those involved in the same scene.

For further analysis, we also calculated the R^2 correlation coefficient (values > 0 showed a linear correlation). Figure 20 shows a scatter plot of our results using the Fc6_YCbCr_8Frames_AP VQA method on one test sequence of 39 videos, compared with the Ground Truth MOS from LIVE-Qualcomm. Our proposed VQA method obtained the best overall performance (all distortions), achieving a PLCC correlation of 0.7749 ± 0.0884, outperforming the best performance reported in [15] by FRIQUEE [90] (PLCC = 0.7349). Furthermore, our VQA method Fc6_YCbCr_8Frames_AP obtained the best performance on videos with exposure distortion, achieving an SROCC of 0.7271 ± 0.3263. Similarly, another proposed method, Fc6_YCbCr_8Frames_NA, got the second-best results on exposure distortion, with an SROCC of 0.6562 ± 0.4141. By analyzing and separating the distortions and comparing them with the

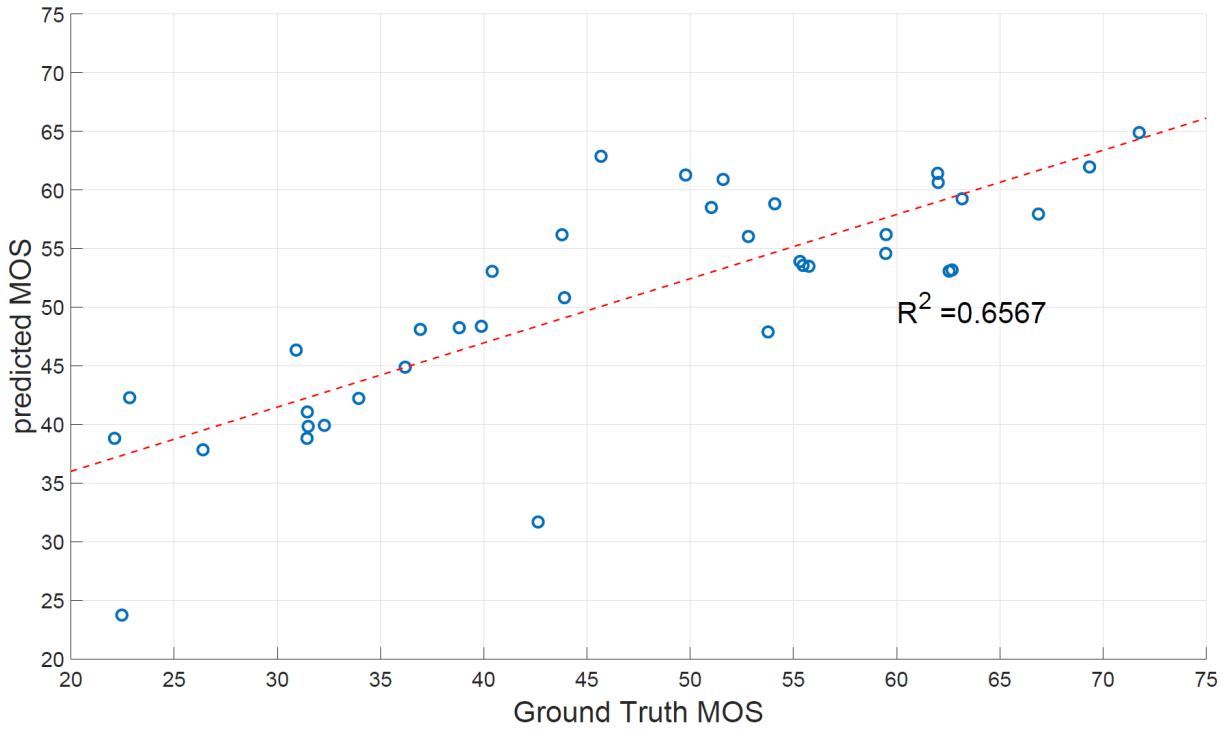


Figure 20: Scatter plot between Ground Truth MOS and Predicted MOS obtained with Fc6_YCbCr_8Frames_AP VQA proposed method. There are 39 videos in the test set. We reported the linear correlation coefficient $R^2 = 0.6567$, and plotted the least-squares line (red).

PLCC score, the Fc6_YCbCr_8Frames_NA method outperformed the competing approaches in color distortion. This method uses the features extracted from the first fully connected layer, with the input data converted to YCbCr format and eight frames of batch size separation.

Thereby, our best-performing proposed methods outperformed the best performance reported in [15] on videos affected by exposure distortion with V-BLIINDS [76], which obtained 0.64290 without informing standard deviation. The results for other methods (i.e., V-BLIINDS [76], VIIDEO [65], NIQE [71], BRISQUE [74], and FRIQUEE [90]), are reproduced here [15] because they also report on median PLCC, and SROCC values using tenfold cross-validation with 100 random training–validation–test splits and use the same dataset. Those methods that use Average Pooling (AP) require lower computational times by reducing the size of the matrix F_{input}, allowing accelerated execution of SVR training and testing.

Conclusions

This chapter evaluated seven state-of-the-art VQA NR metrics, training, and validating the proposed methods using four relevant video quality databases. These VQA datasets present a wide variety of video contents specifically targeted at user-generated content that is prone to capture artifacts, such as camera shake, over and under-exposure, among other authentic distortions. The results show that the TLVQM, QAWV, and FRIQUEE methods outperformed all the reference methods tested in terms of accuracy. The computational advantage of TLVQM is significant given its execution time and efficiency in calculating the objective scores.

Similarly, we proposed an NR VQA method, explicitly aimed at videos with natural distortions, such as color, artifacts, exposure, focus, sharpness, and stabilization. Our method is based on the convolutional neural network approach, using features extracted from several layers of CNN to feed one machine learning SVR model that produced an NR VQA model providing a high level of video quality prediction power. We extensively evaluated the perceptual quality prediction model, obtaining a final correlation of 0.7749 ± 0.0884 with Human Opinion Scores. It shows that it can achieve good video quality prediction, outperforming other state-of-the-art VQA leading models. This VQA method can accurately model the main real-world distortions and predict the quality of real-world videos. Based on findings from related research [61], we believe that our results can be improved by changing the architecture and training of the C3D CNN and using transfer learning to fine-tune the fully connected layer.

Table 12: Median PLCC ± Standard Deviation of state-of-the-art VQA methods tested in LIVE-Qualcomm dataset. The experiments were performed on sets according to the distortion contained. The best scores in each dataset are marked in bold.

VQA Method	Artifacts	Color	Exposure	Focus	Sharpness	Stabilization	All distortions
BRISQUE	0.4177 ± 0.2951	0.4131 ± 0.328	0.4769 ± 0.2931	0.7295 ± 0.2546	0.574 ± 0.2825	0.4021 ± 0.4027	0.5518 ± 0.0974
FRIQUEE	0.7086 ± 0.2300	0.312 ± 0.3850	**0.6913 ± 0.2626**	0.6471 ± 0.3683	0.6556 ± 0.2926	0.6783 ± 0.3313	**0.6179 ± 0.1056**
TLVQM	0.6122 ± 0.2513	**0.4992 ± 0.2201**	0.4396 ± 0.2063	**0.8027 ± 0.2003**	**0.8263 ± 0.1981**	**0.7619 ± 0.2208**	0.6177 ± 0.1409
QAWV	**0.7530 ± 0.2048**	0.3552 ± 0.1828	0.5364 ± 0.3645	0.4024 ± 0.2757	0.7671 ± 0.0619	0.2975 ± 0.1116	0.5600 ± 0.2715
VQPooling NIQE	0.2426 ± 0.0582	-0.0344 ± 0.1282	0.0893 ± 0.0968	0.4693 ± 0.0809	0.5281 ± 0.0762	0.355 ± 0.0718	0.3799 ± 0.0314
Average-pooled NIQE	0.5391 ± 0.0721	0.1488 ± 0.1928	0.106 ± 0.0492	0.4901 ± 0.073	0.3885 ± 0.1232	0.5003 ± 0.0958	0.4606 ± 0.0284
VIIDEO	0.012 ± 0.0903	-0.0861 ± 0.142	0.1857 ± 0.0994	0.1791 ± 0.0834	0.1645 ± 0.0776	-0.1567 ± 0.075	0.1102 ± 0.0321

Table 13: Median SROCC ± Standard Deviation of state-of-the-art VQA methods tested in LIVE-Qualcomm dataset. The experiments were performed on sets according to the distortion contained. The best scores in each dataset are marked in bold.

VQA Method	Artifacts	Color	Exposure	Focus	Sharpness	Stabilization	All distortions
BRISQUE	0.4290 ± 0.3300	0.3210 ± 0.3140	0.3930 ± 0.3210	0.7500 ± 0.2720	0.5360 ± 0.2950	0.3570 ± 0.3680	0.5090 ± 0.1030
FRIQUEE	0.6786 ± 0.2419	0.335 ± 0.3978	**0.6071 ± 0.2985**	0.6571 ± 0.3930	0.6429 ± 0.2784	0.6071 ± 0.2997	**0.5805 ± 0.1025**
TLVQM	0.5714 ± 0.2241	0.4286 ± 0.2561	0.3929 ± 0.2143	**0.7500 ± 0.2295**	**0.7857 ± 0.2233**	**0.6786 ± 0.1808**	0.5737 ± 0.1484
QAWV	**0.7727 ± 0.2236**	**0.4286 ± 0.1117**	0.5238 ± 0.3295	0.3939 ± 0.2029	0.6429 ± 0.0905	0.4048 ± 0.1027	0.506 ± 0.2470
VQPooling NIQE	0.2876 ± 0.0776	-0.179 ± 0.1117	0.1117 ± 0.0968	0.3562 ± 0.1052	0.3892 ± 0.0932	0.3041 ± 0.083	0.3756 ± 0.0318
Average-pooled NIQE	0.3826 ± 0.0867	-0.1053 ± 0.1192	0.2884 ± 0.0823	0.4246 ± 0.0818	0.5775 ± 0.0717	0.4095 ± 0.0737	0.4313 ± 0.0289
VIIDEO	-0.099 ± 0.0779	-0.1107 ± 0.1299	0.38 ± 0.083	0.0765 ± 0.0928	0.1932 ± 0.1014	-0.0632 ± 0.0908	0.1381 ± 0.0359

Table 14: Median RMSE ± Standard Deviation of state-of-the-art VQA methods tested in LIVE-Qualcomm dataset. The experiments were performed on sets according to the distortion contained. The best scores in each dataset are marked in bold.

VQA Method	Artifacts	Color	Exposure	Focus	Sharpness	Stabilization	All distortions
BRISQUE	11.5410 ± 2.8360	**8.3600 ± 1.9440**	10.3390 ± 2.349	**9.6610 ± 2.8450**	9.7060 ± 2.1110	8.5540 ± 2.6090	10.0080 ± 0.9250
FRIQUEE	9.3230 ± 2.5050	10.5390 ± 2.3210	9.2610 ± 2.2500	11.0690 ± 3.1320	8.5310 ± 2.2810	7.4520 ± 2.4520	**9.8350 ± 97289**
TLVQM	10.9150 ± 3.7840	9.4780 ± 2.9300	11.8430 ± 2.5440	9.8290 ± 3.3420	**8.3530 ± 2.7280**	7.3130 ± 2.0190	10.2060 ± 1.5670
QAWV	**8.5378 ± 1.3725**	10.3738 ± 1.2946	**7.9002 ± 1.6594**	13.574 ± 4.1126	9.3686 ± 1.0933	**7.2965 ± 1.0543**	14.4197 ± 2.3647
VQPooling NIQE	18.5909 ± 1.3763	14.9977 ± 1.0532	21.0495 ± 1.4606	15.9568 ± 1.2452	13.0289 ± 0.8736	18.1224 ± 1.4933	19.5398 ± 0.3799
Average-pooling NIQE	20.8613 ± 1.2765	18.9328 ± 0.7685	25.8699 ± 1.1595	21.3108 ± 1.1905	19.9309 ± 1.1202	20.2597 ± 0.6607	21.7613 ± 0.3903
VIIDEO	12.8631 ± 0.5798	13.2405 ± 0.0861	12.5237 ± 0.5433	13.3822 ± 0.7004	14.0196 ± 0.4885	12.5821 ± 0.6533	12.065 ± 0.2591

Table 15: Median PLCC ± Standard Deviation of state-of-the-art methods, evaluated on the LIVE-Qualcomm dataset. The experimental data set were divided according with the capture device. The best scores in each dataset are marked in bold.

VQA Method	Samsung Galaxy S5	Samsung Galaxy S6	HTC One VX	Apple Iphone 5S	LG G2	Nokia Lumia 1020	Samsung Galaxy Note 4	Oppo Find 7	All devices
BRISQUE	0.5355 ± 0.4530	0.3616 ± 0.3578	0.3668 ± 0.3840	0.7460 ± 0.5241	0.4910 ± 0.3392	0.3491 ± 0.6896	0.5183 ± 0.5325	0.0865 ± 0.3623	0.5893 ± 0.2295
FRIQUEE	0.6203 ± 0.6430	**0.7316 ± 0.3778**	0.5268 ± 0.4540	0.7560 ± 0.6441	**0.5710 ± 0.3292**	0.5191 ± 0.6296	0.6383 ± 0.5125	0.4350 ± 0.5123	0.5377 ± 0.2074
TLVQM	**0.7568 ± 0.5957**	0.5901 ± 0.4830	0.3807 ± 0.3980	**0.8888 ± 0.2550**	0.3103 ± 0.4234	0.4651 ± 0.6798	**0.7672 ± 0.4632**	0.2321 ± 0.4804	**0.7431 ± 0.2086**
QAWV	0.1628 ± 0.4470	0.6319 ± 0.4597	0.5687 ± 0.2446	0.8692 ± 0.1211	0.5184 ± 0.2816	**0.6635 ± 0.0847**	0.2311 ± 0.1139	**0.7101 ± 0.5393**	0.5800 ± 0.2349
VQPooling NIQE	0.2588 ± 0.0772	0.3343 ± 0.0683	0.4262 ± 0.0917	0.616 ± 0.0744	0.2455 ± 0.1281	0.2887 ± 0.1112	0.4117 ± 0.073	0.2144 ± 0.0728	0.4156 ± 0.0342
Average-pooled NIQE	0.2898 ± 0.0834	0.3323 ± 0.0751	0.5541 ± 0.063	0.544 ± 0.0843	0.3616 ± 0.0917	-0.2552 ± 0.3154	0.5976 ± 0.0627	0.5409 ± 0.0774	0.1214 ± 0.0387
VIIDEO	0.0147 ± 0.2314	-0.0035 ± 0.0489	**0.6994 ± 0.0453**	-0.0297 ± 0.1121	0.163 ± 0.0817	-0.1418 ± 0.0693	0.1986 ± 0.1247	0.3177 ± 0.1299	0.1812 ± 0.033

Table 16: Median SROCC ± Standard Deviation of state-of-the-art methods, evaluated on the LIVE-Qualcomm dataset. The experimental data set were divided according with the capture device. The best scores in each dataset are marked in bold.

VQA Method	Samsung Galaxy S5	Samsung Galaxy S6	HTC One VX	Apple Iphone 5S	LG G2	Nokia Lumia 1020	Samsung Galaxy Note 4	Oppo Find 7	All devices
BRISQUE	0.4000 ± 0.4580	0.3571 ± 0.3602	0.3714 ± 0.3966	0.8000 ± 0.5071	0.4857 ± 0.3763	0.5000 ± 0.7244	0.4000 ± 0.5194	0.0286 ± 0.3984	0.5864 ± 0.2232
FRIQUEE	**0.5970 ± 0.6180**	**0.7271 ± 0.3602**	0.5043 ± 0.4266	0.7352 ± 0.6071	**0.5357 ± 0.3763**	0.4984 ± 0.5844	0.6241 ± 0.5194	0.3840 ± 0.4951	0.5137 ± 0.3590
TLVQM	0.4000 ± 0.5255	0.4857 ± 0.4242	0.4857 ± 0.3837	**0.8000 ± 0.2781**	0.2857 ± 0.4254	0.5000 ± 0.6670	**0.8000 ± 0.4663**	0.2857 ± 0.4534	**0.7046 ± 0.2052**
QAWV	0.5000 ± 0.5705	0.6666 ± 0.4181	0.5000 ± 0.2672	0.6500 ± 0.1172	0.3818 ± 0.2672	**0.7000 ± 0.0805**	0.1786 ± 0.1417	**0.5250 ± 0.5273**	0.4960 ± 0.2323
VQPooling NIQE	0.3581 ± 0.1028	0.2908 ± 0.0891	0.3755 ± 0.0959	0.6846 ± 0.0873	0.1554 ± 0.0949	0.1392 ± 0.1337	0.5353 ± 0.0807	0.1758 ± 0.1179	0.3798 ± 0.0286
Average-pooled NIQE	0.3758 ± 0.0962	0.4111 ± 0.0932	0.4518 ± 0.079	0.5678 ± 0.1118	0.2037 ± 0.0912	0.2538 ± 0.2028	0.4853 ± 0.0939	0.2433 ± 0.0931	0.1726 ± 0.038
VIIDEO	0.1863 ± 0.1366	0.2274 ± 0.1037	**0.6194 ± 0.0636**	-0.1604 ± 0.1375	0.345 ± 0.0717	-0.0734 ± 0.1735	0.0062 ± 0.1224	0.3948 ± 0.0843	0.1656 ± 0.0331

Table 17: Median RMSE ± Standard Deviation of state-of-the-art methods, evaluated on the LIVE-Qualcomm dataset. The experimental data set were divided according with the capture device. The best scores in each dataset are marked in bold.

VQA Method	Samsung Galaxy S5	Samsung Galaxy S6	HTC One VX	Apple Iphone 5S	LG G2	Nokia Lumia 1020	Samsung Galaxy Note 4	Oppo Find 7	All devices
BRISQUE	8.0993 ± 2.6390	10.1976 ± 2.2885	14.2224 ± 2.7736	10.3809 ± 2.8149	8.4309 ± 2.2771	9.8677 ± 5.5404	10.0416 ± 3.0672	10.0661 ± 2.7636	9.8002 ± 1.9968
FRIQUEE	8.8530 ± 3.7369	**8.4618 ± 5.8141**	10.2060 ± 4.7414	10.0000 ± 5.2019	**8.1342 ± 3.7468**	**9.6075 ± 12.2930**	9.0602 ± 6.9392	10.128 ± 7.9075	11.3010 ± 2.2449
TLVQM	7.3005 ± 7.5952	9.2072 ± 2.8719	13.0060 ± 1.7017	**7.5709 ± 2.8485**	9.4152 ± 3.1761	10.9140 ± 4.5148	**8.4116 ± 2.0898**	9.4318 ± 3.6503	**8.6683 ± 3.0326**
QAWV	**4.6404 ± 2.3863**	11.7063 ± 4.1553	**7.8857 ± 1.5310**	7.7245 ± 1.6291	13.1268 ± 4.5048	10.3327 ± 1.3008	9.0973 ± 0.8471	10.7772 ± 2.7262	13.5597 ± 2.3447
VQPooling NIQE	10.941 ± 0.5634	11.3904 ± 1.0944	8.8865 ± 0.5735	12.6669 ± 1.0803	10.7673 ± 0.936	25.8833 ± 1.384	10.5937 ± 0.6508	11.0426 ± 1.1773	11.4259 ± 0.2802
Average-pooled NIQE	9.2822 ± 0.7811	11.1229 ± 1.1086	9.5627 ± 0.4938	12.9311 ± 1.2404	10.3468 ± 0.6989	12.8961 ± 1.4029	10.6292 ± 0.7766	**8.3124 ± 0.5372**	11.8962 ± 0.343
VIIDEO	76.0245 ± 17.1676	12.2814 ± 1.2175	9.1265 ± 0.7382	20.9853 ± 1.1363	11.7482 ± 0.8945	16.1526 ± 2.0654	14.7634 ± 1.1758	9.6715 ± 0.7306	12.2962 ± 0.3603

Table 18: Median PLCC ± Standard Deviation of proposed VQA method. AP indicates Average Pooling. NA indicates No Average (we use one feature vector each XX frames), and AP* suggests that only one video per unique scene is used. The results of the methods of six upper rows was taken from [15], since they were evaluated on the same dataset

VQA Method	Artifacts	Color	Exposure	Focus	Sharpness	Stabilization	All distortions
FRIQUEE [90]	0.7638	0.3543	0.6808	**0.8107**	0.2203	0.7034	0.7349
BRISQUE [74]	0.6402	0.3392	0.6042	0.4550	0.5371	0.6940	0.5788
Average-pooled NIQE [71]	0.6078	0.2904	0.4625	0.5371	0.5595	0.6015	0.6802
Temporally-pooled NIQE	0.6766	0.3141	0.5213	0.5782	0.5508	0.6510	0.6749
V-BLIINDS [76]	**0.8386**	**0.6645**	**0.6900**	0.8077	**0.6845**	**0.7138**	0.6653
VIIDEO [65]	0.2888	0.3312	0.2073	0.2515	0.3012	0.3697	0.0982
Conv5b_RGB_8Frames_AP	0.2619 ± 0.2933	0.3714 ± 0.3994	0.5429 ± 0.3391	0.4000 ± 0.5107	0.6000 ± 0.3309	0.1429 ± 0.3529	0.6100 ± 0.1446
Conv5b_RGB_16Frames_AP	0.4524 ± 0.3486	0.3714 ± 0.3141	0.6000 ± 0.3042	0.4000 ± 0.5407	0.6000 ± 0.3589	0.2857 ± 0.3182	0.5906 ± 0.1092
Fc6_YCbCr_8Frames_AP	0.5714 ± 0.2635	0.6000 ± 0.3452	0.6000 ± 0.3349	0.5000 ± 0.4169	0.6000 ± 0.3407	0.4643 ± 0.3339	$\mathbf{0.7749 \pm 0.0884}$
Fc6_YCbCr_8Frames_AP*	X	X	X	X	X	X	0.7648 ± 0.1487
Fc6_YCbCr_8Frames_NA	0.5494 ± 0.2380	0.6303 ± 0.3086	0.6183 ± 0.365	0.4828 ± 0.2854	0.5954 ± 0.3447	0.3767 ± 0.3175	0.7146 ± 0.0849
Fc6_MSCN_8Frames_AP	0.5476 ± 0.2399	0.4857 ± 0.3419	0.4286 ± 0.3015	0.3536 ± 0.5015	0.6559 ± 0.3929	0.1429 ± 0.4062	0.5824 ± 0.148
Fc6_MSCN_8Frames_NA	0.5000 ± 0.1992	0.5076 ± 0.2951	0.4146 ± 0.2588	0.4688 ± 0.353	0.5714 ± 0.2442	0.2284 ± 0.2947	0.5428 ± 0.1176
Fc6_MSCN_16Frames_NA	0.4730 ± 0.2061	0.4341 ± 0.2905	0.4635 ± 0.2622	0.5215 ± 0.3040	0.5082 ± 0.263	0.2215 ± 0.3029	0.5453 ± 0.1767
Fc6_MSCN_16Frames_AP	0.5000 ± 0.2887	0.4286 ± 0.3786	0.4857 ± 0.3020	0.3000 ± 0.4920	0.5798 ± 0.3255	0.1786 ± 0.3187	0.6129 ± 0.1439
Fc6_YCbCr_16Frames_AP	0.5476 ± 0.2593	0.4286 ± 0.3244	0.6000 ± 0.3566	0.3929 ± 0.2857	0.6571 ± 0.3381	0.4643 ± 0.3457	0.6380 ± 0.1194
Fc6_YCbCr_16Frames_NA	0.5495 ± 0.2581	0.4211 ± 0.2840	0.4698 ± 0.2685	0.4544 ± 0.3421	0.6192 ± 0.288	0.3757 ± 0.2754	0.5217 ± 0.1272

Table 19: Median SROCC ± Standard Deviation of proposed VQA method. AP indicates Average Pooling. NA indicates No Average (we use one feature vector each XX frames), and AP* suggests that only one video per unique scene is used.

VQA Method	Artifacts	Color	Exposure	Focus	Sharpness	Stabilization	All distortions
FRIQUEE [90]	**0.75**	0.4107	0.6071	0.7879	0.0714	**0.6607**	0.6795
BRISQUE [74]	0.6071	0.3571	0.5536	0.3929	0.4821	0.6429	0.5585
Average-pooled NIQE [71]	0.5	0.3214	0.3929	0.3393	0.5	0.2143	0.5451
Temporally-pooled NIQE	0.5357	0.3214	0.4821	0.3750	0.5179	0.2143	0.5525
V-BLIINDS [76]	0.7321	0.6071	0.6429	**0.8036**	0.6786	**0.6607**	0.6177
VIIDEO [65]	-0.1786	0.1429	-0.0714	0	-0.1786	-0.1071	-0.1414
Conv5b_RGB_8Frames_AP	0.2631 ± 0.2793	0.2127 ± 0.3107	0.5786 ± 0.3288	0.5908±0.4828	**0.7493±0.3068**	0.0 ± 0.2906	0.5752 ± 0.2141
Conv5b_RGB_16Frames_AP	0.3557 ± 0.3308	0.1564 ± 0.28	0.6153 ± 0.3282	0.478 ± 0.4948	0.7627 ± 0.333	0.0 ± 0.2962	0.5868 ± 0.1536
Fc6_YCbCr_8Frames_AP	0.6076±0.2364	0.7077 ± 0.3603	**0.7271±0.3263**	0.4524 ± 0.3819	0.7236 ± 0.346	0.445±0.3156	**0.7517±0.0853**
Fc6_YCbCr_8Frames_AP*	X	X	X	X	X	X	0.7507 ± 0.1281
Fc6_YCbCr_8Frames_NA	0.5365 ± 0.2688	**0.7294±0.3487**	0.6562 ± 0.4141	0.407 ± 0.2864	0.66 ± 0.3744	0.3361 ± 0.3048	0.69 ± 0.2985
Fc6_MSCN_8Frames_AP	0.5776 ± 0.2814	0.4583 ± 0.3482	0.4484 ± 0.3055	0.3401 ± 0.431	0.6493 ± 0.3867	0.1465 ± 0.3774	0.5781 ± 0.171
Fc6_MSCN_8Frames_NA	0.4943 ± 0.219	0.4348 ± 0.2872	0.4029 ± 0.2711	0.3527 ± 0.3269	0.6114 ± 0.2818	0.226 ± 0.3087	0.4631 ± 0.263
Fc6_MSCN_16Frames_NA	0.4761 ± 0.2101	0.3725 ± 0.2798	0.4192 ± 0.2782	0.4534 ± 0.2974	0.5854 ± 0.243	0.281 ± 0.3122	0.4901 ± 0.2458
Fc6_MSCN_16Frames_AP	0.4869 ± 0.298	0.4512 ± 0.3203	0.469 ± 0.2971	0.3096 ± 0.4598	0.6562 ± 0.3403	0.0488 ± 0.291	0.5831 ± 0.1898
Fc6_YCbCr_16Frames_AP	0.5776 ± 0.2549	0.3734 ± 0.2988	0.5396 ± 0.346	0.4009 ± 0.2969	0.7477 ± 0.3627	0.4276 ± 0.3254	0.617 ± 0.1274
Fc6_YCbCr_16Frames_NA	0.5256 ± 0.274	0.4393 ± 0.2799	0.4888 ± 0.2945	0.4744 ± 0.3206	0.7043 ± 0.3372	0.3679 ± 0.2782	0.5238 ± 0.2349

5 VOT tracker algorithm robust against post-capture distortions

Video object tracking (VOT) aims to determine the location of a target over a sequence of frames. VOT in realistic scenarios is a difficult task. The existing body of work has studied various image factors that affect VOT performance. For instance, factors such as occlusion, clutter, object shape, unstable speed, and zooming, that influence video quality affect tracking performance. Nonetheless, there is no clear distinction between scene-dependent challenges such as occlusion and clutter and the challenges imposed by traditional notions of "quality impairments" inherited from capture, compression, processing, and transmission. This chapter is concerned with the latter interpretation of quality as it affects video tracking performance.

The contributions of this chapter are twofold. First, we propose the design and implementation of quality-aware feature selection for VOT. First, we divided each frame of the video into patches of the same size and extracted Histogram of Oriented Gradients (HOG) and natural scene statistics (NSS) features from these patches. Then, we degrade the videos synthetically with different levels of post-capture distortions such as MPEG-4, Additive White Gaussian Noise (AWGN), salt and pepper, and blur. Finally, we define the best set of features HOG and NSS that generate the largest area under the curve in the success plots, yielding an improvement in the video tracker performance in videos affected by post-capture distortions.

To the best of our knowledge, none of these approaches have modeled and quantified the influence of post-capture distortions on object tracking performance. This question is very important because state-of-the-art trackers perform well in videos with few or no distortions, but when they are tested on videos affected by distortions, such as those acquired by surveillance cameras, the performance can be degraded to a large extent. We proposed an approach to integrate NSS perceptual quality features into a video object tracker scheme and demonstrated its performance in several videos affected by post-capture distortions. Previous studies on this topic focused on tasks such as object and face detection [85], dermoscopy [86], and face recognition in long-wave infrared (LWIR) images [193] [194].

Dai et al. [195] studied the influence of shaking motions on VOT. They found that trackers fail owing to this distortion because of two main issues. The first occurred when the entire tracked target was not found in the candidate's patches because it moved out fast owing to significant shaking motion. Second, shaking movement may generate blur distortion, and trackers are confused by blurred boundaries between the foreground and background. To the best of our

knowledge, this is the first study to propose a quality-aware feature extraction approach for VOT. Furthermore, this approach complements our previous work in which we demonstrated the impact of authentic distortions on state-of-the-art video trackers [149].

5.1 Materials and Methods

We present a video object tracker framework based on a support vector machine (SVM) [196–198]. Figure 21 shows a diagram of the proposed approach. Basically, this tracker consists of classification between the target and background of patches represented in a feature space [199].

Video Object Tracker

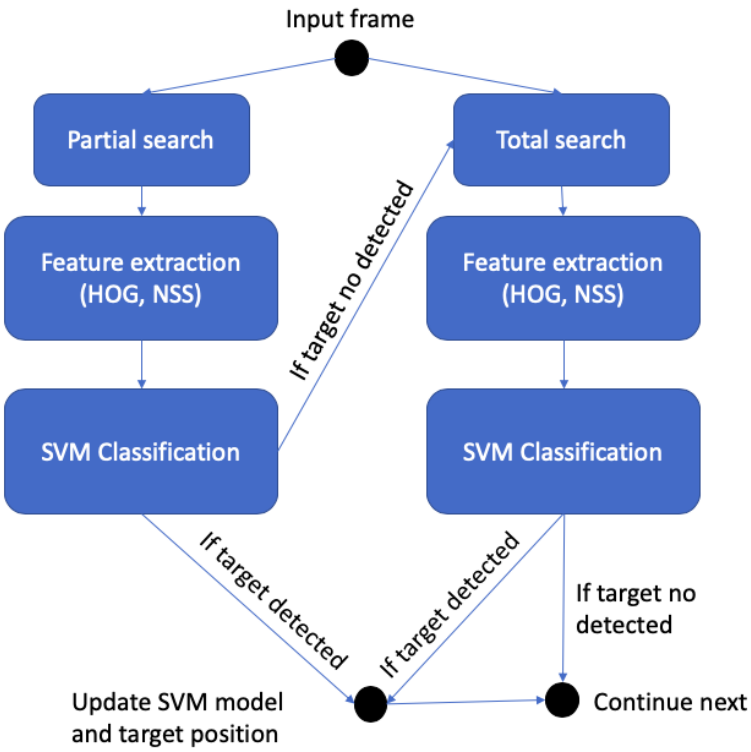


Figure 21: State diagram of the proposed video object-tracking framework.

For starting the tracking, our algorithm requires the bounding box that indicates the location of the target in the first frame. The initial training set is generated as follows: as target class

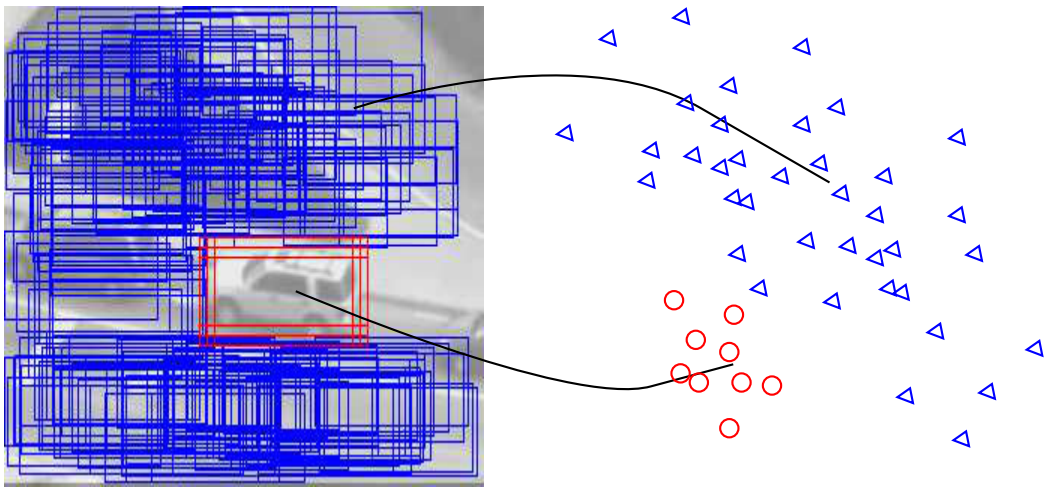


Figure 22: On the left, background and target patches `bgP` and `objP` in blue and red, respectively.

examples, we set the number of target objects $N_{tar} = 10$. These target objects are computed from the first bounding box and its neighborhood (random patches that overlap with at least 80% of the target bounding box). Likewise, as background examples, we randomly selected N_{bg} patches that did not overlap with the target. Figure 22 illustrates the target and background patches used to generate the training set.

In our experiments, N_{bg} corresponded to 10% of feature representation. We used this data to train a linear two-class Support Vector Machine (SVM) classifier. In the next frame, we perform a "partial search" of the target. For this purpose, we find a set of query patches from the image region that contains the bounding boxes that overlap with at least 50$\%$ with the bounding box of the target in the previous frame. Subsequently, we classify the query objects using the current SVM classifier. If one patch is classified as the target, the location is updated accordingly. If more than one patch is classified as a target, we interpolate the bounding boxes of all of them. If the target is not found, we carry out a "total search." The total search consists of the classification of all the patches (with the same size as the target) in the entire image. If at least one patch is classified as the target, the process of updating the location is the same as that in the partial search. After updating the target location, we update the training set by choosing a new target and background examples, as in the first frame. The maximum size of the training set is the feature dimension. When the dataset exceeded the maximum size, we eliminated the oldest examples. If the target is not found in the total search, we move to the next frame without updating the location or the training set.

To evaluate the video tracker performance, we used success plots inspired by the work presented in [3]. To generate a success plot, we calculate an overlap score S for each frame of a video, defined as:

$$S = \frac{|r_t \cap r_o|}{|r_t \cup r_0|}, \tag{17}$$

where r_t denotes the object bounding box estimated by the tracker, r_0 is the ground-truth bounding box, $\cap$ and $\cup$ are the intersection and union operators respectively, and $|\cdot|$ represents the number of pixels within a region. Figure 23 shows a graphic description of the regions that define S.

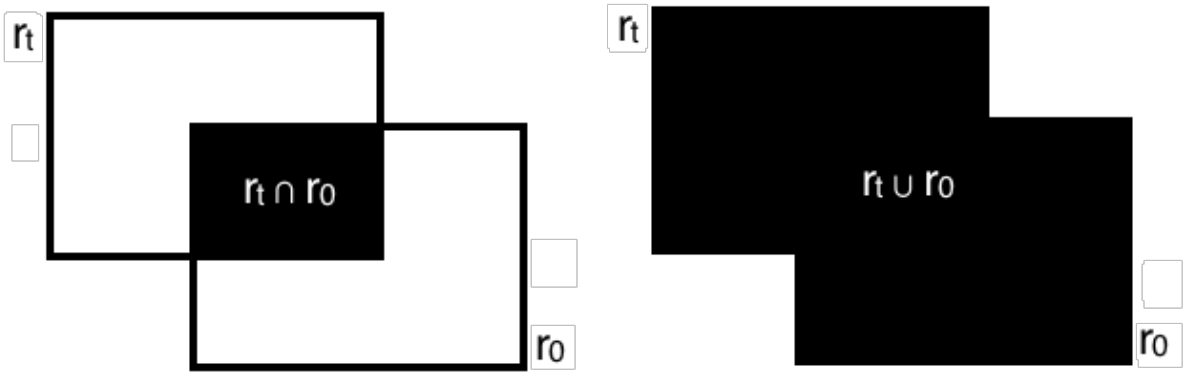


Figure 23: Definition of the regions used for calculating the overlap score S.

We define the area under the curve (AUC) as the trapezoidal integral of the success plot (i.e., S along with the entire video). AUC lies in the range [0,1], and values closer to 1 indicate better tracking performance.

Feature extraction

For the analysis in the vector space (SVM classifier training and test), we represent a patch in the video frame as feature vector $x \in \mathbb{R}^d$ (where d is the dimension of the space), by two types of descriptors:

- **Histogram of Oriented Gradients (HOG)**: HOG is a well-recognized feature owing to its superior performance and relatively simple computation. Since first proposed in [200], HOG and its derivatives have been widely used in numerous object tracking algorithms [164, 201–210]. Even with the popularity of CNNs, HOG-based algorithms are

still favorable in many applications owing to their balanced tradeoff between accuracy and computational complexity [211]. To calculate HOG, a 1-D histogram is computed from the gradient directions in a small region of an image [200]. This region is called a "cell." To reduce the effect of the illumination changes over the image, each cell is normalized by the total energy in a set of neighbor cells called "block." We take as the HOG feature representation $x_{HOG} \in \mathbb{R}^{d_{HOG}}$, the average of the histograms of all the cells inside the corresponding bounding box. The dimension of this representation is given by the number of histogram bins d_{HOG}.

- **Natural Scene Statistics in the spatial domain (NSS)** [74]: We compute the NSS feature representation $x_{NSS} \in \mathbb{R}^{d_{NSS}}$ of a patch by the 36 features extracted from locally normalized luminances as described in [74].

To avoid an undesired effect induced by the scale of the features, we consider two normalization methods for each type of descriptor:

- z-score: the mean μ and standard deviation σ of each feature distribution are 0 and 1, respectively as follows

$$\bar{x}_z = \frac{x - \mu_o}{\sigma_o},$$

where the original mean μ_o and standard deviation σ_o were estimated from the training set.

- 0-1-normalization: each feature distribution is re-scaled in order to set 0 as the minimum value and 1 as the maximum value:

$$\bar{x}_n = \frac{x - x_{\min}}{x_{\max}}.$$

5.2 FRIQUEE and TLVQM Quality Features for Robust Tracking

We experimented with TLVQM [70] and FRIQUEE [90] quality-aware features, concatenating these features with the original features of the tracker DLSSVM. Figures 24 and 25 show the tracking performance results obtained with tracker DLSSVM [156] concatenated with 512 FRIQUEE features. Figure 24 shows the tracking performance of a pristine video. We note that DLSSVM + FRIQUEE presents better performance (larger AUC) compared with baseline

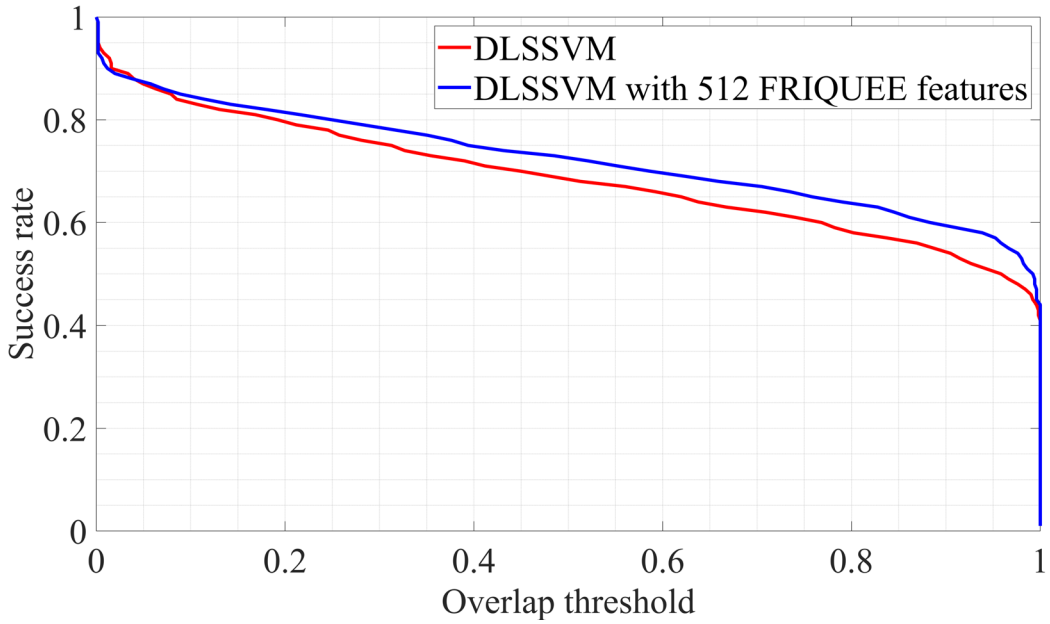
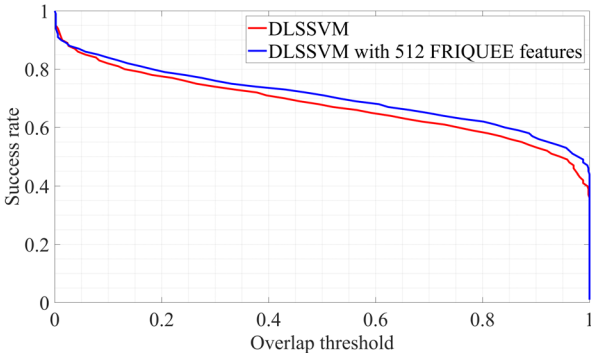


Figure 24: Success plot of tracking in pristine video of person walking at indoor space, red line is DLSSVM tracker, and blue one is DLSSVM added with FRIQUEE features. AUC FRIQUEE = 0.7240, AUC DLSSVM = 0.6639. The performance with FRIQUEE features is better.

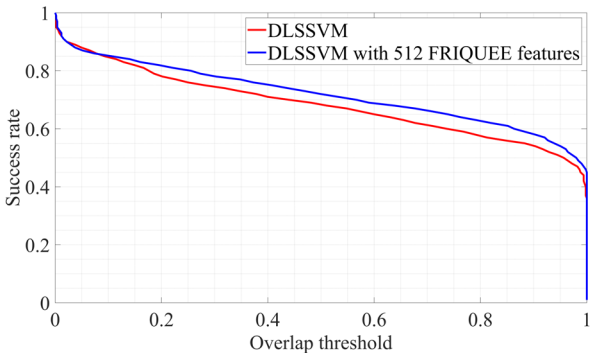
DLSSVM. In this case, the increase in performance is 8.31%, which is not negligible. Figure 25 shows the tracking performance on two videos with post-capture distortions (blur and S&P noise). Table 20 shows the AUC and processing times for sets of authentically distorted video. We observe that the AUC obtained by using FRIQUEE features concatenated with DLSSVM original features is larger. The FRIQUEE-DLSSVM features combination exceeds the baseline performance in 3.96% for blur distortion and 4.9% for S&P post-capture distortion. Nonetheless, because of the time required to extract the features, the processing times for FRIQUEE were one order of magnitude higher than DLSSVM, making it inapplicable for real-time intended trackers.

We repeated the previous experiments, but now tested with authentically distorted videos containing in-capture distortions such as lack of focus (defocus), exposure time (exposure), and focus commingled with exposure. For this instance, we applied z-score normalization to 560 FRIQUEE features. Based on the results shown in Figures 25 and 26, it can be noted superiority of z-normalized (mean 0, standard deviation =1) FRIQUEE features concerning the performance of the baseline tracker DLSSVM.

Similarly, we did other experiments with videos containing simple contents (e.g., Walking Person). These videos have in-capture distortions (exposure and defocus). We added TLVQM

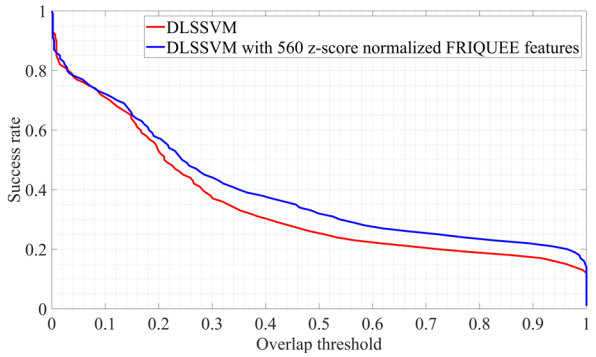


(a) High level of blur, AUC FRIQUEE = 0.6949, AUC DLSSVM = 0.6674

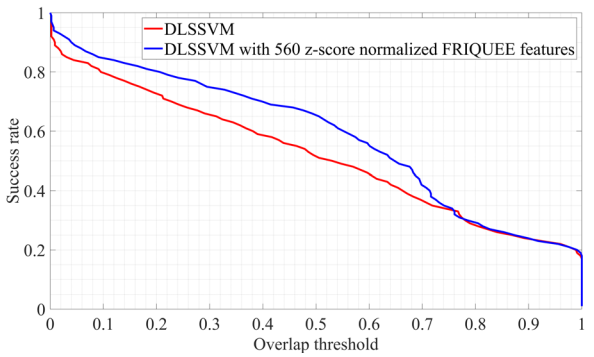


(b) Salt and Pepper noise, AUC FRIQUEE = 0.7087, AUC DLSSVM = 0.6740

Figure 25: Success plot of tracking in videos with high level of post-capture distortions. The video contains one scene with a person walking at indoor space, red line is DLSSVM tracker, and blue one is DLSSVM added with FRIQUEE features. The performance with FRIQUEE features is slightly better.



(a) High level of Exposure(0298ExpIndWLLQC1), AUC FRIQUEE = 0.3825, AUC DLSSVM = 0.3354



(b) High level of Exposure and focus distortions commingled (0370ExFoIndWLLQC1), AUC FRIQUEE = 0.5727, AUC DLSSVM = 0.5080

Figure 26: Success plot of tracking in videos with high level of in-capture authentic distortions. The video contains one scene with a person walking at indoor space, red line is DLSSVM tracker, and blue one is DLSSVM added with FRIQUEE features with z-normalization. The performance with FRIQUEE features is better, increasing the AUC and tracker robustness.

features [70] of Low Complexity (LC), High Complexity (HC), and combined (TLVQM). We calculated Low Complexity features using the previous and the next frame, in addition to the current frame. We computed the HC features for all frames of the video. The results are presented in Fig. 27. The best performance was obtained using TLVQM HC y LC features combined, obtaining an AUC$= 0.5634$, achieving a performance 3.92% higher than that of the DLSSVM baseline.

Table 20: Average AUC and processing times for set 2 of authentically distorted videos.

video	**# frames**	**resolution**	**FRIQUEE time**	**DLSSVM Time**	**AUC DLSSVM**	**AUC FRIQUEE**
0372ExFo_IndWL_LQ_C3	345	1920*956	2622	128	0.4467	0.5167
0276Pri_IndWL_C3	346	1920*1080	2467	88	0.4992	0.5135
0284Exp_IndWL_MQ_C3	346	1920*956	2410	75	0.5493	0.5539
0324Fo_IndWL_LQ_C3	346	1920*956	2528	74	0.4278	0.4362
Average			2506.75	91.25	0.48075	0.50508

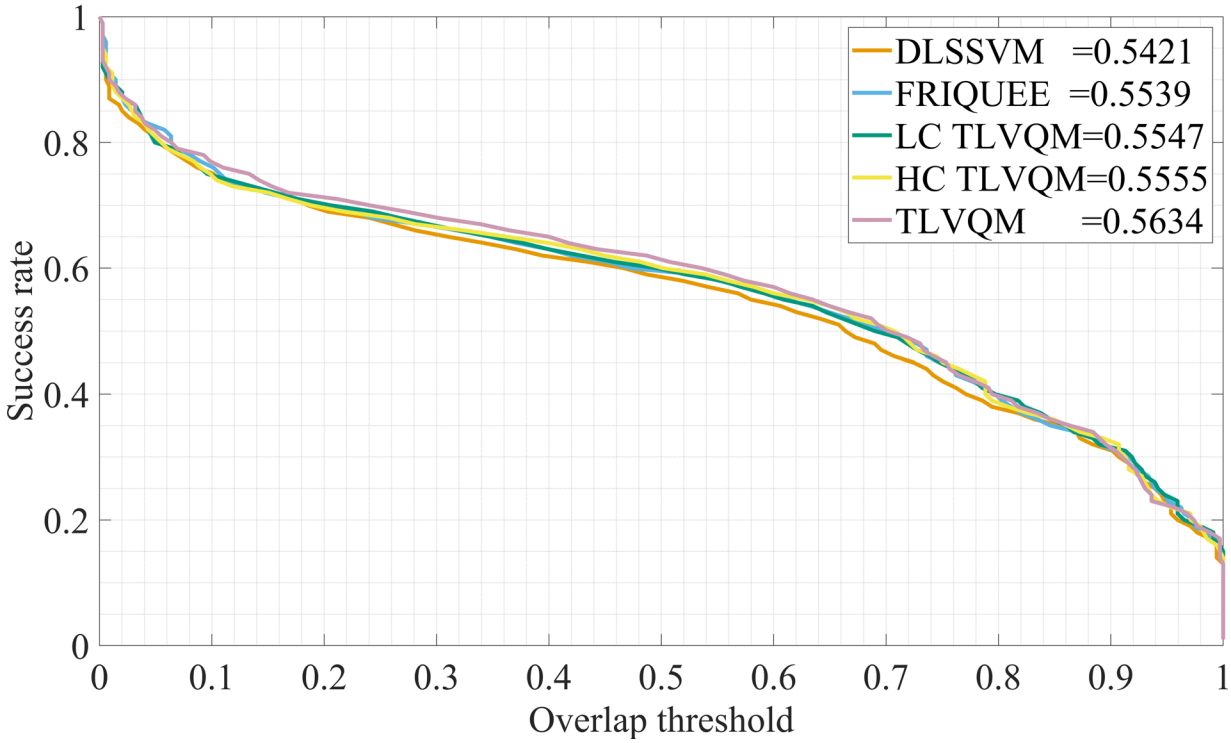


Figure 27: AUC performance using bags of several features tested in a video with Medium Level of Exposure distortion from AD-SVD (0284ExpIndWLMQC3). TLVQM HC and LC combined features get better performance, outperforming in 3.92% the other features.

5.3 Results with HOG and NSS method

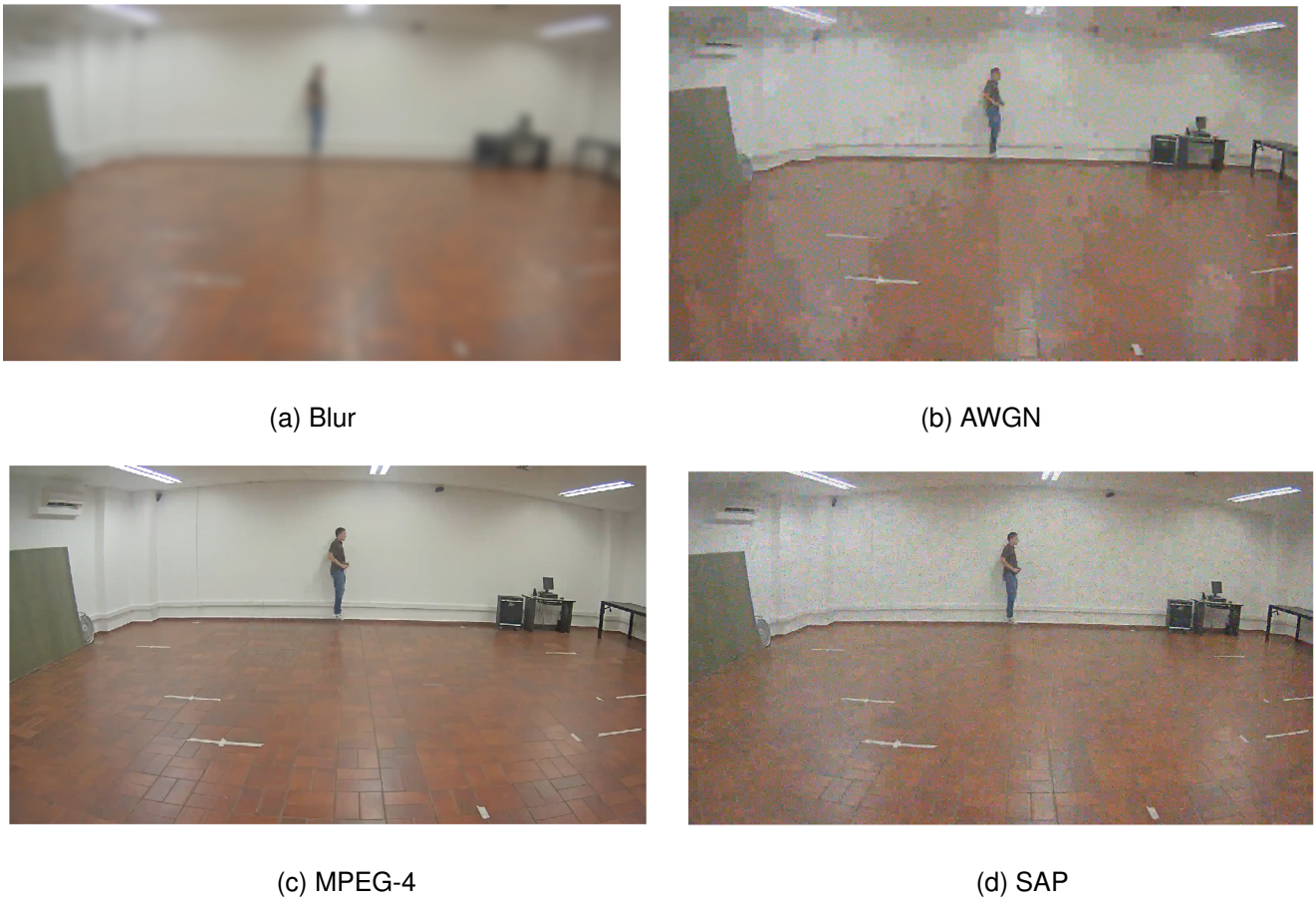
This section shows the object tracking results obtained using the HOG and HOG+NSS methods, tested on 910 videos of the constructed dataset. Our Quality-Aware tracker was implemented with MATLAB 2018a, and all the experiments were computed on a PC equipped with an Intel i7 8750-H CPU, 32GB RAM, and a single NVIDIA GTX 1060 GPU.

Figure 28: Pristine frame of proposed dataset, person jumping in indoor environment.

Image distortions

We consider four basic types of distortions that commonly occur in digital devices and over communication channels. The distortions used here are related to the encoding (compression) and transmission processes (post-capture distortions) [85].

- **AWGN**, Additive White Gaussian Noise: This is a local distortion, in which a zero-mean Gaussian noise of variance parameter is added independently to each pixel. The imnoise() function in MATLAB was used to introduce additive white Gaussian noise to the images.



(a) Blur (b) AWGN

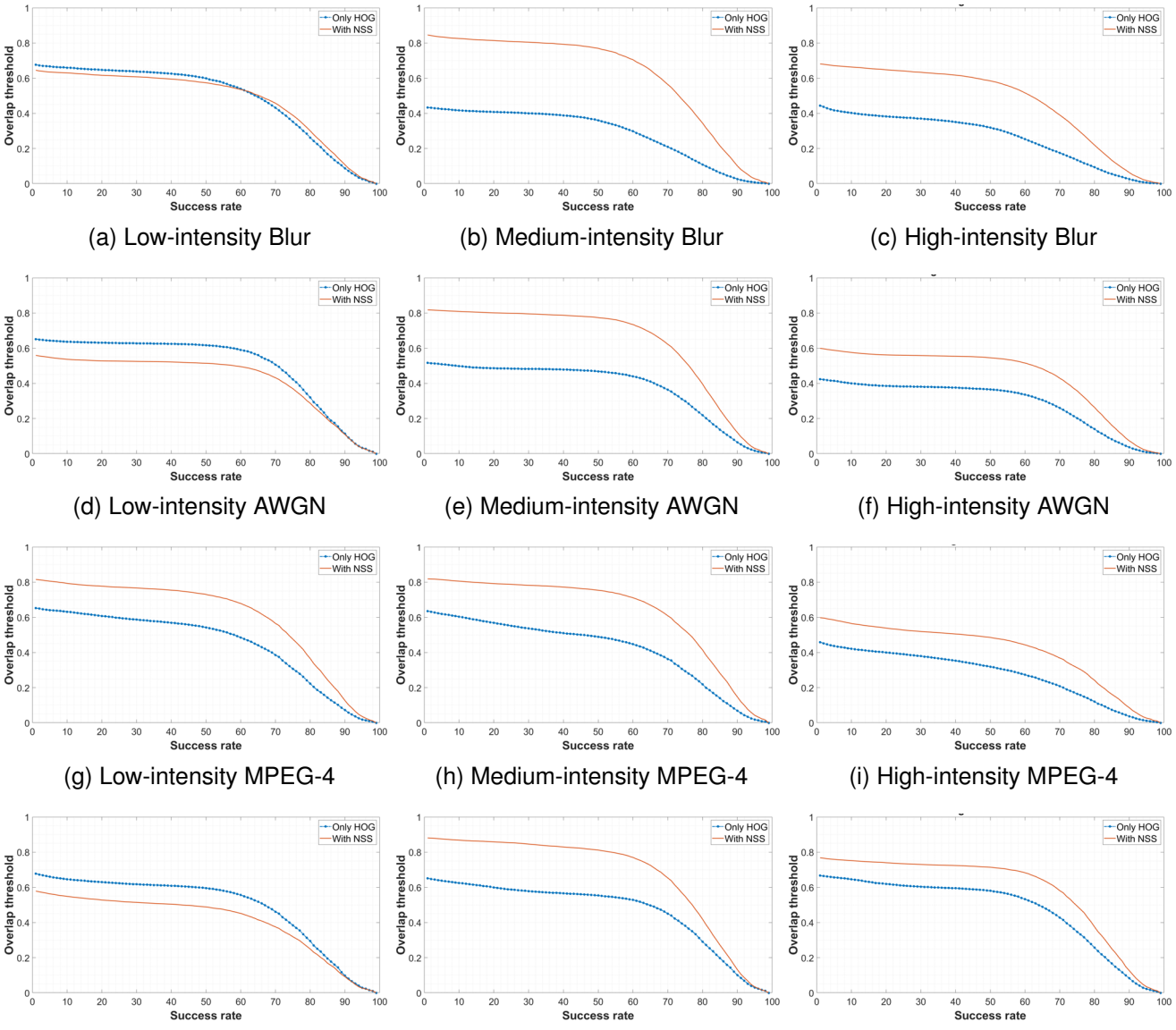
(c) MPEG-4 (d) SAP

Figure 29: Distorted frames with high-level intensities of post-capture distortions.

Three levels of AWGN were added with the noise variance parameters equal to [0.01, 0.05, 0.1].

- **Blur**: This is a global distortion in which each pixel is blurred through convolution with a Gaussian low-pass filter with standard deviation. The imfilter() function in MATLAB was used to introduce Gaussian blur at three levels. The standard deviation of the Gaussian filter was varied over a log scale, σ_B = [5, 10, 15].

- **Salt and pepper noise (SAP)**: This distortion generates only a few noisy pixels. The effect is similar to sprinkling white and black dots (salt and pepper) on the image. One example in which salt and pepper noise arises is the transmission of images over noisy digital links [212]. We used the imnoise() function in MATLAB to introduce SAP noise to the images. We added three levels of SAP noise, where D is the noise density. This noise affects the $D*numel(I)$ pixels. The values used for D are 0.01, 0.05, 0.1 for high, medium,



(a) Low-intensity Blur (b) Medium-intensity Blur (c) High-intensity Blur

(d) Low-intensity AWGN (e) Medium-intensity AWGN (f) High-intensity AWGN

(g) Low-intensity MPEG-4 (h) Medium-intensity MPEG-4 (i) High-intensity MPEG-4

(j) Low-intensity SAP (k) Medium-intensity SAP (l) High-intensity SAP

Figure 30: The overlap success plots of Quality-Aware tracker tested on videos with several post-capture distortions and intensity levels. The red line is the performance of quality-aware method using NSS features. The blue dotted line is the performance of only HOG features representation.

and low levels, respectively.

- **MPEG-4** compression: The MPEG-4 standard is used in a wide variety of video applications, such as DVDs and digital broadcast television. Compression systems such as MPEG-4 produce relatively uniform distortions and quality in the video, both spatially and temporally. MPEG-4 compressed videos exhibit typical compression artifacts such as blocking, ringing, and motion compensation mismatches around the object edges [35]. To generate this distortion, we used the ffmpeg[4] tool with a bitrate of 100Kbps, 200Kbps, and 1Mbps for high, medium, and low levels, respectively.

Proposed post-capture distorted dataset

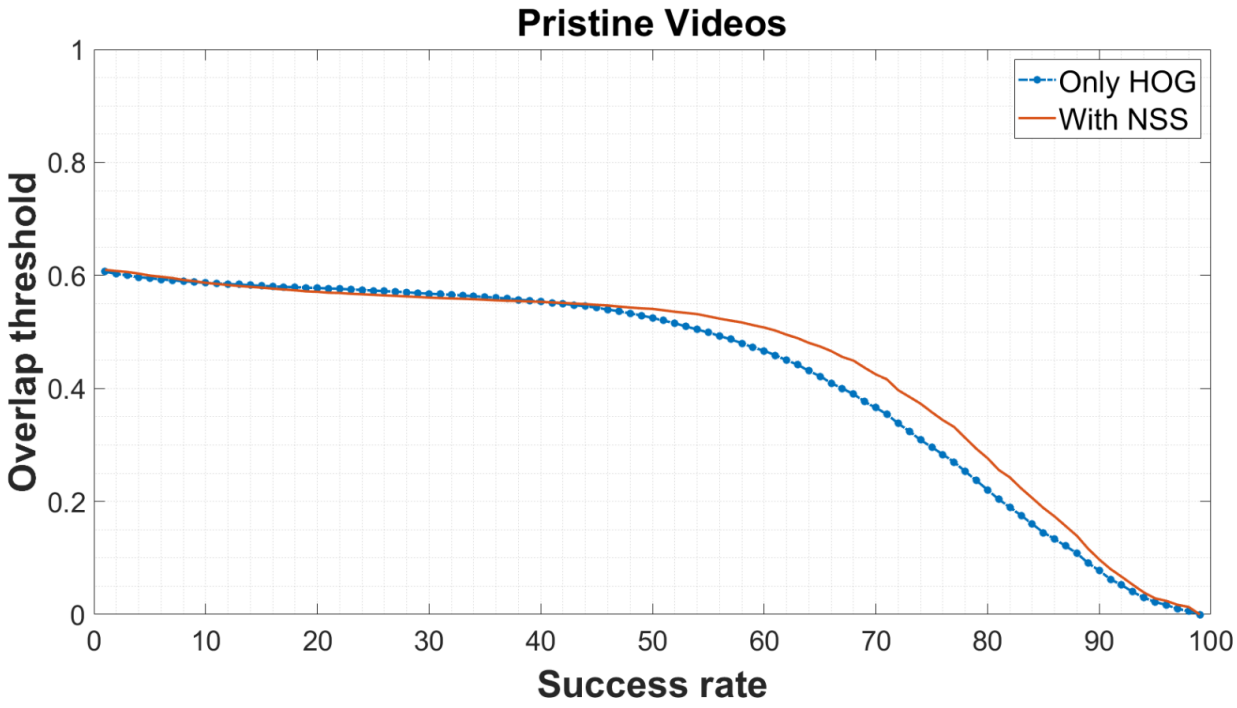


Figure 31: Results in pristine videos of the proposed dataset. The red line is the performance of quality-aware method using NSS features. The blue dotted line is the performance of only HOG features representation.

We recorded 70 pristine videos with activities such as walking, jumping, and sitting. We chose these activities because they are simple, with only one target in the scene. This condition allows us to test several trackers fairly in such a way that the most critical challenge is the distortions

[4]FFmpeg Developers, (2016). Available from http://ffmpeg.org/

Table 21: Statistical significance matrix of AUC values between HOG and HOG+NSS . A value of “1” indicates that the performance of the model with NSS was statistically better than that of the model with only HOG, “0” means that it is statiscally worse, and “-” means that it is statistically indistinguishable

	pristine	MPEG-4	S&P	Blur	Gaussian
Statistical significance	-	1	1	1	1

applied. The videos were recorded using four surveillance cameras in an indoor environment. These pristine videos were impaired with four post-capture distortions: blur, AWGN, MPEG-4, and SAP. All of these distortions have three intensity levels. Hence, the entire dataset had 910 videos (70 pristine and 210 per distortion). Figure 28 shows one pristine video frame, and Figure 29 presents the same frame affected by the four distortions at the highest level.

Figure 30 shows the average success plots from 70 videos corresponding to one level of distortion (low, medium, and high). The HOG+NSS representation performs significantly better than only the HOG features, except when the distortion is low. This difference is due to the specialization of our method in highly distorted videos, indicating that the introduction of NSS features is more helpful in VOT tasks with highly distorted videos. To demonstrate this idea, Figure 28 shows the performance of the pristine videos. This performance is almost equivalent between both methods, being slightly superior the HOG+NSS method. In pristine videos, the best performance, in AUC terms, was obtained using NSS features, as shown in Figure 31.

Statistical significance test

Because non-parametric tests make no assumptions about the probability distributions of the variables, we conducted a Kruskal-Wallis test on each median value of AUC for 70 videos comparing the HOG and HOG+NSS methods to evaluate whether the results presented in 30 were statistically significant. Table 21 tabulates the results of the statistical significance test. From Table 21, we conclude that the HOG+NSS method produced highly competitive object tracking performance on the tested videos with statistical significance against the HOG only algorithm.

Conclusions

We tested the DLSSVM tracker in a set of videos from AD-SVD containing in-capture and post-capture distortions. We analyzed the DLSSVM baseline features and sets of baseline features concatenated with FRIQUEE and TLVQM quality-aware features. Our findings suggest that the TLVQM quality-aware features HC and LC combined improve the tracking performance. Other experiments confirm that z-score normalization must be part of the pre-processing stages intended to improve performance. Besides, the time processing of TLVQM features is an average of 0.25 seconds per frame in FHD resolution. Adding to the DLSSVM baseline time, this processing time allows for at least 1 FPS speed, a typical FPS in trackers not intended for real-time.

The results obtained by selecting and incorporating quality-aware features into the representation of the image patches show an improvement in the VOT performance, in terms of AUC, for blur, SAP, AWGN, and MPEG-4 post-capture and encoding distortions. The performance loss in unconstrained VOT can be due to several scene conditions such as occlusion, scale change, and cluttering, which are different from image quality degradation. This chapter only focused on VOT for scenes without occlusions and scale changes to isolate the loss in performance due to image quality.

6 Video Tracker Performance Prediction in Authentically Distorted Videos

Choosing a VOT algorithm for a particular application is challenging because of the high number of competitive algorithms published each year [213]. To facilitate comparisons across different approaches, we designed a model-agnostic (independent of the tracker model [214]) framework that predicts performance without running the corresponding tracking algorithm.

6.1 Proposed Prediction Method

We learn a mapping [133] between the input video and the area under the curve (AUC) of the success-plot. This process is carried out in two stages, as is shown in Figure 32: i) extraction of a fixed-size set of features, and ii) AUC estimation using a support vector machine regression model.

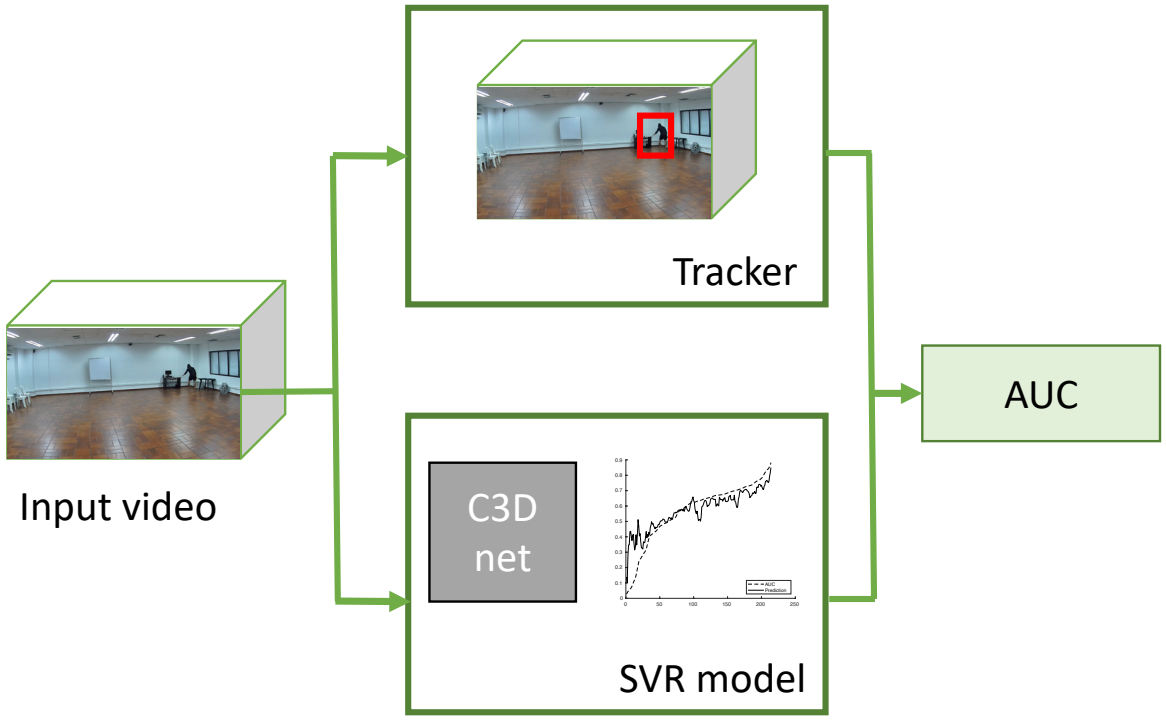


Figure 32: Performance prediction framework.

Figure 33: C3D architecture [12].

Feature extraction

C3D [12] 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 [173], action similarity labeling, scene classification, and object recognition [12]. Similarly, other studies have demonstrated that a CNN trained for object and video recognition could be useful in determining human perceptual characteristics [95, 97, 98, 173]. 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 [12] 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 33. We found better performance when using the corresponding 1024 principal components projection with a retained variance of 99%.

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 [215]:

$$\min_{\mathbf{ff},\mathbf{ff}^*} f(\mathbf{ff},\mathbf{ff}^*) = \frac{1}{2}(\mathbf{ff}-\mathbf{ff}^*)^T K(\mathbf{ff}-\mathbf{ff}^*) + \epsilon\sum_{i=1}^{l}(\alpha_i+\alpha_i^*) + \sum_{i=1}^{l} z_i(\alpha_i-\alpha_i^*)$$

$$\text{subject to } \sum_{i=1}^{l}(\alpha_i-\alpha_i^*) = 0$$
$$0 \le \alpha_i, \alpha_i^* \le C, i = 1,\ldots,l,$$

where $\mathbf{ff}$ and $\mathbf{ff}^*$ 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 new values, we use

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

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

6.2 Results

We used the t-distributed Stochastic Neighbor Embedding (TSNE) method [216], to reduce the dimensional space of the C3D features. TSNE is a variation of Stochastic Neighbor Embedding, which is more straightforward to optimize. We searched for hyper-parameters such as perplexity and the number of C3D features (in powers of 2). Perplexity can be interpreted as a smooth measure of an adequate number of neighbors. Typical values for perplexity are between 5 and 50. The search range adopted for perplexity was $[1-300]$, and for the number of C3D features was $[2^1-2^{11}]$. The results are shown in Figure 34. We obtained a higher accuracy with perplexity = 212 y 8 C3D features. We repeated 100 times the SVR training using this configuration, and the median accuracy obtained was 0.3988 ± 0.0464.The average computational time per iteration was 7.46 seconds.

Tables 22 to 28 tabulate the Spearman-Rank Correlation Coefficient (SRCC), Pearson Linear

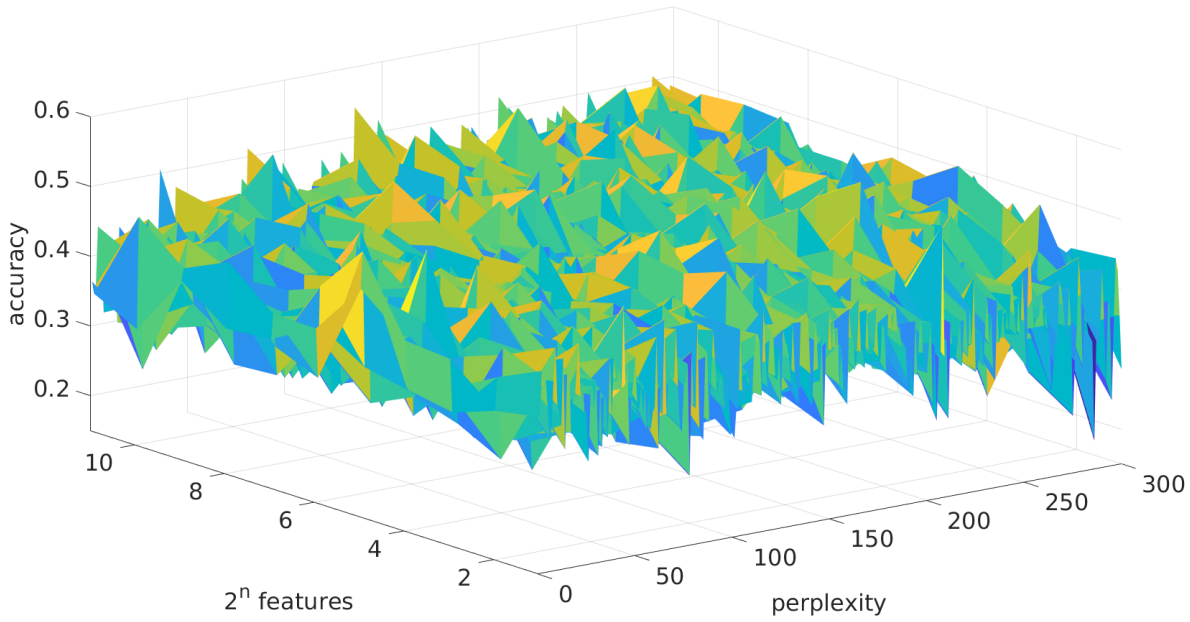


Figure 34: TSNE Reduction results of C3D features of 907 videos with post-capture distortions. The higher accuracy was obtained with perplexity=212 y 8 C3D Features PCA reduction.

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 contained $75\%, 25\%, 5\%$ 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 the training set increased, the prediction of the AUC became more accurate. For instance, the correlations (PLCC, SRCC) of GNet in Theater location (Table 22) were (0.9034,0.8584), (0.7216,0.7279), (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 other VOT algorithms in Tables 23, 24, 26, and 27. 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 [70], 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

Table 22: Theater location.

Tracker	Distribution mix											
	75%			25%			5%			0%		
	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE
CFWCR	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	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	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	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	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	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	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	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	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	0.8226	0.8575	0.0076	0.6169	0.5673	0.0171	0.3617	0.2629	0.0264	0.0537	0.0159	0.0772

Table 23: Parking Lot 2 location.

Tracker	Distribution mix											
	75%			25%			5%			0%		
	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE
CFWCR	-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	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	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	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	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	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	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	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	-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	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 24: Parking Lot 1 Location.

Tracker	Distribution mix											
	75%			25%			5%			0%		
	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE
CFWCR	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	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	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	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	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	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	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	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	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	0.7839	0.8170	0.0203	0.5019	0.5053	0.0396	0.2549	0.2548	0.0528	0.1294	0.1241	0.0553

Table 25: Media Room location.

Tracker	Distribution mix											
	75%			25%			5%			0%		
	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE
CFWCR	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	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	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	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	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	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	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	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	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	0.8474	0.8048	0.0129	0.6810	0.5894	0.0251	0.5259	0.4561	0.0368	0.2178	0.1714	0.0530

Table 26: Industrial Lab location.

Tracker	Distribution mix											
	75%			25%			5%			0%		
	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE
CFWCR	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	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	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	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	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	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	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	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	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	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 27: Guayacanes Hall location.

Tracker	Distribution mix											
	75%			25%			5%			0%		
	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE
CFWCR	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	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	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	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	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	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	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	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	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	0.6556	0.6873	0.0396	0.5407	0.5938	0.0504	0.5057	0.4878	0.0676	0.3660	0.3463	0.1000

Table 28: Electronics Lab location.

Tracker	Distribution mix											
	75%			25%			5%			0%		
	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE
CFWCR	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	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	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	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	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	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	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	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	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	0.6506	0.7265	0.0194	0.4225	0.4149	0.0272	0.2088	0.1870	0.0353	0.0408	0.0493	0.0446

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.

Conclusions

This chapter proposes and tests 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. The PLCC correlation obtained in predicting performance with some state-of-the-art trackers was superior to 0.90, indicating promising results. This precision level can potentially enable the integration of this tracker prediction as a module in a framework where a video can be executed with different trackers, depending on the predicted performance and video characteristics.

7 Tracking Time Reduction using Spatial down-scaling

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 means that videos must be compressed. We can obtain compression by increasing the quantization factor, changing the frame rate, or decreasing the frame resolution [217]. Since these compression alternatives can reduce not only the video quality perceived by a user but also the performance of analysis algorithms such as VOT [150, 218], 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 reduction is achieved by predicting the VOT performance in authentically distorted surveillance videos to determine the optimal frame resolution scale for processing the video. This optimal scale reduces the video storage demands and the execution time of the video tracker, thereby preserving its performance.

7.1 Proposed Method

The experimental results show that 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 the 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 gain in processing time. The decision criterion is the threshold for pl. Figure 35 illustrates this process.

We chose an SVR as our model type and used the C3D features to train the models. We followed the ensemble model approach in which 10 SVR models were trained and the difference between them is only the hyperparameter and the training set used. The final prediction is

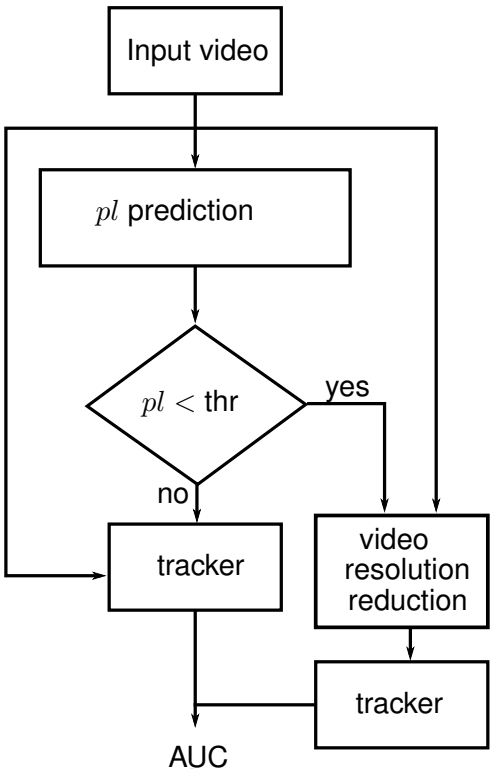


Figure 35: Framework for time reduction - thr represents the threshold for maximum performance loss that the system will allow.

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 scale and tracker, we trained different predictors.

7.2 Results

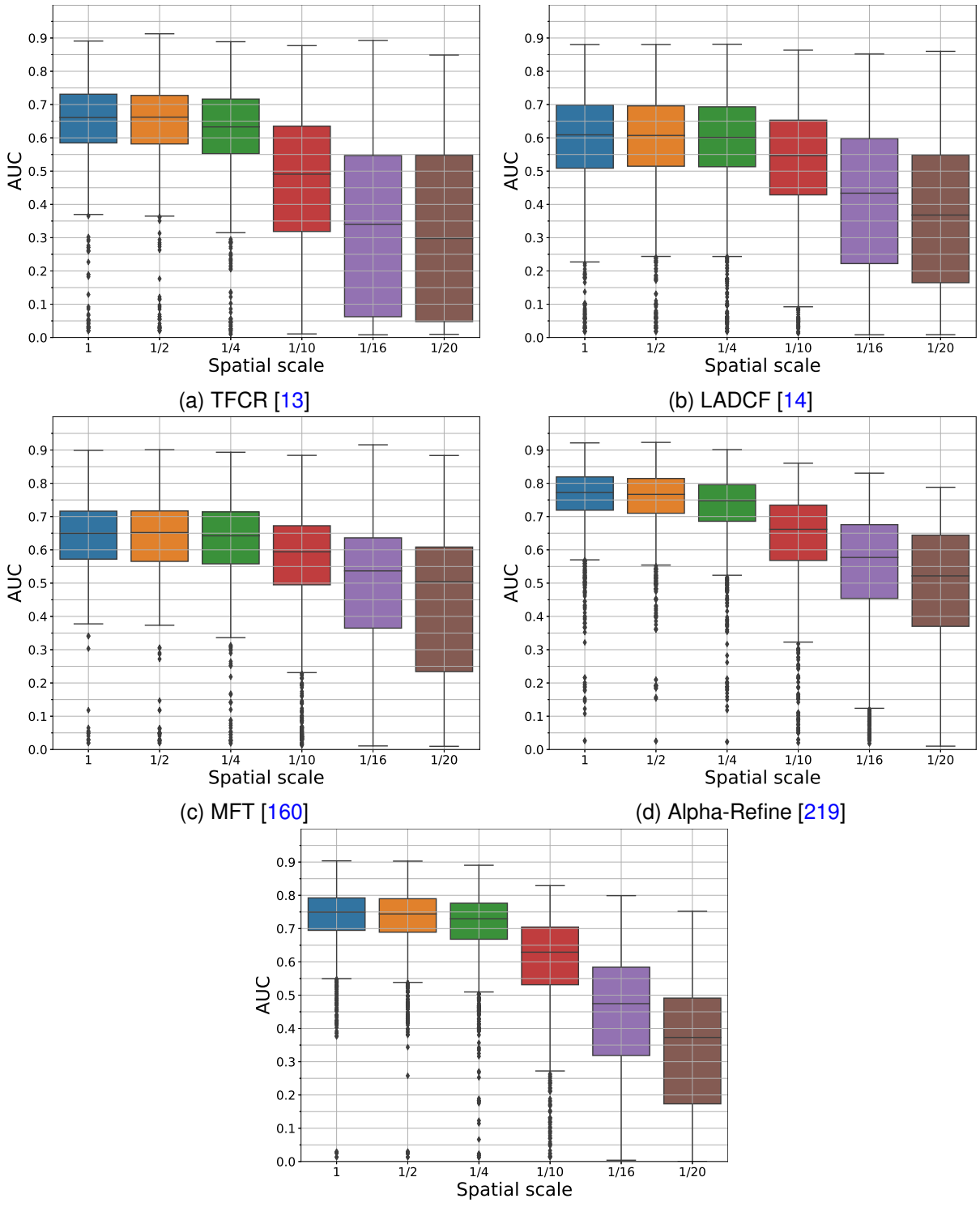
Figure 36 depicts the performance of the trackers when the spatial resolution of each frame in the video was reduced to $1/2$, $1/4$, $1/10$, $1/16$, and $1/20$ of the original resolution. For these experiments, 1385 videos from AD-SVD have been selected so that their frame rate was 30 fps. The video trackers Alpha-Refine [219], SiamRPN++ [220], MFT [160], TFCR [13] and LADCF [14] were selected because they have been recently proposed and delivered state-of-the-art performance at the original video resolution. The results indicate that Alpha-Refine achieves the best performance on all spatial scales tested.

We executed VOT algorithms in different computer environments. For instance, the experiments of LADCF (implemented in MATLAB), SiamRPN++, and Alpha-Refine trackers were accomplished in a computer with the following specifications: processor Intel I7-8750H, 64 GB DDR4-2666 RAM, Disco SSD 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 experiments were conducted on a computer with the following specifications: processor Intel I7-8700K, 40 GB DDR4-2666 RAM, Disco SSD 2 TB, GPU NVIDIA Geforce Titan XP with 12 GB Memory, OS Ubuntu 18.04 LTS. TFCR experiments were carried out in the Frontera Computing System at UT Austin, currently the 10th 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 37 shows the median time required by the trackers to process a video. The videos set to measure times comprises a subset of 140 videos, processed serially, guaranteeing that the computer/node used did not execute other demanding tasks simultaneously. These results show that TFCR and MFT require more time and have lower performance than Alpha-Refine [219] and SiamRPN++ [220].

To train and obtain predictions for the performance loss, a 10-fold cross-validation approach was used. For each 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. The proportions for the training, validation, and test sets, were 75%, 15%, and 10%, respectively.

Figures 38, 39, and 40 depict the results of the framework illustrated in Figure 35 with different values for the threshold. Figure 38e shows that the SiamRPN++ tracker [220] dropped 0.025 in performance on a 1/4 scale, while achieving a 34% time reduction in the total processing time. Figure 39b shows that the TFCR tracker [13] achieves a 65% time reduction with a performance loss of only 0.03 measured by the median AUC. Nonetheless, it is important to consider that the median performance of SiamRPN++ changes from 0.75 to 0.725 at $1/4$ spatial scale, which is still high performance. Meanwhile, at the spatial scale $1/4$, TFCR changes from 0.66 to 0.62, which is comparable to the performance achieved by SiamRPN++ at the spatial scale of $1/10$. TFCR achieves the largest improvement in time reduction of 80% and 84% at spatial scales of $1/10$ and $1/16$, respectively. These results can also be perceived from the drop in the median



(a) TFCR [13]

(b) LADCF [14]

(c) MFT [160]

(d) Alpha-Refine [219]

(e) SiamRPN++ [220]

Figure 36: 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.

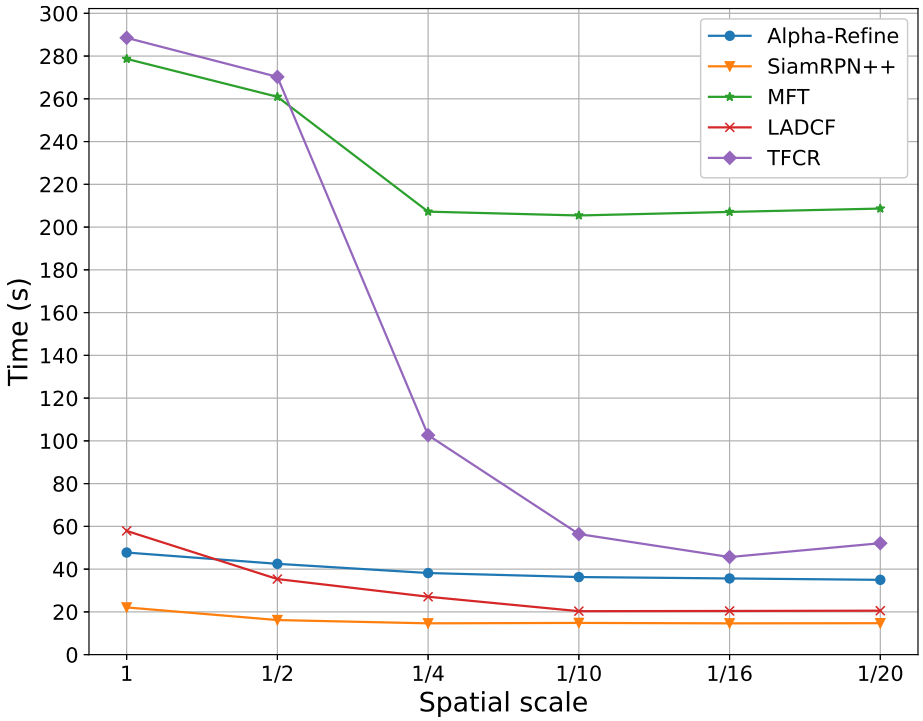
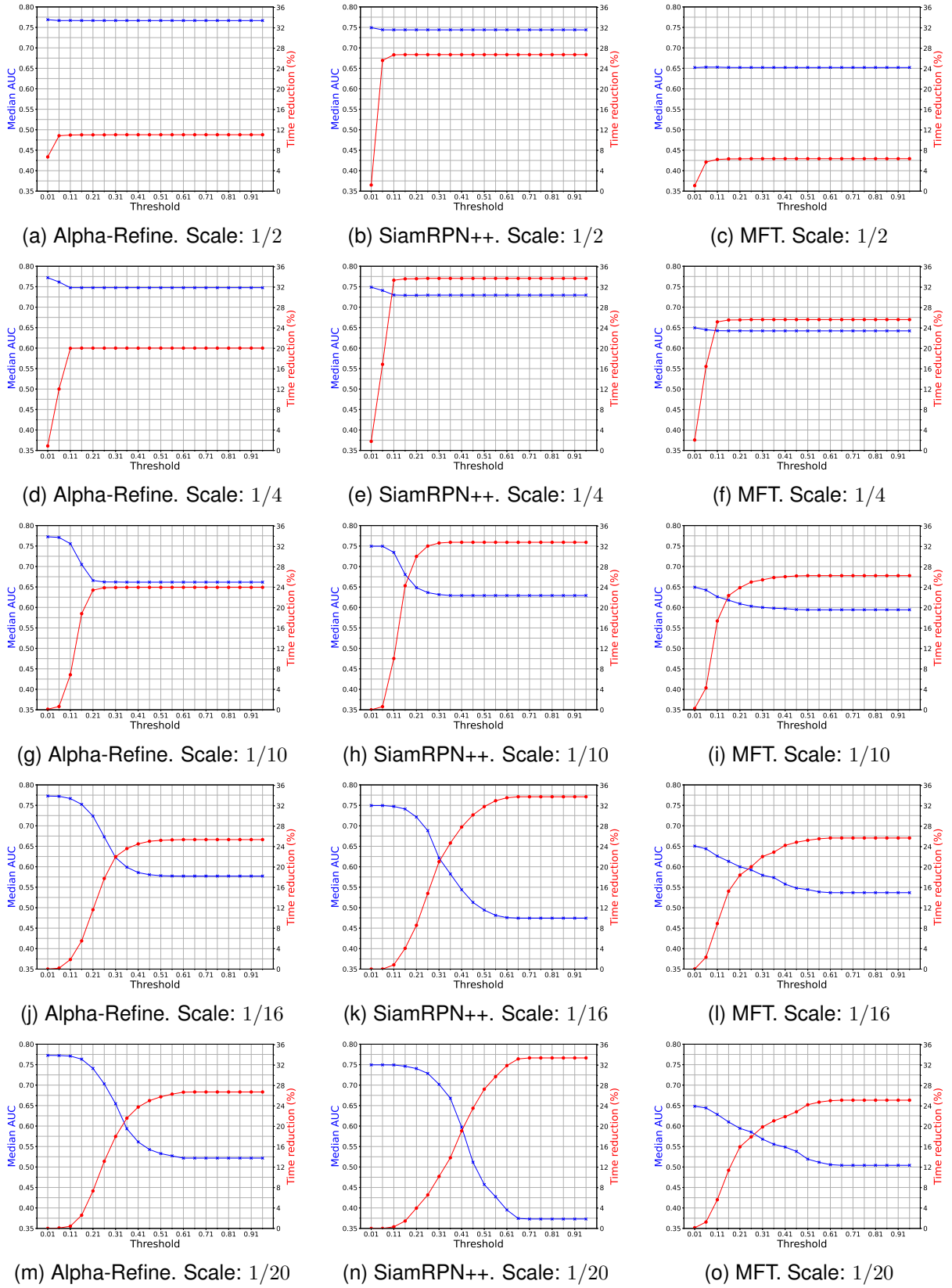


Figure 37: Median time (seconds) required by the trackers to process a video with 450 frames of AD-SVD.

time required per video by TFCR, as depicted in Figure 37. However, the performance drop in the TFCR tracker is 0.16 and 0.31, at the spatial scales $1/10$ and $1/16$, respectively. In general, the best results are obtained between scales $1/4$ and $1/10$ because of requirements imposed by video tracker algorithms on the 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 is a reduction in VOT performance, but not in the time required for VOT algorithm execution.

Conclusions

This chapter proposed a framework to determine the scale with the best trade-off between execution time and VOT algorithm performance. This framework reduces video tracker computational resources, such as time and disk space required for storage, by predicting the performance of the VOT algorithm to determine the optimal spatial scale for processing a video. We achieve this by balancing the processing time and tracking accuracy by predicting the performance in a range of spatial resolutions. Our results indicate that some state-of-the-art



(a) Alpha-Refine. Scale: $1/2$ (b) SiamRPN++. Scale: $1/2$ (c) MFT. Scale: $1/2$

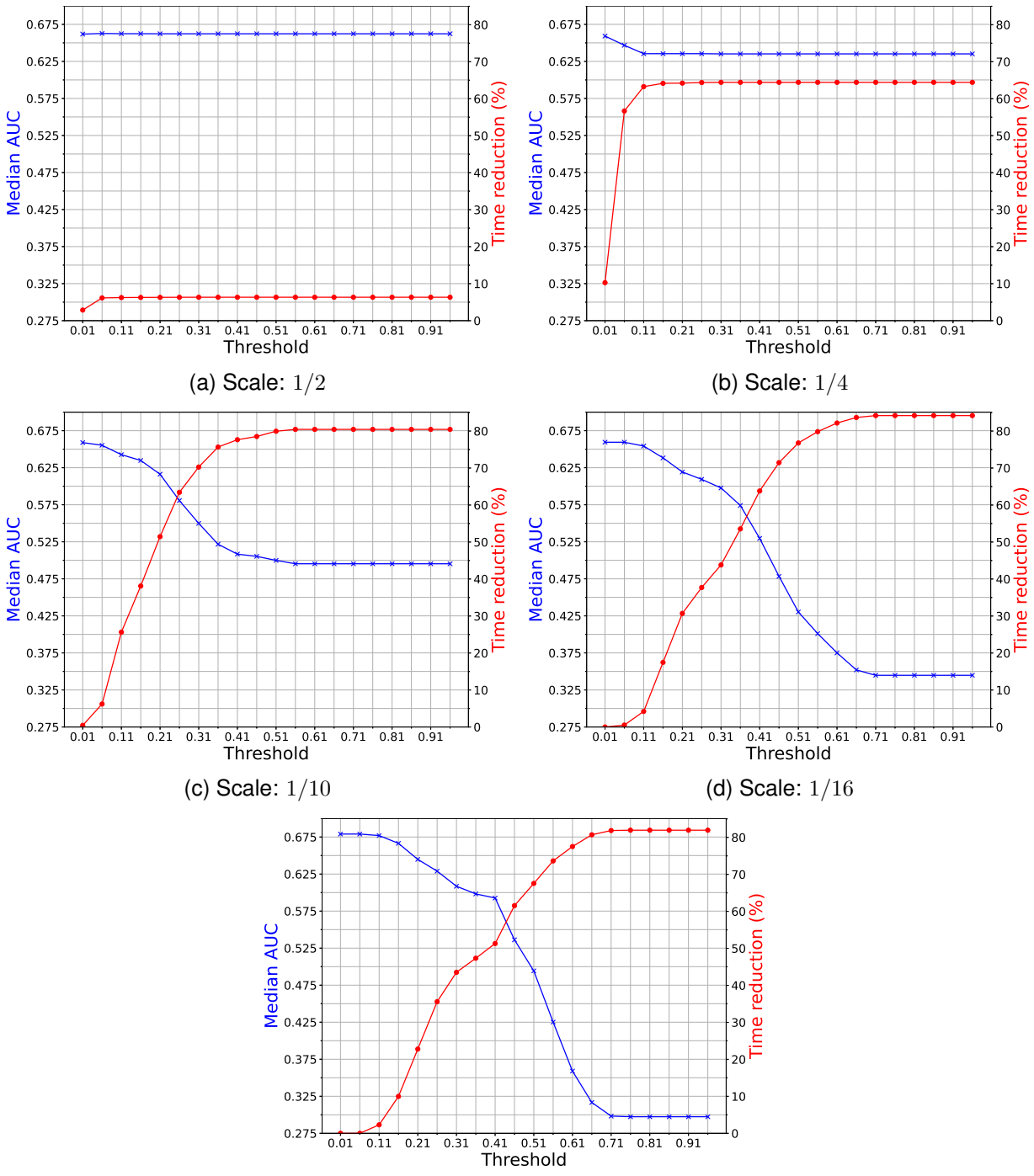
(d) Alpha-Refine. Scale: $1/4$ (e) SiamRPN++. Scale: $1/4$ (f) MFT. Scale: $1/4$

(g) Alpha-Refine. Scale: $1/10$ (h) SiamRPN++. Scale: $1/10$ (i) MFT. Scale: $1/10$

(j) Alpha-Refine. Scale: $1/16$ (k) SiamRPN++. Scale: $1/16$ (l) MFT. Scale: $1/16$

(m) Alpha-Refine. Scale: $1/20$ (n) SiamRPN++. Scale: $1/20$ (o) MFT. Scale: $1/20$

Figure 38: Time reduction vs performance loss.



(a) Scale: $1/2$

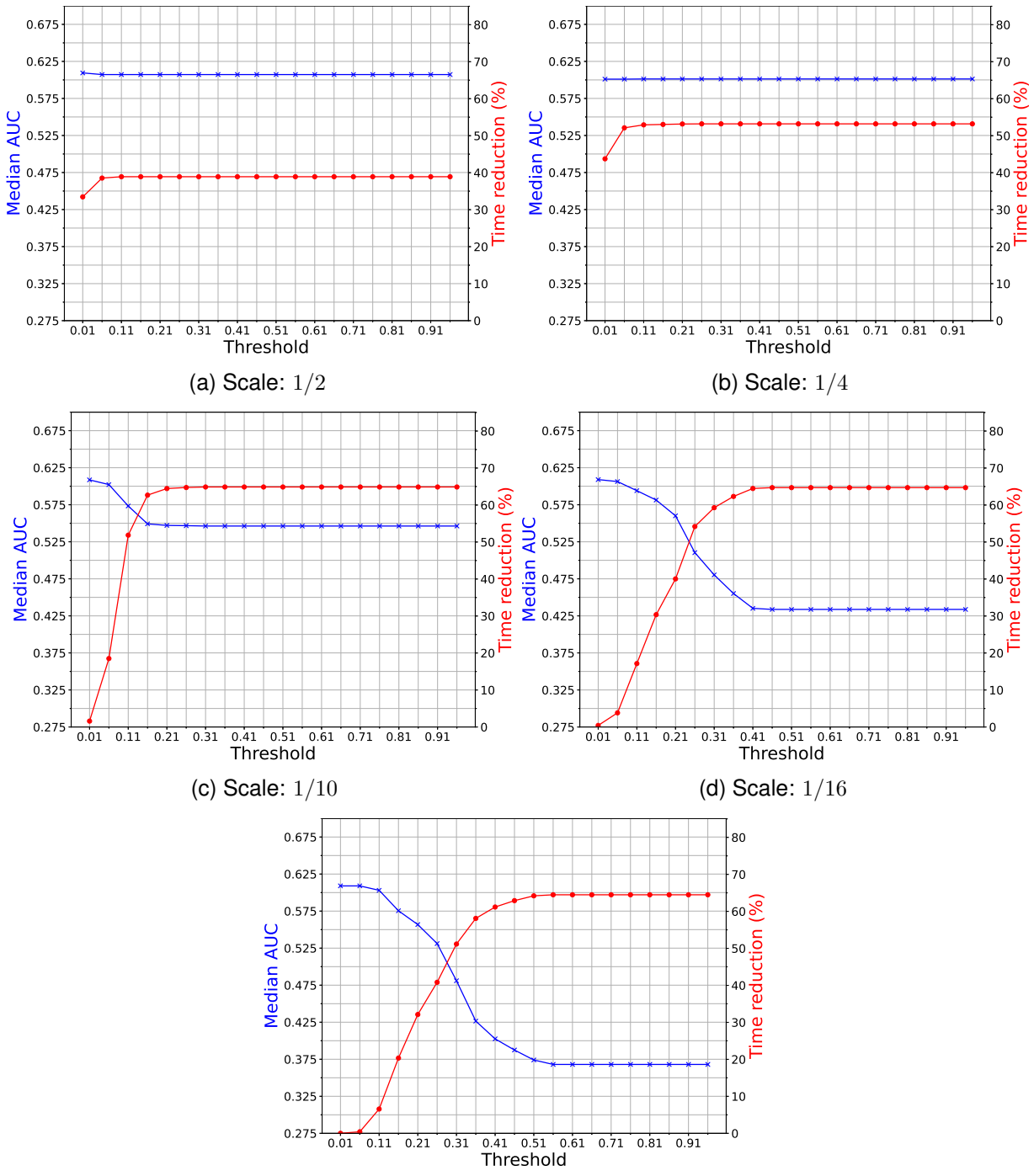
(b) Scale: $1/4$

(c) Scale: $1/10$

(d) Scale: $1/16$

(e) Scale: $1/20$

Figure 39: Time reduction vs performance loss for the tracker TFCR [13].



(a) Scale: $1/2$

(b) Scale: $1/4$

(c) Scale: $1/10$

(d) Scale: $1/16$

(e) Scale: $1/20$

Figure 40: Time reduction vs performance loss for the tracker LADCF [14].

trackers achieve up to a 34% time reduction, with a slight decrease of 3% in tracking performance.

8 Conclusions

We have proposed an NR VQA method, explicitly aimed at videos with natural distortions, such as color, artifacts, exposure, focus, sharpness, and stabilization. Our method is based on a 3D convolutional neural network approach, using features extracted from several layers of the CNN to feed an SVR model to produce an NR VQA model that provides a high level of video quality prediction power. Our VQA method outperforms several state-of-the-art VQA methods, when applied to authentically distorted videos.

Similarly, we carried out an analysis of 11 state-of-the-art trackers. This thesis introduces AD-SVD, a distorted video data set that includes three levels of severity in 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. Therefore, AD-SVD can be considered a solid starting point for studying the influence of distortions on video tracker performance. We demonstrated that in-capture distortions severely affect the performance of state-of-the-art trackers. As expected, the trackers exhibited the best performance for the pristine videos. Therefore, the results reflect the poor performance of the trackers owing to distortions such as underexposure and defocus. In practice, no specific type of distortion consistently generated the worst performance in all scenes and did not affect all trackers in the same manner.

Furthermore, we designed and proposed a framework for robust tracking to in-capture and post-capture distortions. This framework relies on perceptual features to compensate for the impairments produced by these distortions. We analyzed the DLSSVM baseline features and sets of baseline features concatenated with FRIQUEE and TLVQM quality-aware features. Our findings suggest that the TLVQM quality-aware features HC and LC combined conduce to improving tracking performance. Other experiments confirm that z-score normalization must be part of the pre-processing stages intended to improve performance.

Moreover, the time processing of TLVQM features is an average of 0.25 seconds per frame in FHD resolution. Adding to the DLSSVM baseline time, this processing time allows for at least 1 FPS speed, a typical FPS in trackers not intended for real-time. This research provides strong evidence that quality-aware features can improve the robustness of tracking algorithms when are concatenated with baseline features.

Additionally, we 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. The PLCC correlation obtained in predicting performance with some state-of-the-art trackers was superior to 0.90, indicating promising results in prediction capabilities. This precision level can potentially enable the integration of this tracker prediction as a module in a framework where a video can be executed with different trackers, depending on the predicted performance and video characteristics. Similarly, we proposed a framework to determine the best trade-off between time and the tracking performance scale. This proposed framework reduces video tracker computation resources, such as time and disk space required for storage, by predicting the VOT algorithm performance to determine the optimal spatial scale for processing a video. We achieve this by balancing the processing time and tracking accuracy by predicting the performance in a range of spatial resolutions. Our results indicate that some state-of-the-art trackers achieve up to a 34% time reduction, with a slight decrease of 3% in tracking performance.

Overall, limited research has been accomplished in the area of Video Quality applied to tracking tasks. This thesis deepens the understanding of particular topics and issues by providing an experimental perspective. For example, investigations in Video Object Tracking have been predominantly accomplished by scientists who do not necessarily account for the effects of Video Quality attributes in the task. The same applies to computer vision scientists who produce No-Reference Video Quality systems based on subjective human scores, not considering specific task performance. For instance, most NR-VQA datasets created over the years from the computer vision community do not contain in-capture or post-capture distortions in videos with typical video surveillance activities. Similarly, standard distorted datasets are created without controlled-level variations of such distortions. Hence, AD-SVD is proposed as a powerful tool to advance further research on tracking robustness to focus and exposure in-capture distortions.

Finally, we have made progress in the study and connection of two critical areas of computer vision, such as VQA and Tracking. In the last decade, tracking algorithms have been proposed, achieving remarkable results in videos from state-of-the-art databases. However, our results indicate that, when applied to real surveillance videos, which contain mixed in-capture and post-capture distortions, their performance is not maintained and may change dramatically. This performance reduction necessitates the introduction of VQA concepts. Hence, the tracker could

be aware of the quality of the video and adopt strategies and methods depending on the video quality. This quality-aware feature could improve its performance and robustness when applied to real-life videos. In addition, our results show that deep 3D features, extracted from different deep layers of a CNN, have a great capacity for the abstraction of video quality information. When combined with the baseline features of each tracker, these quality-aware features can improve the performance and robustness of the tracking algorithm.

9 Proposals for Future Research

In this section, based on the findings reported, we propose the following directions for further research:

9.1 Time reduction using temporal down-scaling

Tracking objects in real-time plays a critical role in improving the global performance of vision applications. As a fundamental component, tracking algorithms generally require high speed in the real-time handling of video frames. Conventionally, a tracking speed beyond 25 FPS is considered real-time [3, 221]. Recent real-time trackers can be practically categorized into Siamese tracking and CF-based tracking. Siamese networks heavily rely on GPUs, and the running speed on the CPU is approximately $2-3$ FPS [222], as a result of model complexity. CF trackers can achieve real-time speed when using lightweight hand-crafted features such as HOG and ColorNames [159, 161, 171, 205]. However, they typically exhibit an appreciable performance decrease with deep CF trackers. Equipped with CNN features, CF trackers achieve state-of-the-art tracking accuracy but suffer from a high computational load. Methods of feature dimension reduction, such as PCA, can reduce the feature complexity. However, these methods must first extract high-dimensional CNN features from the original deep models [223]. Hence, to improve the tracking speed, it is necessary to reduce the time required to compute deep features used in Siamese networks or CF- Deep combined trackers. One manner of doing this is to reduce the number of images that the CNN network must infer to extract deep features, reducing the overall video frames to process for tracking framework.

Despite its importance and multiple applications, real-time trackers are difficult to implement. This is proved by the fact that many state-of-the-art trackers do not exhibit real-time performance, even with their exceptional accuracy. The FPS rates for some of these trackers, evaluated on the OTB-2013 [224] and VOT-2018 [109] datasets, are: SRDCF= 4.3 [168], MCPF = 1.5 [225], C-COT = 0.7 [170], MCCT=2.8 [159], ECO=2.4 [164], LADCF=1.2 [14], MTSCF = 1.8 [226], SRCT=1.2 [163], MCOT=6.58 [227]. As can be noted, there is an extensive opportunity to improve the speed of these trackers, intended to be suitable for real-time tasks. However, not every tracker is adequate to work well with low-frame rate videos. Some video trackers that use motion features cannot work well with this method as a result of these features

cannot be estimated for low-frame FPS videos [228, 229]. In addition, decreasing temporal resolution has been shown to affect negative object tracking performance. As a hopefully finding, in [230] the authors reported that object tracking could operate on a video with an FPS ≥ 8 without a significant reduction in its performance.

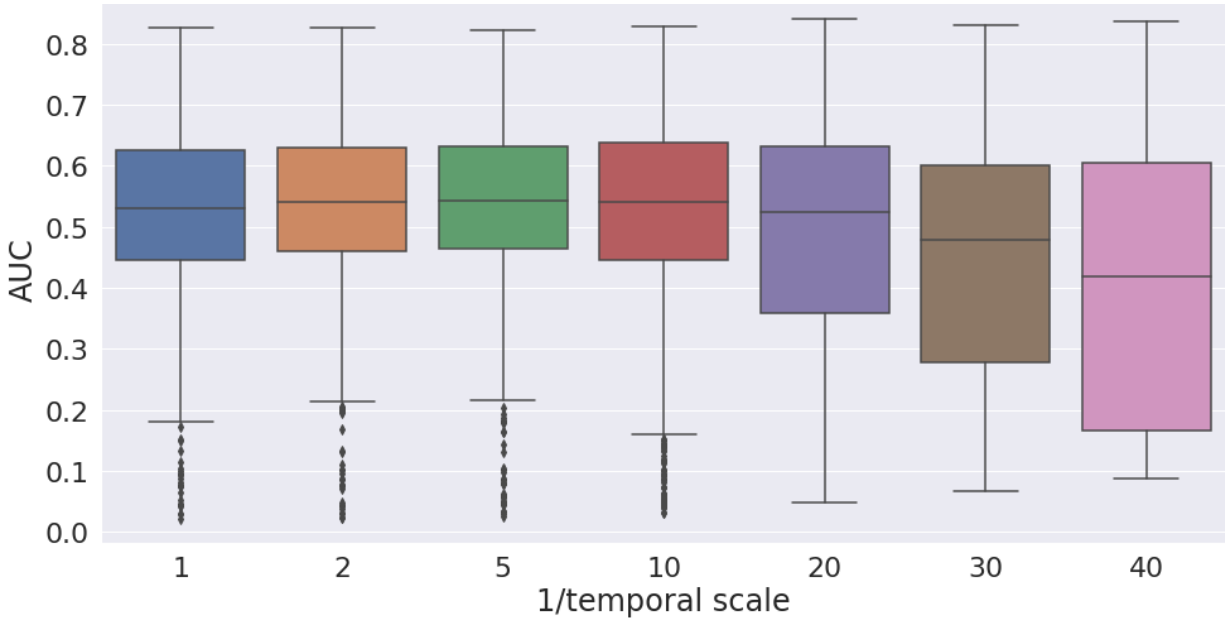


Figure 41: Temporal resolution analysis: AUC performance at different temporal scales.

As an exploratory experiment, we applied the DLSSVM tracker to 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. The DLSSVM initially had a mean FPS of 5.4 [156]. Figure 41 shows that the performance does 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. Another benefit of this method is that having the DLSSVM tracker 5 FPS, and increasing it by ten times, would turn it into a real-time performance tracker, with a minimum accuracy reduction.

Surveillance systems are commonly recorded at 5 FPS and below to reduce the storage requirements. Because of this, low-frame-rate videos are common, as many 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 [228]. 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. Accordingly, reducing the standard frame rate increases the possibility of missing important information from the original video sequence. Substantially, if an object appears in three frames within a second when a 25 FPS rate, the reduction to 1 FPS will result in a significant decrease in the probability of finding this object in one selected frame [231]. Hence, further research is needed to expand and test these preliminary findings on other state-of-the-art trackers on surveillance activities videos.

9.2 Deblurring prior to Video Object Tracking

The occurrence of motion blur along with defocus blur is a common phenomenon in natural images. Blur can be generated in an image by multiple factors such as pixel recording lights from different sources [232], camera shake or fast object movements, or lens defocusing [233]. Three common types of blur are defocus, motion, and Gaussian. The optical theory reveals that the out-of-focus (defocus) blur combined lens defocusing and light diffraction. The defocus blur of interest in this thesis occurs in the imaging process when the image plane is away from the ideal reference plane [234], which is something common in recording surveillance activities.

Some studies have identified the impact of motion blur on tracking task [3, 235]. Recently, an evaluation of the tracking performance of highly blurred video has been proposed as a new baseline for tracking algorithms [7, 164]. Lately, several studies have suggested solutions to blur challenges in tracking performance in different tasks such as face recognition [236] and 3D hand tracking [237]. Ubiquitous image blur degradation usually leads to difficulties in object identification tasks, such as object tracking. In Chapter 3, we demonstrated quantitatively that defocus negatively affects the tracking performance of 11 state-of-the-art trackers.

Since the masterly proposal by Fergus to remove blur in photography in 2006 [238], this area has been an active field of research in Computer Vision related tasks. In recent years, significant progress has been achieved in deblurring methods. In the short term, they can be applied to various camera devices as a pre-processing module to improve the perceptive visual quality. Current Non-Reference image deblurring methods are classified into three main categories: i) the parametric method [239], where the kernel is estimated from the image and later applied to deblur the original image, ii) the non-parametric method [240] which estimates the blur kernel and image simultaneously, and iii) the learning-based method [241], which trains a network to

deblur the image [242]. Liu et al. [234] proposed a defocused image deblurring method that was modeled with a generalized Gaussian (GG) function. This deblurring method computes the pristine image using the obtained blur kernel and a non-blind image deblurring algorithm. Liu's method obtains state-of-the-art results and is a candidate for the deblurring tracker framework outlined. Nonetheless, despite these advances, deblurring tasks remain an ill-posed nature problem, and hence, it isn't easy to find an effective and universal solution to this problem [234].

One decisive factor to consider is the processing time for the deblurring algorithm. Despite the good results in image quality of proposals such as [242, 243], the computational times reported of 8 and 4 seconds, respectively, make them unsuitable for integration into a tracking framework with at least one (1) FPS as the goal. However, some recent proposals obtain exciting time results in deblurring defocus images with a size of 1000×667 pixels in less than 0.5 seconds, measured in a computer with modest computing capabilities [234]. This reduced time makes it possible to select some video frames and apply one deblurring algorithm, intended to obtain a tracking processing rate of at least 1 FPS, making it competitive with some state-of-the-art trackers in no real-time domain. To further reduce this processing time, deblurring can be applied in a Region of Interest (ROI) surrounding the last known position of the object. The size of this ROI can be determined using methods to determine the intensity of the target movement [244]. Thus, the image size for deblurring can be decreased, reducing the deblur processing time.

It can be a naive approximation to deblur all frames in a video, searching to improve the tracking performance. One of the reasons is that the time needed to deblur all frames in the video is considerable and can even be greater than the tracking time. Another reason is that the deblurring process introduces the Gibbs phenomenon, which is seen as ringing artifacts in the image, consequently decreasing the tracking performance. The proposed AD-SVD dataset is a unique tool for performing this blur-related research. AD-SVD allows the quantitative evaluation of trackers' robustness to different levels of defocus, supporting deep explorations of how to defocus change tracking performance. Hence, a system is needed to determine which frames will be deblurred to improve the tracking performance. This expert system could be based on an SVR Regressor, trained with some quality-aware features, especially Deep Perceptual Features [245], which has presented promising results in quality-image related tasks.

9.3 Distortion Un-aware Model based on Deep Features

An important direction of research on the blind video quality assessment (BVQA) problem is to build perceptual models that can predict the quality of distorted images with as little prior knowledge of the distorted videos or their distortions as possible. The VQA method proposed in this thesis requires knowledge of anticipated distortions in the form of training examples and corresponding human opinion scores. Similarly, the existing NR-VQA metrics are primarily aimed at distortion-specific problems, such as compression and transmission distortions [225]. Therefore, similar to the approach presented in NIQE [74], as a future direction of research, we propose a completely blind metric to measure the naturalness of authentically distorted videos by calculating the geometric distance between a pristine model and an input video. The pristine and input video models were built by fitting learned deep spatio-temporal features with Gaussian mixture distributions. The features from the input videos can be learned by using the dataset KonVid-150K [88]. Pristine videos can be obtained from open source websites such as the LIVE Mobile database [185], EPFL-PoliMI video quality assessment database [80] and INRS Audiovisual Quality Dataset [246].

10 Contributions

This research has contributed with original knowledge by:

1. Providing an open-source authentically Distorted Surveillance Videos Dataset (AD-SVD)[5]. AD-SVD allows to test Video Trackers algorithms in videos with authentic distortions in quantified levels. AD-SVD comprises more than 4476 videos affected by in-capture distortions, acquired by four different surveillance cameras and recorded at three outdoor and four indoor locations. To the best of our knowledge, AD-SVD is the largest, densely annotated, and authentically distorted video object tracking benchmark for STT.

2. Presenting a benchmark of 11 state-of-the-art trackers of VOT2017 and VOT-2018 challenges. These trackers were evaluated on the AD-SVD dataset using the success rate as a performance measure. The data indicating that the best-ranked trackers in the VOT 2018 and 2017 contests do not demonstrate the best performance in the AD-SVD dataset implies that a new benchmark baseline is necessary for evaluating the trackers on authentically distorted videos. This benchmark is a valuable contribution because it allows the analysis of VOT algorithm performance on videos affected by authentic in-capture and post-capture distortions, considering that the real videos coming from the cities' surveillance cameras are saturated with in-capture distortions. Hence, it is essential to guarantee the proposed tracking algorithms' applicability and test them in videos with authentic distortions.

3. Developing an original No-Reference VQA Method. This framework is fast and does not longer requires the extraction of handcrafted features. We extracted the convolutional features of the 3-D C3D Convolutional Neural Network and trained a Support Vector Regressor to obtain a VQA score. We carried out transformations to different color spaces to generate discriminant deep features. We extracted features from several layers, with and without overlap, to find the best configuration for improving the VQA score. We tested on the LIVE-Qualcomm dataset. We evaluated the perceptual quality prediction model extensively, obtaining a final correlation of 0.7749 ± 0.0884 with Human Opinion Scores. It shows that it can achieve good video quality prediction, outperforming other state-of-the-art VQA leading models.

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

4. A framework is proposed to improve tracker robustness against in-capture and post-capture distortions. We present the design and implementation of a quality-aware feature selection for VOT. First, we divided each video frame into patches of the same size and extracted the HOG and NSS features from these patches. Furthermore, we defined the best features HOG and NSS that generate the most significant area under the curve in the success plots, yielding an improvement in the video tracker performance in videos affected by post-capture distortions. We proposed an approach to integrate NSS perceptual quality features into a video object tracker scheme and demonstrated its performance in several videos affected by post-capture distortions. This methodology is the first work to propose a quality-aware feature extraction approach for VOT to the best of our knowledge.

5. Designing a model-agnostic (independent of the tracker model) framework that predicts performance without running the corresponding tracking algorithm. We developed 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 in such a way that the probability of obtaining the reference output is maximized without executing the tracking algorithms. To this end, we learn a mapping between the input video and the area under the curve (AUC) of the success-plot. This process is carried out in two stages: i) extraction of a fixed-size set of features, and ii) AUC estimation using a support vector machine regression model. 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.

6. Introducing a framework to reduce video tracker computation resources (time and video storage space). We achieve this by balancing the processing time and tracking accuracy by predicting the performance in a range of spatial resolutions. This time reduction 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 tracker's video storage demands and execution time, thereby preserving its performance. We concluded that We obtained the best results on scales $1/4$ and $1/10$ owing to the requirements imposed by the video tracker algorithms on the input video resolution. A scale smaller than $1/4$ or $1/10$, depending on the tracker, does

not imply a considerable reduction in the median time required per video. Hence, there may be a reduction in VOT performance but not in the time required for VOT algorithm execution. Finally, this approach can reduce the execution time by up to 34% with a slight decrease in performance of 3%.

11 Publications

The development of this thesis has produced the following publications:

Journal Articles:

- R. Gomez-Nieto, J.F Ruiz-Muñoz, Juan Beron, Cesar Ardila, Hernan Benitez, Alan Bovik, ”Quality Aware Features for Performance Prediction and Time Reduction in Video Object Tracking”, submitted on 12-Aug-2021 in IEEE Access, Under Review.

Proceedings Articles:

- R. Gomez-Nieto, H. D. Benitez. Restrepo, and I. Cabezas, “How video object tracking is affected by in-capture distortions?” in ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019, pp. 2227–2231, doi: 10.1109/ICASSP.2019.8683625.
- R. Gomez-Nieto, H. D. Benitez-Restrepo, and J. F. Ruiz-Munoz,“Quality aware feature selection for video object tracking”, Electronic Imaging, vol. 2020, no. 9, pp. 169–1–169–7, 2020, doi:10.2352/ISSN.2470-1173.2020.9.IQSP-169.
- R. Gomez-Nieto, H. D. Benitez-Restrepo, R. F. Quintero et al.,“No reference video quality assessment with authentic distortions using 3-d deep convolutional neural network,” Electronic Imaging, vol. 2020, no. 9, pp. 168–1–168–7, 2020, doi: 10.2352/ISSN.2470-1173.2020.9.IQSP-168.

12 Acknowledgments

Successes never happen individually, and this thesis was made possible due to the support received from numerous people. First and foremost, I want to extend my genuine acknowledgment to my advisor Prof. Hernan Benitez. He has deeply inspired me with his brilliant research abilities and his strive for excellence. His intellect, guidance, and discipline have helped me become a better researcher and person. That, I believe, is an accurate indication of a good advisor. He is incredibly hardworking and disciplined about research and sets an example of distinction as a Ph.D. advisor, professor, and scientist.

Similarly, I would like to thank my co-advisor, Prof. Alan Bovik, from UT Austin. He was extremely welcoming and friendly during my internship at LIVE Laboratory. Their advice was valuable into finding new research paths and methods. Prof. Bovik is a well-recognized researcher, and he demonstrated this extensive knowledge with each piece of advice that he gave me, very helpful in obtaining the thesis results.

I also want to thank Dr. Jose Ruiz from the University of Florida - National University of Colombia, who gave me his advice and shared valuable knowledge in the area of VOT. We also thank the Electronic Engineering students Santiago Bonilla, Cesar Ardila, Alejandro Ledesma, and Stidl Torres for their practical help in recording the AD-SVD dataset videos (Chapter 3) and carrying out extensive experiments that involved testing state-of-the-art VQA algorithms on hundreds of videos of several existing VQA-NR datasets (Chapter 4).

I also want to thank the Pontificia Universidad Javeriana Cali for providing a tuition scholarship for Ph.D. studies. Similarly, thanks to the Doctorate in Engineering and Applied Sciences program, especially its Director Dr. Andres Jaramillo and Assistant Andrea Ramirez, for all the support given in the administrative processes required and financial support in the UT Austin six-month internship.

We acknowledge the funding provided by Minciencias Colombia and Pontificia Universidad Javeriana with the project *Vigilancia Inteligente para la red de cámaras de la Policía Metropolitana de Cali*. We would like to thank NVIDIA Corporation for donating a TITAN XP GPU used in the experiments. We acknowledge 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.

Thanks to all my friends and colleagues at Universidad Javeriana. It was a privilege the get to know each one of them. Mainly, I would like to thank Jenny Gallo, Isabella Lopez, Andrea Ramirez, Juan Manuel Marmolejo, Jan Medina, Diego Ruiz, Juan Pablo Beron, Juan Carlos Romero, Miguel Romero, Juan Camilo Campos, Sergio Ramirez, Pedro Hernandez, Nicolas Lopez, for their friendship and support during this process. Besides, I am very grateful to Dr. Camilo Rocha, Dr. Luis Tobon, and Dr. Eugenio Tamura for their valuable teaching during the doctoral courses. There is no doubt that the knowledge learned under their supervision was crucial to completing this Ph.D. Thesis.

Lastly, but no least, I want to thank my family for their love and support. Special thanks to Ignacio Gomez, Camilo Gomez, Nancy Gomez, Blanca Marin, Consuelo Rondon and Maria Nieto, for their never-ending love, support and patience.

Dedicated to my parents

Maria Elena and Ignacio

13 References

[1] S. Hadfield, K. Lebeda, and R. Bowden, “The visual object tracking vot2014 challenge results,” 2014.

[2] M. Kristan, A. Leonardis, J. Matas *et al.*, “The Eighth Visual Object Tracking VOT2020 Challenge Results BT ,” A. Bartoli and A. Fusiello, Eds. Cham: Springer International Publishing, 2020, pp. 547–601, doi: 10.1007/978-3-030-68238-5_39.

[3] Y. Wu, J. Lim, and M.-H. Yang, “Object tracking benchmark,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1834–1848, 2015, doi: 10.1109/TPAMI.2014.2388226.

[4] M. Mueller, N. Smith, and B. Ghanem, “A benchmark and simulator for uav tracking,” in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe *et al.*, Eds. Cham: Springer International Publishing, 2016, pp. 445–461, doi: 10.1007/978-3-319-46448-0.

[5] P. Liang, E. Blasch, and H. Ling, “Encoding color information for visual tracking: Algorithms and benchmark,” *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5630–5644, 2015, doi: 10.1109/TIP.2015.2482905.

[6] A. Li, M. Lin, Y. Wu *et al.*, “Nus-pro: A new visual tracking challenge,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 335–349, 2016, doi: 10.1109/TPAMI.2015.2417577.

[7] H. Fan, L. Lin, F. Yang *et al.*, “Lasot: A high-quality benchmark for large-scale single object tracking,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 5369–5378, doi: 10.1109/CVPR.2019.00552.

[8] I. Bezzine, Z. A. Khan, A. Beghdadi *et al.*, “Video quality assessment dataset for smart public security systems,” in *2020 IEEE 23rd International Multitopic Conference (INMIC)*, 2020, pp. 1–5, doi: 10.1109/INMIC50486.2020.9318149.

[9] M. Müller, A. Bibi, S. Giancola *et al.*, “TrackingNet: A Large-Scale Dataset and Benchmark for Object Tracking in the Wild,” in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu *et al.*, Eds. Cham: Springer International Publishing, 2018, pp. 310–327, doi: 10.1007/978-3-030-01246-5_19.

[10] 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*, vol. 43, no. 5, pp. 1562–1577, 2021, doi: 10.1109/TPAMI.2019.2957464.

[11] C. A. Ardila Franco and B. V. S, "Benchmarking of state-of-the-art single object video trackers in authentically distorted videos-Pontificia Universidad Javeriana, Cali," Pontificia Universidad Javeriana Colombia, Tech. Rep., 2020.

[12] D. Tran, L. Bourdev, R. Fergus *et al.*, "Learning spatiotemporal features with 3d convolutional networks," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 4489–4497, doi: 10.1109/ICCV.2015.510.

[13] D. Yuan, N. Fan, and Z. He, "Learning target-focusing convolutional regression model for visual object tracking," *Knowledge-Based Systems*, vol. 194, p. 105526, 2020, doi: 10.1016/j.knosys.2020.105526.

[14] T. Xu, Z.-H. Feng, X.-J. Wu *et al.*, "Learning adaptive discriminative correlation filters via temporal consistency preserving spatial feature selection for robust visual object tracking," *IEEE Transactions on Image Processing*, vol. 28, no. 11, pp. 5596–5609, 2019, doi: 10.1109/TIP.2019.2919201.

[15] D. Ghadiyaram, J. Pan, A. C. Bovik *et al.*, "In-capture mobile video distortions: A study of subjective behavior and objective algorithms," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 9, pp. 2061–2077, 2018, doi: 10.1109/TCSVT.2017.2707479.

[16] a. D. G. C. vivotek, "Ip8165hp," https://www.vivotek.com/ip8165hp, 2021, [Online; accessed 25-August-2021].

[17] ——, "Ib8367a," https://www.vivotek.com/ib8367a, 2021, [Online; accessed 25-August-2021].

[18] ——, "Ib8381," https://www.vivotek.com/ib8381, 2021, [Online; accessed 25-August-2021].

[19] A. Communications, "Axis p1435-le network camera," https://www.axis.com/products/axis-p1435-le/support, 2021, [Online; accessed 25-August-2021].

[20] T. Nawaz and A. Cavallaro, “A protocol for evaluating video trackers under real-world conditions,” *IEEE Transactions on Image Processing*, vol. 22, no. 4, pp. 1354–1361, 2013, doi: 10.1109/TIP.2012.2228497.

[21] A. W. M. Smeulders, D. M. Chu, R. Cucchiara *et al.*, “Visual tracking: An experimental survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1442–1468, 2014, doi: 10.1109/TPAMI.2013.230.

[22] B. Karasulu and S. Korukoglu, “A software for performance evaluation and comparison of people detection and tracking methods in video processing,” *Multimedia Tools and Applications*, vol. 55, no. 3, pp. 677–723, 2011, doi: 10.1007/s11042-010-0591-2.

[23] D. M. Chu and A. W. Smeulders, “Thirteen hard cases in visual tracking,” in *2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2010, pp. 103–110, doi: 10.1109/AVSS.2010.85.

[24] K. Seshadrinathan and A. Bovik, *The essential guide to video processing*. Elsevier, 2009, ch. Video quality assessment, pp. 417–436.

[25] L.-H. Chen, C. G. Bampis, Z. Li *et al.*, “Perceptual video quality prediction emphasizing chroma distortions,” *IEEE Transactions on Image Processing*, vol. 30, pp. 1408–1422, 2021, doi: 10.1109/TIP.2020.3043127.

[26] M. Pinson and S. Wolf, “A new standardized method for objectively measuring video quality,” *IEEE Transactions on Broadcasting*, vol. 50, no. 3, pp. 312–322, 2004, doi: 10.1109/TBC.2004.834028.

[27] S. Chikkerur, V. Sundaram, M. Reisslein *et al.*, “Objective video quality assessment methods: A classification, review, and performance comparison,” *IEEE Transactions on Broadcasting*, vol. 57, no. 2, pp. 165–182, 2011, doi: 10.1109/TBC.2011.2104671.

[28] W. Geisler and M. Banks, *Handbook of optics*. Mc Graw Hill, 1995, ch. Visual performance.

[29] C. J. V. den Branden Lambrecht and O. Verscheure, “Perceptual quality measure using a spatiotemporal model of the human visual system,” in *Digital Video Compression: Algorithms and Technologies 1996*, V. Bhaskaran, F. Sijstermans, and S. Panchanathan, Eds., vol. 2668, International Society for Optics and Photonics. SPIE, 1996, pp. 450–461, doi: 10.1117/12.235440.

[30] A. B. Watson, Q. J. Hu, and J. F. M. III, “Digital video quality metric based on human vision,” *Journal of Electronic Imaging*, vol. 10, no. 1, pp. 20–29, 2001, doi: 10.1117/1.1329896.

[31] Z. Wang, A. Bovik, H. Sheikh *et al.*, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004, doi: 10.1109/TIP.2003.819861.

[32] A. Pessoa, A. Falcão, R. Nishihara *et al.*, “Video quality assessment using objective parameters based on image segmentation,” *SMPTE Journal*, vol. 108, no. 12, pp. 865–872, 1999, doi: 10.5594/J04308.

[33] J. Okamoto, T. Hayashi, A. Takahashi *et al.*, “Proposal for an objective video quality assessment method that takes temporal and spatial information into consideration,” *Electron. Commun Japan*, vol. 89, pp. 97–108, 2006, doi: 10.1002/ecja.20265.

[34] Y. Liu, J. Wu, A. Li *et al.*, “Video quality assessment with serial dependence modeling,” *IEEE Transactions on Multimedia*, pp. 1–1, 2021, doi: 10.1109/TMM.2021.3107148.

[35] K. Seshadrinathan and A. C. Bovik, “Motion tuned spatio-temporal quality assessment of natural videos,” *IEEE Transactions on Image Processing*, vol. 19, no. 2, pp. 335–350, 2010, doi: 10.1109/TIP.2009.2034992.

[36] Z. W. D.V.M. and E. P. Simoncelli, “Reduced-reference image quality assessment using a wavelet-domain natural image statistic model,” in *Human Vision and Electronic Imaging X*, B. E. Rogowitz, T. N. Pappas, and S. J. Daly, Eds., vol. 5666, International Society for Optics and Photonics. SPIE, 2005, pp. 149–159, doi: 10.1117/12.597306.

[37] Q. Li and Z. Wang, “Reduced-reference image quality assessment using divisive normalization-based image representation,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 2, pp. 202–211, 2009, doi: 10.1109/JSTSP.2009.2014497.

[38] R. Soundararajan and A. C. Bovik, “Rred indices: Reduced reference entropic differencing framework for image quality assessment,” in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 1149–1152, doi: 10.1109/ICASSP.2011.5946612.

[39] M. Masry, S. Hemami, and Y. Sermadevi, “A scalable wavelet-based video distortion metric and applications,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 2, pp. 260–273, 2006, doi: 10.1109/TCSVT.2005.861946.

[40] X. Yu, N. Birkbeck, Y. Wang *et al.*, “Predicting the quality of compressed videos with pre-existing distortions,” *IEEE Transactions on Image Processing*, pp. 1–1, 2021, doi: 10.1109/TIP.2021.3107213.

[41] S. Mitra, R. Soundararajan, and S. S. Channappayya, “Predicting spatio-temporal entropic differences for robust no reference video quality assessment,” *IEEE Signal Processing Letters*, vol. 28, pp. 170–174, 2021, doi: 10.1109/LSP.2021.3049682.

[42] M. Banitalebi-Dehkordi, A. Ebrahimi-Moghadam, M. Khademi *et al.*, “No-reference video quality assessment based on visual memory modeling,” *IEEE Transactions on Broadcasting*, vol. 66, no. 3, pp. 676–689, 2020, doi: 10.1109/TBC.2019.2957670.

[43] W. Wu, Q. Li, Z. Chen *et al.*, “Semantic information oriented no-reference video quality assessment,” *IEEE Signal Processing Letters*, vol. 28, pp. 204–208, 2021, doi: 10.1109/LSP.2020.3048607.

[44] J.-M. Moreno-Roldán, J. Poncela, P. Otero *et al.*, “A no-reference video quality assessment model for underwater networks,” *IEEE Journal of Oceanic Engineering*, vol. 45, no. 1, pp. 342–346, 2020, doi: 10.1109/JOE.2018.2869441.

[45] S. V. Reddy Dendi and S. S. Channappayya, “No-reference video quality assessment using natural spatiotemporal scene statistics,” *IEEE Transactions on Image Processing*, vol. 29, pp. 5612–5624, 2020, doi: 10.1109/TIP.2020.2984879.

[46] B. Chen, L. Zhu, G. Li *et al.*, “Learning generalized spatial-temporal deep feature representation for no-reference video quality assessment,” *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2021, doi: 10.1109/TCSVT.2021.3088505.

[47] J. Y. Yao and G. Liu, “Bitrate-based no-reference video quality assessment combining the visual perception of video contents,” *IEEE Transactions on Broadcasting*, vol. 65, no. 3, pp. 546–557, 2019, doi: 10.1109/TBC.2018.2878360.

[48] Y. Zhang, X. Gao, L. He *et al.*, “Blind video quality assessment with weakly supervised learning and resampling strategy,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 8, pp. 2244–2255, 2019, doi: 10.1109/TCSVT.2018.2868063.

[49] M. Agarla, L. Celona, and R. Schettini, “No-Reference Quality Assessment of In-Capture Distorted Videos,” *Journal of Imaging*, vol. 6, no. 8, 2020, doi: 10.3390/jimaging6080074.

[50] T. Brandao and M. P. Queluz, “No-reference quality assessment of h.264/avc encoded video,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 11, pp. 1437–1447, 2010, doi: 10.1109/TCSVT.2010.2077474.

[51] S. Daly, *Vision Models and Applications to Image and Video Processing*. Springer-Verlag, 2001, ch. Engineering observations from spatiovelocity and spatiotemporal visual models, pp. 179–200.

[52] A. Mittal, A. K. Moorthy, and A. C. Bovik, “Blind/referenceless image spatial quality evaluator,” in *2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, 2011, pp. 723–727, doi: 10.1109/ACSSC.2011.6190099.

[53] M. A. Saad, A. C. Bovik, and C. Charrier, “A dct statistics-based blind image quality index,” *IEEE Signal Processing Letters*, vol. 17, no. 6, pp. 583–586, 2010, doi: 10.1109/LSP.2010.2045550.

[54] H. Tang, N. Joshi, and A. Kapoor, “Learning a blind measure of perceptual image quality,” in *CVPR 2011*, 2011, pp. 305–312, doi: 10.1109/CVPR.2011.5995446.

[55] P. Ye and D. Doermann, “No-reference image quality assessment using visual codebooks,” *IEEE Transactions on Image Processing*, vol. 21, no. 7, pp. 3129–3138, 2012, doi: 10.1109/TIP.2012.2190086.

[56] M. T. Vega, D. C. Mocanu, and J. e. a. Famaey, “Deep learning for quality assessment in live video streaming,” *IEEE Signal Processing Letters*, vol. 24, no. 6, pp. 736–740, 2017, doi: 10.1109/LSP.2017.2691160.

[57] V. Frants, V. Voronin, A. Zelenskiy *et al.*, “Blind visual quality assessment for smart cloud-based video storage,” in *2018 IEEE International Conference on Smart Cloud (SmartCloud)*, 2018, pp. 171–174, doi: 10.1109/SmartCloud.2018.00036.

[58] M. Alizadeh, A. Mohammadi, and M. Sharifkhani, “No-reference deep compressed-based video quality assessment,” in *2018 8th International Conference on Computer and Knowledge Engineering (ICCKE)*, 2018, pp. 130–134, doi: 10.1109/ICCKE.2018.8566395.

[59] Y. Fazliani, E. Andrade, and S. Shirani, “Learning based hybrid no-reference video quality assessment of compressed videos,” in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2019, pp. 1–5, doi: 10.1109/ISCAS.2019.8702584.

[60] N. Barman, E. Jammeh, S. A. Ghorashi *et al.*, “No-reference video quality estimation based on machine learning for passive gaming video streaming applications,” *IEEE Access*, vol. 7, pp. 74 511–74 527, 2019, doi: 10.1109/ACCESS.2019.2920477.

[61] D. Varga and T. Szirányi, “No-reference video quality assessment via pretrained CNN and LSTM networks,” *Signal, Image and Video Processing*, vol. 13, no. 8, pp. 1569–1576, 2019, doi: 10.1007/s11760-019-01510-8.

[62] Z. Shi and C. Huang, “Network video quality assessment method using fuzzy decision tree,” *IET Communications*, vol. 13, no. 14, pp. 2192–2198, 2019, doi: 10.1049/iet-com.2019.0062.

[63] S. Göring, J. Skowronek, and A. Raake, “DeViQ – a deep no reference video quality model,” *Electronic Imaging*, vol. 2018, no. 14, pp. 1–6, jan 2018, doi: 10.2352/issn.2470-1173.2018.14.hvei-518.

[64] J. Korhonen, “Learning-based prediction of packet loss artifact visibility in networked video,” in *2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*, 2018, pp. 1–6, doi: 10.1109/QoMEX.2018.8463394.

[65] A. Mittal, M. A. Saad, and A. C. Bovik, “A completely blind video integrity oracle,” *IEEE Transactions on Image Processing*, vol. 25, no. 1, pp. 289–300, 2016, doi: 10.1109/TIP.2015.2502725.

[66] N. Barman, S. Schmidt, S. Zadtootaghaj *et al.*, *An Evaluation of Video Quality Assessment Metrics for Passive Gaming Video Streaming*. New York, NY, USA: Association for Computing Machinery, 2018, p. 7–12, doi: 10.1145/3210424.3210434.

[67] S. Zadtootaghaj, N. Barman, S. Schmidt *et al.*, “Nr-gvqm: A no reference gaming video quality metric,” in *2018 IEEE International Symposium on Multimedia (ISM)*, 2018, pp. 131–134, doi: 10.1109/ISM.2018.00031.

[68] S. Göring, R. R. R. Rao, and A. Raake, “nofu — a lightweight no-reference pixel based video quality model for gaming content,” in *2019 Eleventh International*

Conference on Quality of Multimedia Experience (QoMEX), 2019, pp. 1–6, doi: 10.1109/QoMEX.2019.8743262.

[69] J. Søgaard, S. Forchhammer, and J. Korhonen, "No-reference video quality assessment using codec analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 10, pp. 1637–1650, 2015, doi: 10.1109/TCSVT.2015.2397207.

[70] J. Korhonen, "Two-level approach for no-reference consumer video quality assessment," *IEEE Transactions on Image Processing*, vol. 28, no. 12, pp. 5923–5938, 2019, doi: 10.1109/TIP.2019.2923051.

[71] A. Mittal, R. Soundararajan, and A. C. Bovik, "Making a "completely blind" image quality analyzer," *IEEE Signal Processing Letters*, vol. 20, no. 3, pp. 209–212, 2013, doi: 10.1109/LSP.2012.2227726.

[72] C. G. Bampis, Z. Li, and A. C. Bovik, "Continuous prediction of streaming video qoe using dynamic networks," *IEEE Signal Processing Letters*, vol. 24, no. 7, pp. 1083–1087, 2017, doi: 10.1109/LSP.2017.2705423.

[73] C. G. Bampis, Z. Li, A. K. Moorthy *et al.*, "Study of temporal effects on subjective video quality of experience," *IEEE Transactions on Image Processing*, vol. 26, no. 11, pp. 5217–5231, 2017, doi: 10.1109/TIP.2017.2729891.

[74] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Transactions on Image Processing*, vol. 21, no. 12, pp. 4695–4708, 2012, doi: 10.1109/TIP.2012.2214050.

[75] S. Suthaharan, "No-reference visually significant blocking artifact metric for natural scene images," *Signal Processing*, vol. 89, no. 8, pp. 1647–1652, 2009, doi: 10.1016/j.sigpro.2009.02.007.

[76] M. A. Saad, A. C. Bovik, and C. Charrier, "Blind prediction of natural video quality," *IEEE Transactions on Image Processing*, vol. 23, no. 3, pp. 1352–1365, 2014, doi: 10.1109/TIP.2014.2299154.

[77] M. A. Saad and A. C. Bovik, "Blind quality assessment of videos using a model of natural scene statistics and motion coherency," in *2012 Conference Record of the Forty Sixth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, 2012, pp. 332–336, doi: 10.1109/ACSSC.2012.6489018.

[78] A. Mittal, A. K. Moorthy, and A. C. Bovik, “No-reference approaches to image and video quality assessment,” in *Multimedia Quality of Experience (QoE)*. John Wiley & Sons, Ltd, nov 2015, ch. 5, pp. 99–121.

[79] H. Boujut, J. Benois-Pineau, T. A. O. Hadar *et al.*, “No-reference video quality assessment of H.264 video streams based on semantic saliency maps,” in *Proc. SPIE*, 2012, pp. 82 930T–1—-82 930T, doi: 10.1117/12.905379.

[80] F. De Simone, M. Tagliasacchi, M. Naccari *et al.*, “A h.264/avc video database for the evaluation of quality metrics,” in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010, pp. 2430–2433, doi: 10.1109/ICASSP.2010.5496296.

[81] C. e. a. Sheikh HR, Wang Z, “Live image quality assessment database release 2.” *http://live.ece.utexas.edu/research/quality*.

[82] Z. Wang, E. Simoncelli, and A. Bovik, “Multiscale structural similarity for image quality assessment,” in *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers, 2003*, vol. 2, 2003, pp. 1398–1402 Vol.2, doi: 10.1109/ACSSC.2003.1292216.

[83] P. V. Vu, C. T. Vu, and D. M. Chandler, “A spatiotemporal most-apparent-distortion model for video quality assessment,” in *2011 18th IEEE International Conference on Image Processing*, 2011, pp. 2505–2508, doi: 10.1109/ICIP.2011.6116171.

[84] R. Soundararajan and A. C. Bovik, “Video quality assessment by reduced reference spatio-temporal entropic differencing,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 4, pp. 684–694, 2013, doi: 10.1109/TCSVT.2012.2214933.

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

[86] F. Xie, Y. Lu, A. C. Bovik *et al.*, “Application-driven no-reference quality assessment for dermoscopy images with multiple distortions,” *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 6, pp. 1248–1256, 2016, doi: 10.1109/TBME.2015.2493580.

[87] J. Kim, H. Zeng, D. Ghadiyaram *et al.*, “Deep convolutional neural models for picture-quality prediction: Challenges and solutions to data-driven image quality

assessment," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 130–141, 2017, doi: 10.1109/MSP.2017.2736018.

[88] F. Götz-Hahn, V. Hosu, H. Lin *et al.*, "Konvid-150k: A dataset for no-reference video quality assessment of videos in-the-wild," *IEEE Access*, vol. 9, pp. 72 139–72 160, 2021, doi: 10.1109/ACCESS.2021.3077642.

[89] J. Kim, A.-D. Nguyen, and S. Lee, "Deep cnn-based blind image quality predictor," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 1, pp. 11–24, 2019, doi: 10.1109/TNNLS.2018.2829819.

[90] D. Ghadiyaram and A. C. Bovik, "Perceptual quality prediction on authentically distorted images using a bag of features approach," *Journal of Vision*, vol. 17, no. 1, pp. 1–25, 2017, doi: 10.1167/17.1.32.

[91] U. Rajashekar, Z. Wang, and E. P. Simoncelli, "Perceptual quality assessment of color images using adaptive signal representation," in *Human Vision and Electronic Imaging XV*, B. E. Rogowitz and T. N. Pappas, Eds., vol. 7527, International Society for Optics and Photonics. SPIE, 2010, pp. 467–475, doi: 10.1117/12.845312.

[92] D. Ghadiyaram and A. C. Bovik, "Massive online crowdsourced study of subjective and objective picture quality," *IEEE Transactions on Image Processing*, vol. 25, no. 1, pp. 372–387, 2016, doi: 10.1109/TIP.2015.2500021.

[93] ——, "Crowdsourced study of subjective image quality," in *2014 48th Asilomar Conference on Signals, Systems and Computers*, 2014, pp. 84–88, doi: 10.1109/ACSSC.2014.7094402.

[94] J. Chen, D. Huang, H. Huang *et al.*, "Content-aware video quality modeling based on neural network," in *2018 IEEE International Conference on Automation, Electronics and Electrical Engineering (AUTEEE)*, 2018, pp. 173–176, doi: 10.1109/AUTEEE.2018.8720774.

[95] R. Zhang, P. Isola, A. A. Efros *et al.*, "The unreasonable effectiveness of deep features as a perceptual metric," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 586–595, doi: 10.1109/CVPR.2018.00068.

[96] C. Doersch, A. Gupta, and A. A. Efros, “Unsupervised visual representation learning by context prediction,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1422–1430, doi: 10.1109/ICCV.2015.167.

[97] A. Dosovitskiy and T. Brox, “Generating images with perceptual similarity metrics based on deep networks,” in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS’16. Red Hook, NY, USA: Curran Associates Inc., 2016, p. 658–666, doi: 10.5555/3157096.3157170.

[98] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual Losses for Real-Time Style Transfer and Super-Resolution,” in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe *et al.*, Eds. Cham: Springer International Publishing, 2016, pp. 694–711, doi: 10.1007/978-3-319-46475-6_43.

[99] C. Szegedy, V. Vanhoucke, S. Ioffe *et al.*, “Rethinking the inception architecture for computer vision,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826, doi: 10.1109/CVPR.2016.308.

[100] K. Seshadrinathan, R. Soundararajan, A. C. Bovik *et al.*, “Study of subjective and objective quality assessment of video,” *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1427–1441, 2010, doi: 10.1109/TIP.2010.2042111.

[101] P. V. Vu and D. M. Chandler, “ViS3: an algorithm for video quality assessment via analysis of spatial and spatiotemporal slices,” *Journal of Electronic Imaging*, vol. 23, no. 1, pp. 1–25, 2014, doi: 10.1117/1.JEI.23.1.013016.

[102] D. Ghadiyaram, A. C. Bovik, H. Yeganeh *et al.*, “Study of the effects of stalling events on the quality of experience of mobile streaming videos,” in *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2014, pp. 989–993, doi: 10.1109/GlobalSIP.2014.7032269.

[103] T. Virtanen, M. Nuutinen, M. Vaahteranoksa *et al.*, “Cid2013: A database for evaluating no-reference image quality assessment algorithms,” *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 390–402, 2015, doi: 10.1109/TIP.2014.2378061.

[104] M. Nuutinen, T. Virtanen, M. Vaahteranoksa *et al.*, “Cvd2014—a database for evaluating no-reference video quality assessment algorithms,” *IEEE Transactions on Image Processing*, vol. 25, no. 7, pp. 3073–3086, 2016, doi: 10.1109/TIP.2016.2562513.

[105] D. Ghadiyaram, J. Pan, and A. C. Bovik, "A subjective and objective study of stalling events in mobile streaming videos," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 1, pp. 183–197, 2019, doi: 10.1109/TCSVT.2017.2768542.

[106] C. G. Bampis, Z. Li, A. K. Moorthy *et al.*, "Live netflix video quality of experience database," accesed on May 2017. [Online]. Available: http://live.ece.utexas.edu/research/LIVE_NFLXStudy/nflx_index.html

[107] A. Tsifouti, M. M. Nasralla, M. Razaak *et al.*, "A methodology to evaluate the effect of video compression on the performance of analytics systems," in *Proc. SPIE 8546, Optics and Photonics for Counterterrorism, Crime Fighting, and Defence VIII*, vol. 8546, 2012, doi: 10.1117/12.974618.

[108] M. Kristan, A. Leonardis, J. Matas *et al.*, "The visual object tracking vot2017 challenge results," in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2017, pp. 1949–1972, doi: 10.1109/ICCVW.2017.230.

[109] ——, "The sixth visual object tracking vot2018 challenge results," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, September 2018, doi: 10.1007/978-3-030-11009-3_1.

[110] L. Čehovin, A. Leonardis, and M. Kristan, "Visual object tracking performance measures revisited," *IEEE Transactions on Image Processing*, vol. 25, no. 3, pp. 1261–1274, 2016, doi: 10.1109/TIP.2016.2520370.

[111] E. P. Simoncelli and B. A. Olshausen, "Natural Image Statistics and Neural Representation," *Annual Review of Neuroscience*, vol. 24, no. 1, pp. 1193–1216, Mar. 2001, doi: 10.1146%2Fannurev.neuro.24.1.1193.

[112] W. S. Geisler, "Visual Perception and the Statistical Properties of Natural Scenes," *Annual Review of Psychology*, vol. 59, no. 1, pp. 167–192, 2008, doi: 10.1146%2Fannurev.psych.58.110405.085632.

[113] D. L. Ruderman, "The statistics of natural images," *Network: Computation in Neural Systems*, vol. 5, no. 4, pp. 517–548, 1994, doi: 10.1088/0954-898X_5_4_006.

[114] I. F. Nizami, M. Majid, and K. Khurshid, "Efficient feature selection for blind image quality assessment based on natural scene statistics," in *2017 14th International Bhurban*

Conference on Applied Sciences and Technology (IBCAST), 2017, pp. 318–322, doi: 10.1109/IBCAST.2017.7868071.

[115] J. M. Chaquet, E. J. Carmona, and A. Fernández-Caballero, "A survey of video datasets for human action and activity recognition," *Computer Vision and Image Understanding*, vol. 117, no. 6, pp. 633–659, Jun. 2013, doi: 10.1016%2Fj.cviu.2013.01.013.

[116] S. M. Marvasti-Zadeh, L. Cheng, H. Ghanei-Yakhdan *et al.*, "Deep learning for visual tracking: A comprehensive survey," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–26, 2021, doi: 10.1109/TITS.2020.3046478.

[117] J. C. R. Fisher, "Caviar: Context aware vision using image-based active recognition," Ph.D. dissertation, University of Edinburgh, 2005.

[118] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local svm approach," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 3, 2004, pp. 32–36 Vol.3, doi: 10.1109/ICPR.2004.1334462.

[119] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action mach a spatio-temporal maximum average correlation height filter for action recognition," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8, doi: 10.1109/CVPR.2008.4587727.

[120] H. K. Galoogahi, A. Fagg, C. Huang *et al.*, "Need for speed: A benchmark for higher frame rate object tracking," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 1134–1143, doi: 10.1109/ICCV.2017.128.

[121] H. Yu, G. Li, W. Zhang *et al.*, "The Unmanned Aerial Vehicle Benchmark: Object Detection, Tracking and Baseline," *International Journal of Computer Vision*, vol. 128, no. 5, pp. 1141–1159, 2020, doi: 10.1007/s11263-019-01266-1.

[122] D. Du, P. Zhu, L. Wen *et al.*, "Visdrone-sot2019: The vision meets drone single object tracking challenge results," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 199–212, doi: 10.1109/ICCVW.2019.00029.

[123] C. Liu, W. Ding, J. Yang *et al.*, "Aggregation signature for small object tracking," *IEEE Transactions on Image Processing*, vol. 29, pp. 1738–1747, 2020, doi: 10.1109/TIP.2019.2940477.

[124] J. Liu, J. Luo, and M. Shah, “Recognizing realistic actions from videos “in the wild”,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1996–2003, doi: 10.1109/CVPR.2009.5206744.

[125] F. C. Heilbron, V. Escorcia, B. Ghanem *et al.*, “Activitynet: A large-scale video benchmark for human activity understanding,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 961–970, doi: 10.1109/CVPR.2015.7298698.

[126] W. Ruan, C. Liang, Y. Yu *et al.*, “Correlation discrepancy insight network for video re-identification,” *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 16, no. 4, Dec. 2020, doi: 10.1145/3402666.

[127] M. Ye, J. Shen, G. Lin *et al.*, “Deep learning for person re-identification: A survey and outlook,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021, doi: 10.1109/TPAMI.2021.3054775.

[128] K. Zhang, J. Chen, Y. Li *et al.*, “Visual tracking and depth estimation of mobile robots without desired velocity information,” *IEEE Transactions on Cybernetics*, vol. 50, no. 1, pp. 361–373, 2020, doi: 10.1109/TCYB.2018.2869623.

[129] L. Wang, L. Zhang, and Z. Yi, “Trajectory predictor by using recurrent neural networks in visual tracking,” *IEEE Transactions on Cybernetics*, vol. 47, no. 10, pp. 3172–3183, 2017, doi: 10.1109/TCYB.2017.2705345.

[130] G. S. Walia, H. Ahuja, A. Kumar *et al.*, “Unified graph-based multicue feature fusion for robust visual tracking,” *IEEE Transactions on Cybernetics*, vol. 50, no. 6, pp. 2357–2368, 2020, doi: 10.1109/TCYB.2019.2920289.

[131] Y. Yuan, Y. Lu, and Q. Wang, “Tracking as a whole: Multi-target tracking by modeling group behavior with sequential detection,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 12, pp. 3339–3349, 2017, doi: 10.1109/TITS.2017.2686871.

[132] J. M. Irvine, R. J. Wood, D. Reed *et al.*, “Video image quality analysis for enhancing tracker performance,” in *2013 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, 2013, pp. 1–9, doi: 10.1109/AIPR.2013.6749326.

[133] A. Gala and S. Shah, “Joint modeling of algorithm behavior and image quality for algorithm performance prediction,” in *Proceedings of the British Machine Vision Conference*. BMVA Press, 2010, pp. 31.1–31.11, doi: 10.5244/c.24.31.

[134] M. Han, A. Sethi, W. Hua *et al.*, “A detection-based multiple object tracking method,” in *2004 International Conference on Image Processing, 2004. ICIP ’04.*, vol. 5, 2004, pp. 3065–3068 Vol. 5, doi: 10.1109/ICIP.2004.1421760.

[135] R. Kaucic, A. Amitha Perera, G. Brooksby *et al.*, “A unified framework for tracking through occlusions and across sensor gaps,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, vol. 1, 2005, pp. 990–997 vol. 1, doi: 10.1109/CVPR.2005.53.

[136] M. Jenadeleh and M. E. Moghaddam, “BIQWS: efficient Wakeby modeling of natural scene statistics for blind image quality assessment,” *Multimedia Tools and Applications*, vol. 76, no. 12, pp. 13 859–13 880, 2017, doi: 10.1007/s11042-016-3785-4.

[137] D. L. Ruderman and W. Bialek, “Statistics of natural images: Scaling in the woods,” *Phys. Rev. Lett.*, vol. 73, pp. 814–817, Aug. 1994, doi: 10.1103/PhysRevLett.73.814.

[138] K. Sharifi and A. Leon-Garcia, “Estimation of shape parameter for generalized gaussian distributions in subband decompositions of video,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, no. 1, pp. 52–56, 1995, doi: 10.1109/76.350779.

[139] N.-E. Lasmar, Y. Stitou, and Y. Berthoumieu, “Multiscale skewed heavy tailed model for texture analysis,” in *2009 16th IEEE International Conference on Image Processing (ICIP)*, 2009, pp. 2281–2284, doi: 10.1109/ICIP.2009.5414404.

[140] R. Krupiński and J. Purczyński, “Approximated fast estimator for the shape parameter of generalized Gaussian distribution,” *Signal Processing*, vol. 86, no. 2, pp. 205–211, feb 2006, doi: 10.1016/j.sigpro.2005.05.003.

[141] L. Tang, L. Li, K. Gu *et al.*, “Blind quality index for camera images with natural scene statistics and patch-based sharpness assessment,” *Journal of Visual Communication and Image Representation*, vol. 40, pp. 335–344, Oct. 2016, doi: 10.1016/j.jvcir.2016.07.007.

[142] P. Gupta, J. L. Glover, N. G. Paulter *et al.*, “Studying the Statistics of Natural X-ray Pictures,” *Journal of Testing and Evaluation*, vol. 46, no. 4, pp. 1478–1488, May 2018, doi: 10.1520/JTE20170345.

[143] S. Bosse, D. Maniry, K.-R. Müller *et al.*, “Deep neural networks for no-reference and full-reference image quality assessment,” *IEEE Transactions on Image Processing*, vol. 27, no. 1, pp. 206–219, 2018, doi: 10.1109/TIP.2017.2760518.

[144] D. Ghadiyaram and A. C. Bovik, “Feature maps driven no-reference image quality prediction of authentically distorted images,” in *Human Vision and Electronic Imaging XX*, vol. 9394. International Society for Optics and Photonics, 2015, p. 93940J, doi: 10.1117/12.2084807.

[145] D. Ghadiyaram and A. Bovik, “Automatic quality prediction of authentically distorted pictures,” in *Proc. SPIE*, 2015.

[146] B. A. Olshausen and D. J. Field, “Emergence of simple-cell receptive field properties by learning a sparse code for natural images,” *Nature*, vol. 381, 1996, doi: 10.1038/381607a0.

[147] M. Kim, S. Kumar, V. Pavlovic *et al.*, “Face tracking and recognition with visual constraints in real-world videos,” in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8, doi: 10.1109/CVPR.2008.4587572.

[148] P. Korshunov and W. T. Ooi, “Video quality for face detection, recognition, and tracking,” *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 7, no. 3, Sep. 2011, doi: 10.1145/2000486.2000488.

[149] R. G. Nieto, H. D. B. Restrepo, and I. Cabezas, “How video object tracking is affected by in-capture distortions?” in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 2227–2231, doi: 10.1109/ICASSP.2019.8683625.

[150] Z. Xu, R. Hu, J. Chen *et al.*, “How much bandwidth does surveillance system require?” in *2015 IEEE International Conference on Image Processing (ICIP)*, 2015, pp. 1762–1766, doi: 10.1109/ICIP.2015.7351103.

[151] R. G. Nieto, H. D. B. Restrepo, R. F. Quintero *et al.*, “No reference video quality assessment with authentic distortions using 3-d deep convolutional neural network,” *Electronic Imaging*, vol. 2020, no. 9, pp. 168–1–168–7, 2020, doi: 10.2352/ISSN.2470-1173.2020.9.IQSP-168.

[152] R. G. Nieto, H. D. B. Restrepo, and J. F. Ruiz-Munoz, “Quality aware feature selection for video object tracking,” *Electronic Imaging*, vol. 2020, no. 9, pp. 169–1–169–7, 2020, doi: 10.2352/ISSN.2470-1173.2020.9.IQSP-169.

[153] M. Kristan, J. Matas, A. Leonardis *et al.*, “The Visual Object Tracking VOT2015 Challenge Results,” in *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, vol. 2015-Febru. IEEE, dec 2015, pp. 564–586, doi: 10.1109/ICCVW.2015.79.

[154] D. Ghadiyaram, J. Pan, A. C. Bovik *et al.*, “Subjective and objective quality assessment of mobile videos with in-capture distortions,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 1393–1397, doi: 10.1109/ICASSP.2017.7952385.

[155] Darkpgmr, “Darklabel - video/image labeling and annotation tool,” Available: https://github.com/darkpgmr/DarkLabel, 2020.

[156] J. Ning, J. Yang, S. Jiang *et al.*, “Object tracking via dual linear structured svm and explicit feature map,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4266–4274, doi: 10.1109/CVPR.2016.462.

[157] Z. He, Y. Fan, J. Zhuang *et al.*, “Correlation filters with weighted convolution responses,” in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2017, pp. 1992–2000, doi: 10.1109/ICCVW.2017.233.

[158] T. Kokul, C. Fookes, S. Sridharan *et al.*, “Gate connected convolutional neural network for object tracking,” in *2017 IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 2602–2606, doi: 10.1109/ICIP.2017.8296753.

[159] N. Wang, W. Zhou, Q. Tian *et al.*, “Multi-cue correlation filters for robust visual tracking,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4844–4853, doi: 10.1109/CVPR.2018.00509.

[160] S. Bai, Z. He, Y. Dong *et al.*, “Multi-hierarchical independent correlation filters for visual tracking,” in *2020 IEEE International Conference on Multimedia and Expo (ICME)*, 2020, pp. 1–6, doi: 10.1109/ICME46284.2020.9102759.

[161] F. Li, C. Tian, W. Zuo *et al.*, “Learning spatial-temporal regularized correlation filters for visual tracking,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4904–4913, doi: 10.1109/CVPR.2018.00515.

[162] M. Che, R. Wang, Y. Lu *et al.*, “Channel Pruning for Visual Tracking,” in *Computer Vision – ECCV 2018 Workshops*, L. Leal-Taixé and S. Roth, Eds. Cham: Springer International Publishing, 2019, pp. 70–82, doi: 10.1007/978-3-030-11009-3_3.

[163] H. Lee and D. Kim, "Salient region-based online object tracking," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018, pp. 1170–1177, doi: 10.1109/WACV.2018.00133.

[164] M. Danelljan, G. Bhat, F. S. Khan *et al.*, "Eco: Efficient convolution operators for tracking," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6931–6939, doi: 10.1109/CVPR.2017.733.

[165] D. S. Bolme, J. R. Beveridge, B. A. Draper *et al.*, "Visual object tracking using adaptive correlation filters," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2544–2550, doi: 10.1109/CVPR.2010.5539960.

[166] S. Du and S. Wang, "An overview of correlation-filter-based object tracking," *IEEE Transactions on Computational Social Systems*, pp. 1–14, 2021, doi: 10.1109/TCSS.2021.3093298.

[167] C. Ma, J.-B. Huang, X. Yang *et al.*, "Hierarchical convolutional features for visual tracking," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 3074–3082, doi: 10.1109/ICCV.2015.352.

[168] M. Danelljan, G. Hager, F. S. Khan *et al.*, "Learning spatially regularized correlation filters for visual tracking," in *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE, dec 2015, doi: 10.1109/iccv.2015.490.

[169] M. Danelljan, G. Häger, F. S. Khan *et al.*, "Convolutional features for correlation filter based visual tracking," in *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, 2015, pp. 621–629, doi: 10.1109/ICCVW.2015.84.

[170] M. Danelljan, A. Robinson, F. Shahbaz Khan *et al.*, "Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe *et al.*, Eds. Cham: Springer International Publishing, 2016, pp. 472–488, doi: 10.1007/978-3-319-46454-1_29.

[171] M. Danelljan, G. Häger, F. S. Khan *et al.*, "Accurate scale estimation for robust visual tracking," in *Proceedings of the British Machine Vision Conference 2014*. British Machine Vision Association, 2014, doi: 10.5244/c.28.65.

[172] Y. Qi, S. Zhang, L. Qin *et al.*, “Hedged deep tracking,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4303–4311, doi: 10.1109/CVPR.2016.466.

[173] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan *et al.*, “Beyond short snippets: Deep networks for video classification,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 4694–4702, doi: 10.1109/CVPR.2015.7299101.

[174] C. Lin, “Face detection in complicated backgrounds and different illumination conditions by using YCbCr color space and neural network,” *Pattern Recognition Letters*, vol. 28, no. 16, pp. 2190–2200, 2007, doi: 10.1016/j.patrec.2007.07.003.

[175] S. Lee, Y. Kwak, Y. J. Kim *et al.*, “Contrast-preserved chroma enhancement technique using ycbcr color space,” *IEEE Transactions on Consumer Electronics*, vol. 58, no. 2, pp. 641–645, 2012, doi: 10.1109/TCE.2012.6227471.

[176] C. G. Bampis, Z. Li, and A. C. Bovik, “Enhancing temporal quality measurements in a globally deployed streaming video quality predictor,” in *2018 25th IEEE International Conference on Image Processing (ICIP)*, 2018, pp. 614–618, doi: 10.1109/ICIP.2018.8451275.

[177] H. Men, H. Lin, and D. Saupe, “Spatiotemporal feature combination model for no-reference video quality assessment,” in *2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, may 2018, doi: 10.1109/qomex.2018.8463426.

[178] I. Abouelaziz, M. El Hassouni, and H. Cherifi, “Blind 3D mesh visual quality assessment using support vector regression,” *Multimedia Tools and Applications*, vol. 77, no. 18, pp. 24 365–24 386, 2018, doi: 10.1007/s11042-018-5706-1.

[179] B. Appina, S. V. R. Dendi, K. Manasa *et al.*, “Study of subjective quality and objective blind quality prediction of stereoscopic videos,” *IEEE Transactions on Image Processing*, vol. 28, no. 10, pp. 5027–5040, 2019, doi: 10.1109/TIP.2019.2914950.

[180] S. V. R. Dendi, G. Krishnappa, and S. S. Channappayya, “Full-reference video quality assessment using deep 3d convolutional neural networks,” in *2019 National Conference on Communications (NCC)*, 2019, pp. 1–5, doi: 10.1109/NCC.2019.8732265.

[181] A. Aldahdooh, E. Masala, O. Janssens *et al.*, “Improved performance measures for video quality assessment algorithms using training and validation sets,” *IEEE Transactions on Multimedia*, vol. 21, no. 8, pp. 2026–2041, 2019, doi: 10.1109/TMM.2018.2882091.

[182] M. Jenadeleh, “Blind image and video quality assessment,” phdthesis, Universitat Konstanz, Aug. 2018. [Online]. Available: http://nbn-resolving.de/urn:nbn:de:bsz:352-2-g5si24h73rb40

[183] D. Li, T. Jiang, and M. Jiang, “Unified Quality Assessment of in-the-Wild Videos with Mixed Datasets Training,” *International Journal of Computer Vision*, vol. 129, no. 4, pp. 1238–1257, 2021, doi: 10.1007/s11263-020-01408-w.

[184] ——, “Quality assessment of in-the-wild videos,” in *Proceedings of the 27th ACM International Conference on Multimedia*, ser. MM ’19. New York, NY, USA: Association for Computing Machinery, 2019, p. 2351–2359, doi: 10.1145/3343031.3351028.

[185] Z. Sinno and A. C. Bovik, “Large-scale study of perceptual video quality,” *IEEE Transactions on Image Processing*, vol. 28, no. 2, pp. 612–627, 2019, doi: 10.1109/TIP.2018.2869673.

[186] V. Hosu, F. Hahn, M. Jenadeleh *et al.*, “The konstanz natural video database (konvid-1k),” in *2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)*, 2017, pp. 1–6, doi: 10.1109/QoMEX.2017.7965673.

[187] M. Narwaria, W. Lin, and A. Liu, “Low-complexity video quality assessment using temporal quality variations,” *IEEE Transactions on Multimedia*, vol. 14, no. 3, pp. 525–535, 2012, doi: 10.1109/TMM.2012.2190589.

[188] J. Park, K. Seshadrinathan, S. Lee *et al.*, “Video quality pooling adaptive to perceptual distortion severity,” *IEEE Transactions on Image Processing*, vol. 22, no. 2, pp. 610–620, 2013, doi: 10.1109/TIP.2012.2219551.

[189] A. Likas, N. Vlassis, and J. J. Verbeek, “The global k-means clustering algorithm,” *Pattern Recognition*, vol. 36, no. 2, pp. 451–461, 2003, doi: 10.1016/S0031-3203(02)00060-2.

[190] N. D. Narvekar and L. J. Karam, “A no-reference image blur metric based on the cumulative probability of blur detection (cpbd),” *IEEE Transactions on Image Processing*, vol. 20, no. 9, pp. 2678–2683, 2011, doi: 10.1109/TIP.2011.2131660.

[191] J. A. L. Mazuera and S. A. T. Moron, “Evaluation of no-reference quality prediction metrics in videos impaired by authentic distortions,” Jan. 2021, bachellor Degree Thesis.

[192] F. De Simone, M. Naccari, M. Tagliasacchi *et al.*, “Subjective assessment of h.264/avc video sequences transmitted over a noisy channel,” in *2009 International Workshop on Quality of Multimedia Experience*, 2009, pp. 204–209, doi: 10.1109/QOMEX.2009.5246952.

[193] R. Soundararajan and S. Biswas, “Machine vision quality assessment for robust face detection,” *Signal Processing: Image Communication*, vol. 72, no. July 2018, pp. 92–104, 2019, doi: 10.1016/j.image.2018.12.012.

[194] C. G. Rodríguez-Pulecio, H. D. Benítez-Restrepo, and A. C. Bovik, “Making long-wave infrared face recognition robust against image quality degradations,” *Quantitative InfraRed Thermography Journal*, 2019, doi: 10.1080/17686733.2019.1579020.

[195] M. Dai, S. Cheng, X. He *et al.*, “Object tracking in the presence of shaking motions,” *Neural Computing and Applications*, vol. 31, no. 10, pp. 5917–5934, 2019, doi: 10.1007/s00521-018-3387-3.

[196] M. B. Blaschko and C. H. Lampert, “Learning to Localize Objects with Structured Output Regression,” in *Computer Vision – ECCV 2008*, D. Forsyth, P. Torr, and A. Zisserman, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 2–15, doi: 10.1007/978-3-540-88682-2_2.

[197] A. Vedaldi, V. Gulshan, M. Varma *et al.*, “Multiple kernels for object detection,” in *2009 IEEE 12th International Conference on Computer Vision*, 2009, pp. 606–613, doi: 10.1109/ICCV.2009.5459183.

[198] P. F. Felzenszwalb, R. B. Girshick, D. McAllester *et al.*, “Object detection with discriminatively trained part-based models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010, doi: 10.1109/TPAMI.2009.167.

[199] S. Avidan, “Support vector tracking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 1064–1072, 2004, doi: 10.1109/TPAMI.2004.53.

[200] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, vol. 1, 2005, pp. 886–893 vol. 1, doi: 10.1109/CVPR.2005.177.

[201] S. Javed, A. Mahmood, J. Dias *et al.*, "Hierarchical spatiotemporal graph regularized discriminative correlation filter for visual object tracking," *IEEE Transactions on Cybernetics*, pp. 1–16, 2021, doi: 10.1109/TCYB.2021.3086194.

[202] L. Bertinetto, J. Valmadre, S. Golodetz *et al.*, "Staple: Complementary learners for real-time tracking," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1401–1409, doi: 10.1109/CVPR.2016.156.

[203] H. K. Galoogahi, A. Fagg, and S. Lucey, "Learning background-aware correlation filters for visual tracking," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 1144–1152, doi: 10.1109/ICCV.2017.129.

[204] K. Dai, D. Wang, H. Lu *et al.*, "Visual tracking via adaptive spatially-regularized correlation filters," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4665–4674, doi: 10.1109/CVPR.2019.00480.

[205] J. F. Henriques, R. Caseiro, P. Martins *et al.*, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015, doi: 10.1109/TPAMI.2014.2345390.

[206] H. Liu, Q. Hu, B. Li *et al.*, "Robust long-term tracking via instance-specific proposals," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 4, pp. 950–962, 2020, doi: 10.1109/TIM.2019.2908715.

[207] W. Ruan, M. Ye, Y. Wu *et al.*, "Ticnet: A target-insight correlation network for object tracking," *IEEE Transactions on Cybernetics*, pp. 1–13, 2021, doi: 10.1109/TCYB.2021.3070677.

[208] M. Jain, A. Tyagi, A. V. Subramanyam *et al.*, "Channel graph regularized correlation filters for visual object tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2021, doi: 10.1109/TCSVT.2021.3063144.

[209] X.-F. Zhu, X.-J. Wu, T. Xu *et al.*, "Robust visual object tracking via adaptive attribute-aware discriminative correlation filters," *IEEE Transactions on Multimedia*, pp. 1–1, 2021, doi: 10.1109/TMM.2021.3050073.

[210] X. Lan, Z. Yang, W. Zhang *et al.*, "Spatial-temporal regularized multi-modality correlation filters for tracking with re-detection," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 17, no. 2, May 2021, doi: 10.1145/3430257.

[211] W. Zhou, S. Gao, L. Zhang *et al.*, "Histogram of oriented gradients feature extraction from raw bayer pattern images," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 5, pp. 946–950, 2020, doi: 10.1109/TCSII.2020.2980557.

[212] A. C. Bovik, *The Essential Guide to Image Processing*. Elsevier Science, 2009.

[213] A. Dutta, A. Mondal, N. Dey *et al.*, "Vision Tracking: A Survey of the State-of-the-Art," *SN Computer Science*, vol. 1, no. 1, p. 57, 2020, doi: 10.1007/s42979-019-0059-z.

[214] J. Jiarpakdee, C. Tantithamthavorn, H. K. Dam *et al.*, "An empirical study of model-agnostic techniques for defect prediction models," *IEEE Transactions on Software Engineering*, pp. 1–1, 2020, doi: 10.1109/TSE.2020.2982385.

[215] P.-H. Chen, R.-E. Fan, and C.-J. Lin, "A study on smo-type decomposition methods for support vector machines," *Trans. Neur. Netw.*, vol. 17, no. 4, p. 893–908, Jul. 2006, doi: 10.1109/TNN.2006.875973.

[216] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008. [Online]. Available: http://jmlr.org/papers/v9/vandermaaten08a.html

[217] 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 *et al.*, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 34–45, doi: 10.1007/978-3-642-13789-1_4.

[218] Y. Fang, Y. Yuan, L. Li *et al.*, "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.

[219] B. Yan, X. Zhang, D. Wang *et al.*, "Alpha-refine: Boosting tracking performance by precise bounding box estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 5289–5298.

[220] B. Li, W. Wu, Q. Wang *et al.*, "Siamrpn++: Evolution of siamese visual tracking with very deep networks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4277–4286, doi: 10.1109/CVPR.2019.00441.

[221] M. H. Abdelpakey and M. S. Shehata, “Dp-siam: Dynamic policy siamese network for robust object tracking,” *IEEE Transactions on Image Processing*, vol. 29, pp. 1479–1492, 2020, doi: 10.1109/TIP.2019.2942506.

[222] C. Huang, S. Lucey, and D. Ramanan, “Learning policies for adaptive tracking with deep feature cascades,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 105–114, doi: 10.1109/ICCV.2017.21.

[223] N. Wang, W. Zhou, Y. Song *et al.*, “Real-time correlation tracking via joint model compression and transfer,” *IEEE Transactions on Image Processing*, vol. 29, pp. 6123–6135, 2020, doi: 10.1109/TIP.2020.2989544.

[224] Y. Wu, J. Lim, and M.-H. Yang, “Online object tracking: A benchmark,” in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2411–2418, doi: 10.1109/CVPR.2013.312.

[225] T. Zhang, C. Xu, and M.-H. Yang, “Learning multi-task correlation particle filters for visual tracking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 365–378, 2019, doi: 10.1109/TPAMI.2018.2797062.

[226] K. Nai, Z. Li, Y. Gan *et al.*, “Robust visual tracking via multitask sparse correlation filters learning,” *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2021, doi: 10.1109/TNNLS.2021.3097498.

[227] Y. Zheng, X. Liu, B. Xiao *et al.*, “Multi-task convolution operators with object detection for visual tracking,” *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2021, doi: 10.1109/TCSVT.2021.3071128.

[228] Z. Luo, P.-M. Jodoin, S.-Z. Su *et al.*, “Traffic analytics with low-frame-rate videos,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 4, pp. 878–891, 2018, doi: 10.1109/TCSVT.2016.2632439.

[229] Y. Li, H. Ai, T. Yamashita *et al.*, “Tracking in low frame rate video: A cascade particle filter with discriminative observers of different life spans,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 1728–1740, 2008, doi: 10.1109/TPAMI.2008.73.

[230] D. Siqueira and A. M. C. Machado, “People detection and tracking in low frame-rate dynamic scenes,” *IEEE Latin America Transactions*, vol. 14, no. 4, pp. 1966–1971, 2016, doi: 10.1109/TLA.2016.7483541.

[231] A. Tsifouti, “Image usefulness of compressed surveillance footage with different scene contents,” Ph.D. dissertation, University of Westminster, 2016. [Online]. Available: https://westminsterresearch.westminster.ac.uk/download/3bbc35ba061a3755610c3826258b466f00d7d3df6955dac3412fbb25898f4355/19461609/Tsifouti_Anastasia_thesis.pdf

[232] R. Dash and B. Majhi, “Motion blur parameters estimation for image restoration,” *Optik*, vol. 125, no. 5, pp. 1634–1640, 2014, doi: 10.1016/j.ijleo.2013.09.026.

[233] I. Iraei and K. Faez, “A motion parameters estimating method based on deep learning for visual blurred object tracking,” *IET Image Processing*, vol. 15, no. 10, pp. 2213–2226, 2021, doi: 10.1049/ipr2.12189.

[234] Y.-Q. Liu, X. Du, H.-L. Shen *et al.*, “Estimating generalized gaussian blur kernels for out-of-focus image deblurring,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 3, pp. 829–843, 2021, doi: 10.1109/TCSVT.2020.2990623.

[235] Q. Guo, W. Feng, R. Gao *et al.*, “Exploring the effects of blur and deblurring to visual object tracking,” *IEEE Transactions on Image Processing*, vol. 30, pp. 1812–1824, 2021, doi: 10.1109/TIP.2020.3045630.

[236] Y. Qi, S. Zhang, F. Jiang *et al.*, “Siamese local and global networks for robust face tracking,” *IEEE Transactions on Image Processing*, vol. 29, pp. 9152–9164, 2020, doi: 10.1109/TIP.2020.3023621.

[237] G. Park, A. Argyros, J. Lee *et al.*, “3d hand tracking in the presence of excessive motion blur,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 5, pp. 1891–1901, 2020, doi: 10.1109/TVCG.2020.2973057.

[238] R. Fergus, B. Singh, A. Hertzmann *et al.*, “Removing camera shake from a single photograph,” *ACM Trans. Graph.*, vol. 25, no. 3, p. 787–794, Jul. 2006, doi: 10.1145/1141911.1141956.

[239] L. Huang and Y. Xia, “Joint blur kernel estimation and CNN for blind image restoration,” *Neurocomputing*, vol. 396, pp. 324–345, 2020, doi: 10.1016/j.neucom.2018.12.083.

[240] W. Wang and M. K. Ng, "Convex regularized inverse filtering methods for blind image deconvolution," *Signal, Image and Video Processing*, vol. 10, no. 7, pp. 1353–1360, 2016, doi: 10.1007/s11760-016-0924-3. [Online]. Available: https://doi.org/10.1007/s11760-016-0924-3

[241] X. Tao, H. Gao, X. Shen *et al.*, "Scale-recurrent network for deep image deblurring," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8174–8182, doi: 10.1109/CVPR.2018.00853.

[242] L. Huang, Y. Xia, and T. Ye, "Effective blind image deblurring using matrix-variable optimization," *IEEE Transactions on Image Processing*, vol. 30, pp. 4653–4666, 2021, doi: 10.1109/TIP.2021.3073856.

[243] J. Pan, D. Sun, H. Pfister *et al.*, "Deblurring images via dark channel prior," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 10, pp. 2315–2328, 2018, doi: 10.1109/TPAMI.2017.2753804.

[244] L. Xu, H. Luo, B. Hui *et al.*, "Real-Time Robust Tracking for Motion Blur and Fast Motion via Correlation Filters," *Sensors*, vol. 16, no. 9, 2016, doi: 10.3390/s16091443.

[245] B. Yan, B. Bare, and W. Tan, "Naturalness-aware deep no-reference image quality assessment," *IEEE Transactions on Multimedia*, vol. 21, no. 10, pp. 2603–2615, 2019, doi: 10.1109/TMM.2019.2904879.

[246] E. Demirbilek and J.-C. Grégoire, "Inrs audiovisual quality dataset," in *Proceedings of the 24th ACM International Conference on Multimedia*, ser. MM '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 167–171, doi: 10.1145/2964284.2967204.

[247] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[248] A. Paszke, S. Gross, and F. Massa, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, and A. B. et al., Eds., vol. 32. Curran Associates, Inc., 2019.

[249] I. Jung, J. Son, M. Baek *et al.*, "Real-Time MDNet," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu *et al.*, Eds. Cham: Springer International Publishing, 2018, pp. 89–104, doi: 10.1007/978-3-030-01225-0_6.

[250] M. Danelljan, G. Bhat, F. S. Khan *et al.*, "Atom: Accurate tracking by overlap maximization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019, doi: 10.1109/CVPR.2019.00479.

[251] G. Bhat, M. Danelljan, L. Van Gool *et al.*, "Learning discriminative model prediction for tracking," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 6181–6190, doi: 10.1109/ICCV.2019.00628.

[252] O. Russakovsky, J. Deng, H. Su *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015, doi: 10.1007/s11263-015-0816-y.

[253] E. Real, J. Shlens, S. Mazzocchi *et al.*, "Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7464–7473, doi: 10.1109/CVPR.2017.789.

[254] T.-Y. Lin, M. Maire, S. Belongie *et al.*, "Microsoft COCO: Common Objects in Context," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele *et al.*, Eds. Cham: Springer International Publishing, 2014, pp. 740–755, doi: 10.1007/978-3-319-10602-1_48.

[255] A. G. Howard, M. Zhu, B. Chen *et al.*, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," 2017.

[256] K. He, X. Zhang, S. Ren *et al.*, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[257] S. Boyd, N. Parikh, E. Chu *et al.*, *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*. ACM DL, 2011, doi: 10.1561/2200000016.

[258] M. Danelljan, G. Häger, F. S. Khan *et al.*, "Discriminative scale space tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 8, pp. 1561–1575, 2017, doi: 10.1109/TPAMI.2016.2609928.

[259] Y. Song, C. Ma, X. Wu *et al.*, "Vital: Visual tracking via adversarial learning," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8990–8999, doi: 10.1109/CVPR.2018.00937.

Appendices

A Abbreviations

- **ADMM**: Alternating Direction Method of Multipliers.
- **AGGD**: Asymmetric Generalized Gaussian Distribution.
- **AUC**: Area Under the Curve.
- **AVC**: Advanced Video Coding.
- **AWGN**: Additive White Gaussian Noise.
- **BRISQUE**: Blind/Referenceless Image Spatial Quality Evaluator.
- **CF**: Correlational Filter.
- **CNN**: Convolutional Neural Network.
- **DCF**: Discriminative Correlation Filter.
- **DCT**: Discrete Cosine Transform.
- **DSVD**: Distorted Surveillance Video Dataset.
- **FFT**: Fast Fourier Transform.
- **FPS**: Frames Per Second rate.
- **FR**: Full-Reference.
- **GGD**: Generalized Gaussian Distribution.
- **HC**: High Complexity.
- **HOG**: Histogram of Oriented Gradients.
- **HVS**: Human Visual System.
- **IQA**: Image Quality Assessment.
- **LC**: Low Complexity.

- **LTT**: Long Term Tracking.
- **LWIR**: Long-Wave InfraRed.
- **MOS**: Mean Opinion Scores.
- **MOVIE**: MOtion-based Video Integrity Evaluation.
- **MSCN**: Mean Subtraction and Contrast Normalization.
- **MSE**: Mean Squared Error.
- **MS-SSIM**:MultiScale-Structural SIMilarity index.
- **NR**: No-Reference.
- **NSS**: Natural Scene Statistics.
- **NVS**: Natural Video Statistics.
- **PCA**: Principal Component Analysis.
- **PCC**: Pearson Correlation Coefficient.
- **PDF**: Probability Density Distribution Function.
- **PLCC**: Pearson Linear Correlation Coefficient.
- **PO**: Passing Out.
- **PSNR**: Peak-Signal-to-Noise Ratio.
- **RK**: Robbing with a Knife.
- **ROI**: Region of Interest.
- **RR**: Reduced Reference.
- **SAP**: Salt And Pepper.
- **SROCC**: Spearman's Rank Correlation Coefficient.
- **STT**: Short Term Tracking.
- **SVD**: Singular Value Decomposition.

- **SVM**: Support Vector Machine.
- **SVR**: Support Vector Regressor.
- **TLVQM**: Two-Level Video Quality Model.
- **TSNE**: t-distributed Stochastic Neighbor Embedding.
- **VOT**: Video Object Tracking.
- **VQA**: Video Quality Assessment.
- **VQM**: Video Quality Metric.

B State-of-the-art Trackers implemented for benchmarking and spatial scale analysis

1. **CFWCR:** *Correlation Filters with Weighted Convolution Responses*. Tracker in second position of VOT-2017. "CFWCR adopts Efficient Convolution Operators tracker as the baseline approach. A continuous convolution operator based tracker is derived which fully exploits the discriminative power in the CNN feature representations. First, each individual feature extracted from different layers of the deep pre-trained CNN is normalised, and after that, the weighted convolution responses from each feature block are summed to produce the final confidence score. It is also found that the 10-layers design is optimal for continuous scale estimation. The empirical evaluations demonstrate clear improvements by the proposed tracker based on the Efficient Convolution Operators Tracker (ECO)." [108, 157]

2. **Gnet:** *gNetTracker*. Tracker in fifth position of VOT-2017. "The tracker Gnet integrates *GoogLeNet* features with the spatially regularized model (SRDCF) and ECO model. In both cases, it was observed that tracking accuracy increased. The spatially regularized model on different combination of layers is evaluated. The results of these evaluations on VOT 2016 dataset indicated that features extracted from inception are most suitable for the purpose of object tracking. This finding is in direct contrast to the finding of previous studies done on VGGNet which recommended the use of shallower layers for tracking based on the argument that shallower layers have more resolution and hence can be used for object localization. It was found that a combination of shallow layers (like inception module) with deeper layers result in slight improvement in the performance of tracker but also leads to significant increase in computational cost." [108]

3. **CPT:** *Channel Pruning for Visual Tracking*, tracker in eighth position of VOT-2018. "In order to improve the tracking speed, the tracker CPT is proposed. The tracker introduces an effective channel pruning based VGG network to fast extract the deep convolutional features. In this way, it can obtain deeper convolutional features for better representations of various objects' variations without worrying about the speed of suppression. To further reduce the redundancy features, the Average Feature Energy Ratio is proposed to extract effective convolutional channel of the selected deep convolution layer and increase the tracking speed. The method also ameliorates the optimization process in minimizing the

location error as adaptive iterative optimization strategy." [109, 162]

4. **DeepSTRCF [161]***Spatial-Temporal Regularized Correlation Filters with Deep CNN Features*. Tracker in the seventh position of VOT-2018. DeepSTRCF implements a variant of the STRCF tracker with deep CNN features. STRCF addresses the computational inefficiency problem of the SRDCF tracker from two aspects: (i) a temporal regularization term to remove the need for formulation on large training sets, and (ii) an ADMM algorithm to solve the STRCF model efficiently. Therefore, it can provide more robust models and much faster solutions than SRDCF thanks to online Passive-Aggressive learning and ADMM solver, respectively [109]. We implemented this tracker on MatLab running on a GPU.

5. **SRCT:** *Salient Region weighted Correlation Filter Tracker*. Tracker in twelfth position of VOT-2018. "SRCT is the ensemble tracker composed of Salient Region-based Tracker and ECO tracker. The score map of Salient Region based Tracker is weighted to the score map of ECO tracker in spatial domain." [109, 163]

6. **CPT fast:** *Channel Pruning for Visual Tracking*. Tracker in fourteenth position of VOT-2018. "The fast CPT (called CPT fast) method is based on CPT tracker and the DSST method which is applied to estimate the tracking object's scale." [109, 162]

7. **MCCT [159]:** *Multi-Cue Correlation Tracker*. Tracker in the sixth position of VOT-2017 and twentieth position of VOT-2018. MCCT combines different types of features. It constructs multiple experts through Discriminative Correlation Filter -DCF- tracking the target independently in each frame. The divergence of multiple experts reveals the reliability of the current tracking, which is quantified for adaptively updating the experts and keep them from corruption. For estimating the target scale, MCCT follows the DCCT tracker. The expert with the highest robustness score is selected after evaluating the overall reliability of each node [108].

8. **Scale-DLSSVM**: *Scale Dual Linear Structured Support Vector Machine, Multi-Scale estimation and linear kernels*. "Efficient dual linear SSVM (DLSSVM) algorithm to enable fast learning and execution during tracking. By analyzing the dual variables, we propose a primal classifier update formula where the learning step size is computed in closed form. This online learning method significantly improves the robustness of the proposed linear SSVM with lower computational cost. Second, we approximate the intersection

kernel for feature representations with an explicit feature map to further improve tracking performance. Finally, we extend the proposed DLSSVM tracker with multi-scale estimation to address the "drift" problem." [156]

9. **TFCR [13]**: *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 some effects of the negative samples that act on the object appearance model. TFCR use a target-focused regression model to train the convolutional neural network (VGGNet [247]), which pay more attention to the target sample and reduces the influence of the background samples on the target appearance model. TFCR extracts some search patches at different scales with the exact central location and feeds them into the feature extractor to resolve the scale-related challenges. After that, select the optimal scale factor by searching for the maximum value in the prediction maps.

10. **Alpha-Refine [219]**: Alpha-Refine was the winner of the VOT2020 Real-Time Challenge with an EAO of 0.499. It is a module implemented in Pytorch [248], which refine the base tracker outputs and improve the tracking performance. This module consists of a pixel-wise correlation, a corner prediction head, and an auxiliary mask head (can be deactivated at inference stage to improve speed), introducing pixel-level supervision into the training as the core components. Alpha-Refine modules were trained for 40 epochs and 500 iterations, each one on eight Nvidia 2080Ti GPU. This module introduces additional computation loads of around 5-6 ms per frame. Alpha-Refine module was tested within six trackers [164,220,249–251], trained on some segmentation datasets, and tested on multiple tracking benchmarks [2, 7, 9, 10], increasing up to 7.4% the AUC of the original baseline tracker. In our experiments, we used SiamRPN++ [220] as the base tracker for the Alpha-Refine module.

11. **SiamRPN++ [220]**: 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 [252], trained with other sets [253, 254], and tested in tracking datasets [3, 4, 109]. SiamRPN++ replace cross-correlation with depthwise correlation, reducing the computational cost and memory usage. SiamRPN++ has a FPS of 35, possible to increase to 70 FPS using MobileNet [255] backbone. SiamRPN++ has a 0.414 EAO score on VOT2018, which is 4.0% higher than the single-layer baseline.

12. **MFT [160]**: MFT was the winner of the VOT2018 challenge [109]. MFT, implemented on MATLAB, consists of hierarchical features selection, independent group CF online learning, adaptive multi-branch CF fusion and motion estimation module (alleviates the problem of fast motion). This tracker uses multi-hierarchical deep features (ResNet [256] before ReLU, reduced by PCA-256) with different semantic information to track multi-scale objects. The motion estimation module (improve the robustness to motion blur), based on Kalman filters, previously generates one Gaussian motion map with motion information. Then hierarchical features from different layers are extracted by ResNet and are multiplied by the Gaussian motion map. These deep features are independently fed into different CFs to online update the parameters, using weights to give attention to different channels. Finally, an adaptive weight scheme is utilized to generate the final score map to locate the target. This tracker benefits from online learning to adapt appearance changes and scale variances to the detriment of being computationally demanding.

13. **LADCF [14]**: LADCF (MATLAB implemented) constructs the 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 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) with one augmented Lagrangian method, used to optimize the variables iteratively (using ADMM [257]).

14. **C-COT [170]**: This tracker learns a discriminative continuous convolution operator as its tracking model. It poses a learning problem in the continuous spatial domain. This method enables a natural and efficient fusion of multi-resolution feature maps, e.g., using several convolutional layers from a pre-trained CNN. The continuous formulation also enables highly accurate localization by sub-pixel refinement [109].

15. **ECO [164]**: Efficient Convolution Operators for Tracking improves both speed and performance by introducing several efficient strategies. ECO addresses the problems of computational complexity and over-fitting in state-of-the-art DCF trackers by introducing: (i) a factorized convolution operator, which drastically reduces the number of parameters

in the model; (ii) a compact generative model of the training sample distribution that significantly reduces memory and time complexity, while providing better diversity of samples; (iii) a conservative model update strategy with improved robustness and reduced complexity [108].

16. **DSST [258]:** The Discriminative Scale Space Tracker -DSST- extends the Minimum Output Sum of Squared Errors -MOSSE- tracker [165] with robust scale estimation. The DSST additionally learns a one-dimensional discriminative scale filter that is used to estimate the target size. For the translation filter, the intensity features employed in the MOSSE tracker are combined with a pixel-dense representation of HOG-features [108].

17. **VITAL [259]** carries out tracking using adversarial learning. It uses a generative network to generate masks for augmenting positive samples randomly. Masks are applied to adaptively dropout input features and capture a variety of appearance changes. With adversarial learning, the VITAL network identifies the mask that maintains the most robust features of the target objects over a long temporal span. It also proposes a high-order cost-sensitive loss, decreasing the effect of easy negative samples and facilitating the training of the classification network. In this way, class imbalance issues are handled.

C Recording Equipment Specifications

In this section, we explained with details the hardware specifications of the four video surveillance cameras used to capture the videos of AD-SVD dataset [11].

Table 29: IP8165HP VIVOTEK Camera Specifications [16].

• 2-Megapixel CMOS Sensor. • 30 fps @ 1920x1080, 60 fps @ 1920x1080 (one-stream mode only). • Motorized, P-iris Lens. • 3D Noise Reduction for Low-light Conditions WDR Pro (100dB) to Provide Extreme Visibility in High Light Contrast Scenes. • RBF Design to Assist for Precise Focus Adjustment. • EIS (Electronic Image Stabilization) to Control • Image Stability. • Two-way Audio. • Snapshot Focus. • Video Rotation for Corridor Format.

Table 30: IB8367A VIVOTEK Camera Specifications [17].

• 30 fps @ 1920x1080. • Removable IR-cut Filter for Day & Night Function. • Built-in IR Illuminators, Effective up to 30 Meters. • Supports ONVIF Standard to Simplify Integration and Enhance Interoperability. • Smart Stream II to Optimize Bandwidth Efficiency. • SNV (Supreme Night Visibility). • 3D Noise Reduction. • Weather-proof IP66-rated and Vandal-proof IK10-rated Metal Housing. • VIVOCloud App & Portal for 24/7 Surveillance. • Trend Micro IoT Security within Standard Warranty Period.

Table 31: IB8381 VIVOTEK Camera Specifications [18].

- 25 fps @ 2560x1920, 30 fps @ 1920x1080.
- 3 - 9 mm Vari-focal, P-iris Lens.
- Removable IR-cut Filter for Day & Night Function.
- Built-in IR Illuminators, Effective up to 30 Meters.
- Two-way Audio.
- WDR Enhancement for Unparalleled Visibility in Extremely Bright and Dark Environment.
- Smart Focus System for Remote and Precise Focus Adjustment.
- Weather-proof IP67-rated Housing.
- Built-in SD/SDHC/SDXC Card Slot for On-board Storage.
- Mounting Bracket with Cable Management for Protected Installation.

Table 32: AXIS P14 Camera Specifications [19]

- 1920x1080 resolution.
- 2 Megapixel image sensor.
- Slim, lightweight bullet camera.
- HDTV 1080p at up to 60 fps.
- Built-in IR LEDs with OptimizedIR, 30 meters (98 feet).
- Lightfinder, WDR and Zipstream.
- Two lens alternatives.
- Forensic capture WDR technology.

D Distortions Specifications per Camera

In this section, we detailed the parameters configured to generate the in-capture distortions in the four video surveillance cameras used [11]. These parameters are shown in the Tables 33, 34, 35, and 36.

Table 33: IP8165HPA - VIVOTEK Distortion Specifications [11]

Distortion	Indoor		Outdoor	
	Focus Level	Exposure Level	Focus Level	Exposure Level
Pristine	Auto	Auto	Auto	Auto
Focus 1	M	Auto	M	Auto
Focus 2	M	Auto	M	Auto
Focus 3	M	Auto	M	Auto
Exposure 1	Auto	-0,6	Auto	0,6
Exposure 2	Auto	-1,3	Auto	1,3
Exposure 3	Auto	-2	Auto	2
Focus 1 + Exposure 1	M	-0,6	M	0,6
Focus 1 + Exposure 3	M	-2	M	2
Focus 3 + Exposure 3	M	-2	M	2

Table 34: IB8367A - VIVOTEK Distortion Specifications [11].

Distortion	Indoor		Outdoor	
	Focus Level	Exposure Level	Focus Level	Exposure Level
Pristine	Auto	Auto	Auto	Auto
Focus 1	M	Auto	M	Auto
Focus 2	M	Auto	M	Auto
Focus 3	M	Auto	M	Auto
Exposure 1	Auto	-0,7	Auto	0,7
Exposure 2	Auto	-1,3	Auto	1,3
Exposure 3	Auto	-2	Auto	2
Focus 1 + Exposure 1	M	-0,7	M	0,7
Focus 1 + Exposure 3	M	-2	M	2
Focus 3 + Exposure 3	M	-2	M	2

Table 35: IB8381 - VIVOTEK Distortion Specifications [11].

Distortion	Indoor		Outdoor	
	Focus Level	Exposure Level	Focus Level	Exposure Level
Pristine	Auto	Auto	Auto	Auto
Focus 1	M	Auto	M	Auto
Focus 2	M	Auto	M	Auto
Focus 3	M	Auto	M	Auto
Exposure 1	Auto	-0,6	Auto	0,6
Exposure 2	Auto	-1,3	Auto	1,3
Exposure 3	Auto	-2	Auto	2
Focus 1 + Exposure 1	M	-0,6	M	0,6
Focus 1 + Exposure 3	M	-2	M	2
Focus 3 + Exposure 3	M	-2	M	2

Table 36: P14 - Axis Communications Distortion Specifications [11].

Distortion	Indoor		Outdoor	
	Focus Level	Exposure Level	Focus Level	Exposure Level
Pristine	Auto	Auto	Auto	Auto
Focus 1	M	Auto	M	Auto
Focus 2	M	Auto	M	Auto
Focus 3	M	Auto	M	Auto
Exposure 1	Auto	1/120	Auto	1/500
Exposure 2	Auto	1/250	Auto	1/150
Exposure 3	Auto	1/500	Auto	1/120
Focus 1 + Exposure 1	M	1/120	M	1/500
Focus 1 + Exposure 3	M	1/500	M	1/120
Focus 3 + Exposure 3	M	1/500	M	1/120