IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received 1 June 2023, accepted 26 June 2023, date of publication 5 July 2023, date of current version 11 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3292539

RESEARCH ARTICLE

Online Action Detection in Surveillance Scenarios: A Comprehensive Review and Comparative Study of State-of-the-Art Multi-Object Tracking Methods

JUMABEK ALIKHANOV, (Student Member, IEEE), AND HAKIL KIM, (Member, IEEE)
Department of Electrical and Computer Engineering, Inha University, Incheon 402-751, South Korea

Corresponding author: Hakil Kim (hikim@inha.ac.kr)

This research was supported by the Brain Korea (BK) 21 Four Program funded by the Ministry of Education (MOE) and National Research Foundation (NSF) of Korea.

ABSTRACT Online action detection in surveillance scenarios presents considerable challenges, particularly due to the dynamically changing environments and real-time processing requirements. Within this context, Multi-Object Tracking (MOT) serves as a critical component of the online action detection pipeline. Despite the emergence of several state-of-the-art (SOTA) object trackers in recent years, a gap remains in the comprehensive evaluation of these trackers specifically for action detection in surveillance scenarios. This paper bridges this gap by offering a thorough study of SOTA MOT trackers, aimed at determining the influential factors affecting their performance in surveillance settings and identifying the trackers optimally suited for an online action detection pipeline. For relevance and rigor, we introduce SurvTrack, a new dataset derived from a subset of VIRAT—dataset explicitly designed for action detection tasks—but intended for object tracking. SurvTrack is utilized to assess these trackers under various conditions, including differing image resolutions and detector confidence thresholds. This study uncovers the distinctive strengths and weaknesses of each tracker, providing invaluable insights for researchers and practitioners in surveillance and action detection. Importantly, this work focuses on tracking methods within the action detection domain, underscoring the development of a tracker explicitly designed for action detection on pertinent datasets, such as VIRAT.

INDEX TERMS Video surveillance, object detection, multi-object tracking, action detection, deep learning, computer vision.

I. INTRODUCTION

Multi-object tracking via computer vision is a fundamental task that involves detecting and tracking multiple objects from video streams over time. It has numerous real-world applications, such as security surveillance, traffic monitoring, and human-computer interaction. The goal is to accurately track multiple objects while handling challenges like occlusions, appearance changes, and interactions between tracked objects. This requires development of sophisticated methods that combine techniques from object detection, tracking, and data association. Multi-object tracking remains an active research area with ongoing efforts to improve accuracy, efficiency, and robustness. Many problems such as video surveillance, autonomous cars, action recognition, crowd behaviour analysis would benefit from a high-quality tracking algorithm [1]. For instance, recently, one of the key uses of multi-object tracking is action detection, where trackers generate potential action proposals [2], [3], [4]. Action detection in surveillance scenarios is a critical task, and the Video Image Retrieval and Analysis Tool (VIRAT) dataset is one of the most suitable video datasets for this purpose. An action

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

detection system comprises multiple components: object detection, multi-object tracking, proposal generation, and action classification [2], [3], [4]. Therefore, a tailored multi-object tracker is essential for achieving a high-performing action detection system in surveillance settings.

However, the majority of state-of-the-art (SOTA) trackers are not specifically designed or evaluated for CCTV surveillance like video from VIRAT dataset [5]. Instead, they focus on improving performance on particular datasets, such as MOT17 [27], MOT20 [13], CrowdHuman [28], Citypersons [29], ETHZ [30], DanceTrack [31]. It is well-established that a method that performs well on one dataset does not necessarily perform well on another. Research knowledge can be increased when we evaluate SOTA trackers in the same setting by a third party - for instance, in a setting that uses the same detector for all trackers. Moreover, state-of-the-art methods are typically evaluated in an offline setting by interpolating predicted tracks, which generally increases performance. Consequently, these trackers are not optimal candidates for action detection in surveillance scenarios that require online tracking, that have low-quality camera views, or that feature small scales objects. The goal of this work is to enable researchers to properly select a multi-object tracker for action detection. To this end, we perform a thorough comparison of state-of-the-art trackers from multiple perspectives on a subset of the VIRAT dataset, which we named SurvTrack, and is representative of real-time action detection scenarios for the evaluation of trackers. Fig. 2 highlights the scene, view and occlusion differences of MOT17 and SurvTrack datasets. Our contributions can be summarized as follows:

- We introduce the SurvTrack dataset, derived from the VIRAT dataset, to prepare trackers for online action detection.
- We compare state-of-the-art object trackers (Deep OC-SORT, ByteTrack, StrongSORT, and OC-SORT) on the SurvTrack dataset, focusing on online tracking in a CCTV action detection scenario.
- We analyze failure cases, the impact of appearance modeling, effects of image resolution, and the effect of detector confidence thresholds on tracker performance.

In this work, our scope is limited to multi-object visual tracking, using only camera input. Furthermore, we focus on online action detection scenario, which requires multi-object trackers to operate in an online setting. This means that evaluations are conducted without interpolating the entire length of the predicted tracks, which is common in offline learning settings. It is noteworthy to mention that our experiments and methods dealt with multi-object tracking, not action detection. however, the contribution of this work is to prepare trackers for action detection tasks, specifically in surveillance scenarios. Our code, implementation details, and the proposed dataset SurvTrack will be made publicly available upon publication at the following URL: https://github.com/Jumabek/SurvTrack to facilitate reproducibility and further research in this area.

The remainder of this paper is organized as follows. Section II discusses the relevant background for this study. Section II-E describes our procedure for deriving the SurvTrack dataset, and provides a brief explanation of multi-object trackers, evaluation metrics, and hardware settings. Section III presents the experiments and results, followed by a discussion of the findings in Section IV. Finally, Section V concludes the paper.

II. RELATED WORK

This section discusses the related work in the field of multi-object tracking, focusing on methods applicable to surveillance scenarios such as those found in the VIRAT dataset. We also briefly address the significance of action detection, and the components involved in these systems.

A. ACTION DETECTION

Action detection is a crucial research area in computer vision, with applications spanning security surveillance, human-computer interaction, sports analysis, and more. An action detection system typically consists of several components, including object detection, multi-object tracking, proposal generation, and action classification [2], [3], [4]. Fig. 1 highlights the multiple components involved in the action detection pipeline, including multi-object tracking. It is important to note that activity recognition is another component of the pipeline, which can also be considered as a standalone application. For instance, there is ongoing research in the field of human activity recognition [6], [7]. Activity recognition is also being actively utilized in elderly care facilities [8], [9], demonstrating its significance in various domains. To achieve high performance in action detection systems, particularly in surveillance scenarios, a tailored multi-object tracker is essential.

B. OBJECT TRACKING

Early object tracking techniques can be categorized into different classes, including template-based methods, feature-based methods, motion-based methods, and appearance-based methods [18]. However, these methods often struggle with challenges like occlusions, appearance changes, and varying object scales, which are common in surveillance scenarios. Later, deep learning-based tracking methods emerged, leveraging the capabilities of neural networks in order to enhance tracking performance. Notable approaches include Siamese networks, R-CNN-based trackers, and the learning of embeddings. While these methods have demonstrated remarkable results on various datasets [17], their performance on surveillance datasets like VIRAT remains underexplored.

In the last 2 years several SOTA MOT methods have emerged, including BoT-SORT [19], StrongSORT [20], Deep OC-SORT [21], Observation-Centric SORT [22], and ByteTrack [23]. These methods employ strategies such as data association, trajectory prediction, and online learning. However, these SOTA trackers have not been specifically

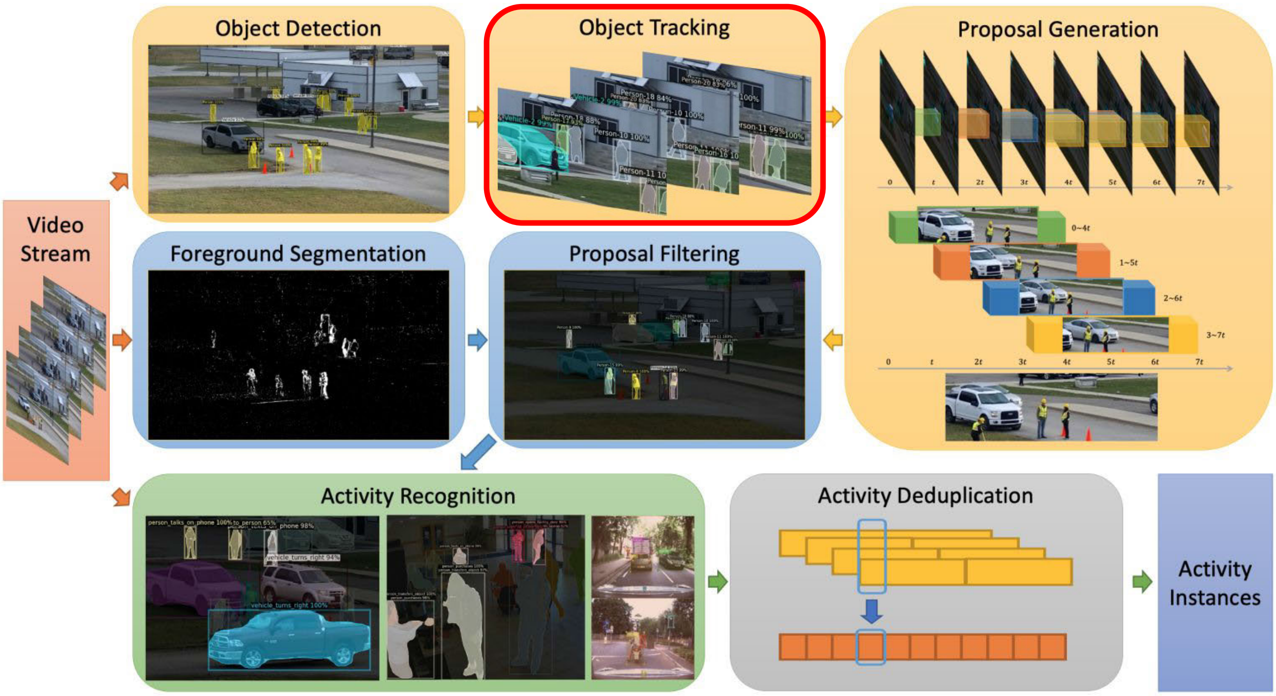


FIGURE 1. Architecture of Argus++. A video stream is processed frame-by-frame through object detection and tracking to generate overlapping cube proposals. With frame-level foreground segmentation, stable proposals are filtered out. Activity recognition models determine the classification scores for each proposal. These over-sampled cubes are deduplicated to produce the final activity instances. Adapted from [4]. This architecture highlights the position of multi-object tracking within the action detection pipeline.

designed or evaluated for surveillance scenarios like the VIRAT dataset. Instead, they concentrate on enhancing performance on particular datasets as mentioned earlier. A method excelling in one dataset may not necessarily perform well in another, making it crucial to evaluate these methods on surveillance-specific datasets.

C. MULTI-OBJECT TRACKING AND SEGMENTATION

In addition to Multi-Object Tracking (MOT), there is an extension called Multi-Object Tracking and Segmentation (MOTS) that combines tracking with pixel-level segmentation. MOTS aims to provide richer information about the scene by not only identifying the objects but also segmenting them at the pixel level. Voigtlaender et al. introduced the concept of MOTS and prepared pixel-level annotations for existing datasets such as KITTI and MOT Challenge [10]. These annotated datasets are named KITTI MOTS and MOTS Challenge, respectively. By providing pixel-level annotations, MOTS allows for more comprehensive analysis and applications, enabling a deeper understanding of the scene beyond traditional 2D bounding boxes. Another significant dataset related to autonomous driving is BDD100K, which consists of 100,000 videos and 10 tasks for evaluating computer vision algorithms in the context of autonomous driving [11]. Although BDD100K encompasses various tasks, it also includes annotations for object detection, instance segmentation, and tracking, making it relevant for evaluation in the MOTS domain. Cui et al. proposed the DGL-MOTS dataset, which focuses on capturing diverse driving scenarios and surpasses KITTI MOTS and BDD100K in terms of annotation quality, data diversity, and temporal representation [12]. The DGL-MOTS dataset offers a broader range of situations and challenges, providing a valuable resource for developing and evaluating advanced MOTS algorithms. These MOTS-related datasets, including KITTI MOTS, MOTS Challenge, BDD100K, and DGL-MOTS, play a vital role in advancing research on multi-object tracking and segmentation. They facilitate the development and evaluation of algorithms that can handle complex scenarios and provide precise and detailed object-level information.

D. OFFLINE vs. ONLINE TRACKING

Online object tracking refers to the task of locating and following objects of interest in a video as it unfolds, without prior knowledge of the future frames [16]. Many SOTA methods are evaluated in an offline setting where predicted tracks are interpolated, generally leading to improved performance. However, this approach may not be suitable for action detection in surveillance scenarios, where online tracking is necessary and when camera views are often of low quality with small-scale objects.

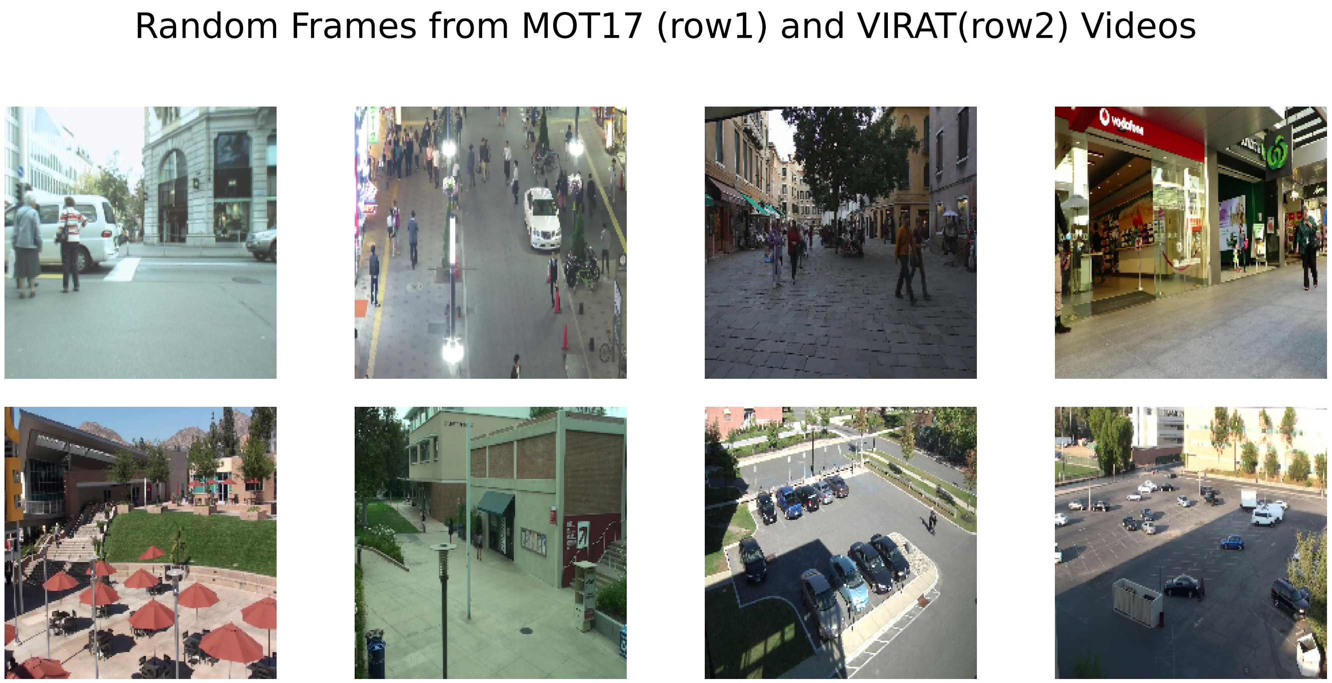


FIGURE 2. Representative frames from (first row) the MOT17 and (second row) the SurvTrack - a subset of VIRAT dataset.

E. GENERAL MOT DATASETS VERSUS SURVEILLANCE SCENARIOS

Surveillance involves various computer vision applications, including action detection. Unlike object detection or tracking, action detection is a more complex and higher-level task that helps machines understand the scene. To design a well-performing action detection system in such scenarios, a multi-object tracker tailored for surveillance scenarios is required. However, most MOT benchmark datasets are not challenging enough for surveillance scenarios. Hence, there is a lack of representative datasets in surveillance scenarios. Although the VIRAT dataset is a good candidate for this task, it requires extra steps to prepare it for MOT experiments, and there is a lot of misannotation such as ghost annotation bounding boxes that do not correspond to any object. To the best of our knowledge, no prior work has explored the performance of state-of-the-art trackers for surveillance scenarios. In general, current benchmark datasets fail to cover the following conditions that are present in surveillance scenarios.

- Camera Viewpoints: Many MOT datasets are captured from a single fixed camera viewpoint, which does not reflect real-world surveillance scenarios where cameras are often placed at different angles and positions.
- Object Types: MOT datasets typically focus on tracking a limited set of object types, such as pedestrians and vehicles, which may not be representative of the full range of objects that surveillance systems need to track.
- Object Scale: Many MOT datasets have a limited range of object scales, which may not reflect the full range of object sizes encountered in real-world surveillance.
- Lighting Conditions: MOT datasets are often captured under controlled lighting, which may not reflect the wide range of lighting conditions encountered in real-world surveillance.
- Occlusion and Clutter: Many MOT datasets do not include enough occlusions and clutter which is common in real-world surveillance scenarios. Even though the presence of such challenges can greatly affect the performance of tracking algorithms.
- Data Variety: Many MOT datasets have limited variety, such as the number of scenes, objects, and frames, which may not reflect the full range of scenarios encountered in real-world surveillance.

In summary, although various SOTA trackers have been developed, their suitability for surveillance scenarios like the VIRAT dataset remains largely unexplored. The objective of this work is to enable researchers to make informed decisions when selecting a multi-object tracker for action detection in surveillance scenarios. To achieve this, a comprehensive comparison of SOTA trackers on the SurvTrack dataset, a representative dataset for real-time action detection is conducted.

III. METHODOLOGY

A. MULTI-OBJECT TRACKERS

The following multi-object trackers exhibit a range of methods and have demonstrated strong performance with various benchmarks:

- **ByteTrack**: ByteTrack is a simple, effective, and generic association method that focuses on associating almost every detection box, including low-score ones,

TABLE 1. Comparison of methodology among different MOT trackers.

Tracker	Motion Model	Appearance Model	FPS
ByteTrack	Kalman filter	None	1429
OC-SORT	Modified Kalman filter	None	1250
Deep OC-SORT	Modified Kalman filter	Deep appearance model	30
BoT-SORT	Accurate Kalman filter	ReID	50
StrongSORT	Kalman filter	Upgraded DeepSORT	38

to recover true objects and filter out background detections. It achieves state-of-the-art performance on multiple tracking benchmark datasets, including MOT17, MOT20, HiEve, and BDD100K.

- **OC-SORT**: OC-SORT is an observation-centric method that improves upon the basic Kalman filter to obtain state-of-the-art tracking performance. By using object observations to compute a virtual trajectory during occlusion, it corrects errors accumulated in the filter parameters. OC-SORT maintains simplicity, online operation, and real-time performance, achieving state-of-the-art results on multiple datasets.
- **Deep OC-SORT**: Deep OC-SORT extends the motion-based OC-SORT by adaptively integrating appearance matching using deep appearance features. in MOT competition it achieved 1st place when using the MOT20 and 2nd place when using MOT17 with good higher-order tracking accuracy (HOTA) scores and it set new SOTA performance on the DanceTrack benchmark dataset.
- **BoT-SORT**: BoT-SORT combines motion and appearance information, camera motion compensation, and an accurate Kalman filter state vector to create a robust tracker. It ranked first on MOT17 and MOT20 test sets in terms of MOTA, IDF1, and HOTA metrics.
- **StrongSORT**: StrongSORT revisits the classic DeepSORT tracker and improves upon it in terms of detection, embedding, and association. Combined with the appearance-free (AFLink) model and Gaussian-smoothed interpolation (GSI), StrongSORT++ achieved top ranking on MOT17 and MOT20 datasets in terms of HOTA and IDF1 metrics [20].

It is important to note that, among the aforementioned trackers, ByteTrack and OC-SORT do not have an appearance model (Table 1). Other trackers use some variant of appearance modeling, while all trackers use a Kalman filter or its modified version for motion modeling. Since there is no extra step for appearance modeling, ByteTrack and OC-SORT enjoy high processing speed.

B. EVALUATION OF MULTI-OBJECT-TRACKING METHODS

1) ERROR TYPES

Evaluation metrics are derived from the types of errors made during tracking. Commonly employed error types from [15] are used as shown in Fig. 3.

2) EVALUATION METRICS

In the following, the three most common MOT metrics for evaluating multi-object tracking approaches are described, including their formula.

a: MULTIPLE OBJECT TRACKING ACCURACY

MOTA is a popular metric used to evaluate the overall performance of a multi-object tracking system. It considers factors such as detection accuracy, false positives, missed detections, and ID switches. MOTA does not include a measure of localization error, and detection performance significantly outweighs association performance. The equation for MOTA can be seen in (1) where FN_t represents the number of false negatives, FP_t denotes the number of false positives, IDSW_t indicates the number of identity switches, and GT_t signifies the number of ground-truth objects in frame t.

$$\mathrm{MOTA} = 1 - \frac{\sum_t (\mathrm{FN}_t + \mathrm{FP}_t + \mathrm{IDSW}_t)}{\sum \mathrm{GT}_t} \tag{1}$$

b: MULTIPLE OBJECT TRACKING PRECISION

MOTP is a metric that measures the precision of the estimated object positions. It calculates the average distance between the ground-truth and estimated object positions. The equation for MOTP is shown in (2) where $d_{t,i}$ is the Euclidean distance between the predicted position of object i and its ground-truth position in frame t, and C_t is the number of matches in frame t.

$$\mathrm{MOTP} = \frac{\sum_t \sum_i d_{t,i}}{\sum_t C_t} \tag{2}$$

c: ID-F1 SCORE

This metric is used to evaluate the accuracy of object tracking over time. It measures the percentage of correctly tracked object trajectories, taking into account the number of true positives and false positives. The equation for IDF1 in (3) where TP_i is the number of true positives for object i, FP_i is the number of false positives, and FN_i is the number of false negatives.

$$\mathrm{IDF1} = 2 \cdot \frac{\sum_i \mathrm{TP}_i}{2 \cdot \sum_i \mathrm{TP}_i + \mathrm{FP}_i + \mathrm{FN}_i} \tag{3}$$

3) PRIMARY METRIC

For an online activity detection application, maintaining consistent object identities across frames can be critical to accurately recognizing and analyzing the activities performed. In this case, IDF1 score would be a more appropriate primary metric.

C. DATASETS

1) VIRAT DATASET

VIRAT [26] is a large-scale video dataset that was collected to evaluate the methods of video understanding in a ground-based surveillance scenario. The dataset consists of high-resolution, full-motion video recorded from fixed and

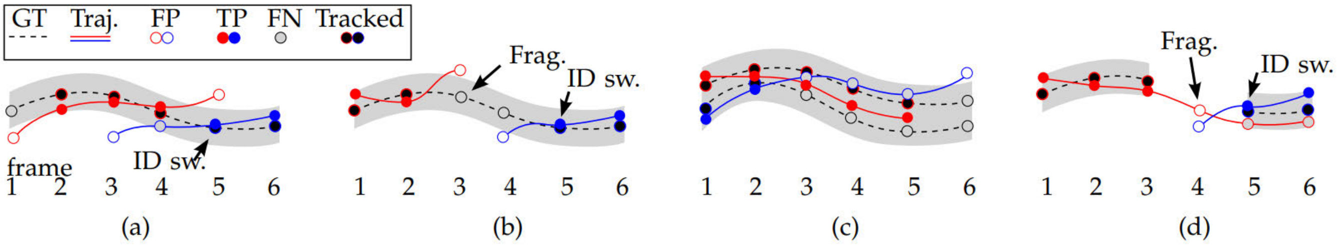


FIGURE 3. Four cases illustrating tracker-to-target assignments. (a) An ID switch occurs when the mapping switches from the previously assigned red track to the blue one. (b) A track fragmentation is counted in frame 3 because the target is tracked in frames 1-2, then interrupts, and then reacquires its 'tracked' status at a later point. A new (blue) track hypothesis also causes an ID switch at this point. (c) Although the tracking results is reasonably good, an optimal single-frame assignment in frame 1 is propagated through the sequence, causing 5 missed targets (FN) and 4 false positives (FP). Note that no fragmentations are counted in frames 3 and 6 because tracking of those targets is not resumed at a later point. (d) A degenerate case illustrating that target re-identification is not handled correctly. An interrupted ground truth trajectory will typically cause a fragmentation. Also note the less intuitive ID switch, which is counted because blue is the closest target in frame 5 that is not in conflict with the mapping in frame 4 (adapted from [15]).

moving cameras in various outdoor environments, including parking lots, roads, and airport tarmacs. The VIRAT dataset was a pivotal choice for our research due to its distinctive application in action detection, particularly in the widely recognized MEVA challenge [5]. Moreover, VIRAT's provision of bounding box information for every frame facilitated the creation of the SurvTrack dataset, which specifically caters to object tracking tasks. Notably, VIRAT stands out as the sole dataset known to us that has been utilized for action detection in the surveillance domain, while also being adaptable for multi-object tracking purposes.

The videos in the VIRAT dataset feature a variety of real-world scenarios, including pedestrians, vehicles, bicycles, and other objects, making it an ideal resource for evaluating multi-object tracking methods in a CCTV setting. The videos in the dataset are annotated with object-level information, including object type, position, and appearance, making it easy to evaluate the performance of tracking methods. In terms of its utility for multi-object tracking in CCTV scenarios, the VIRAT dataset provides a realistic, challenging benchmark for evaluating the performance of tracking methods in complex, real-world scenarios. The wide variety of objects and scenes in the dataset, combined with high-resolution video footage, makes it an ideal resource for researchers and practitioners working on multi-object tracking for CCTV applications. In this work, Release 2.0 version of the dataset is used for preparing our own SurvTrack dataset, which includes 11 scenes. Each scene contains multiple video clips and clips may contain activities from the 12 categories shown in Table 2.

It is noteworthy to mention that, although the VIRAT dataset contains information regarding activities shown in Table 2, in this work our scope is to use only annotated object tracks. The VIRAT dataset presents several challenges for multi-object tracking due to the following factors:

- **Crowded scenes:** Many videos feature crowded scenes with multiple people and vehicles moving in different directions, making it difficult to track individual objects.
- **Scale variations:** Objects can vary in size and scale, making it challenging to detect and track them consistently. This can be also seen in Figures 5 and 6.
- **Occlusions:** The presence of occlusions makes it difficult to track objects throughout the entire video sequence, particularly when objects are partially or fully occluded.
- **Lighting conditions:** Lighting conditions can vary significantly, especially in outdoor scenarios, making it challenging to detect and track objects accurately.
- **Camera angle:** as CCTV cameras are usually mounted in ceilings at certain angle, this creates extra challenges for computer vision applications

TABLE 2. Complete list of activities in the VIRAT dataset.

No.	Activity
1	Person loading an object to a vehicle
2	Person unloading an object from a car/vehicle
3	Person opening a vehicle/car trunk
4	Person closing a vehicle/car trunk
5	Person getting into a vehicle
6	Person getting out of a vehicle
7	Person gesturing
8	Person digging
9	Person carrying an object
10	Person running
11	Person entering a facility
12	Person exiting a facility

2) THE SurvTrack DATASET

To address the lack of datasets tailored for CCTV surveillance scenarios and that are applicable to action detection, we created a new dataset, named SurvTrack which is a subset of the VIRAT dataset. The following describes the process of deriving the subset and the exploratory analysis of SurvTrack.

a: EXCLUSION OF LENGTHY VIDEOS

It is noteworthy that some videos in the VIRAT dataset are considerably longer than those in the MOT17 benchmark dataset, which contains videos of up to 1500 frames. As depicted in Fig. 4, we excluded from the study the majority

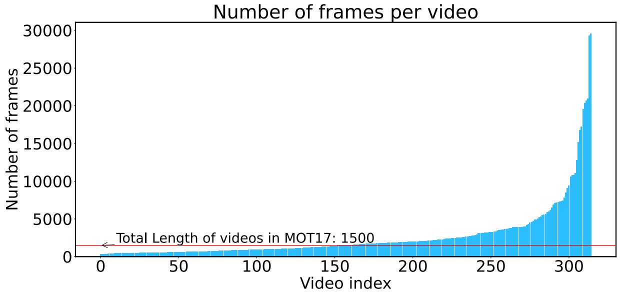


FIGURE 4. Video lengths of Virat dataset.

TABLE 3. Videos with incomplete annotations in the dataset.

No.	Video Name
1	VIRAT_S_000201_02_000590_000623.mp4
2	VIRAT_S_000201_03_000640_000672.mp4
3	VIRAT_S_000201_06_001354_001397.mp4
4	VIRAT_S_000203_03_000736_000785.mp4
5	VIRAT_S_000203_08_001702_001734.mp4
6	VIRAT_S_000207_00_000000_000045.mp4
7	VIRAT_S_000207_02_000498_000530.mp4
8	VIRAT_S_000207_03_000556_000590.mp4
9	VIRAT_S_000207_04_000902_000934.mp4

of videos that exceeded 1500 frames. Out of the 329 videos, we considered 156 videos with frame lengths below 1500.

b: EXCLUSION OF VIDEOS WITH INCOMPLETE ANNOTATIONS

After selecting shorter videos, we identified some videos with incomplete annotations. By inspecting each video's annotations, we discovered nine videos that were incomplete in Table 3.

c: EXPLORATORY DATA ANALYSIS

In total, our SurvTrack dataset contains 145 video files extracted from the VIRAT dataset. Among these, 131 have a resolution of 1280 × 720, while the remaining 14 are 1920 × 1080.

The annotation sizes are visualized as a heatmap in Fig. 5. The majority of object widths range between 40 and 80 pixels, with heights varying from 20 to 50 pixels. These dimensions are relatively small compared to the 1280 × 720 resolution, thus making tracking more challenging in this dataset. Fig. 7 depicts the distribution of classes in the SurvTrack dataset. It is noteworthy to mention that for simplification, we ignored object and bicycle classes they are in the minority. For comparison, heat map of object scales in MOT17 dataset is presented in Fig. 6. As can be seen, most of the objects (peoples) in MOT17 have heights larger than 40 pixels, whereas in SurvTrack, the majority of objects are smaller than 40 pixels. This difference can be attributed to cameras in SurvTrack videos where viewing from an angle makes humans look different. Another explanation could be that SurvTrack includes vehicles, and although vehicles are usually wider than humans, they are shorter.

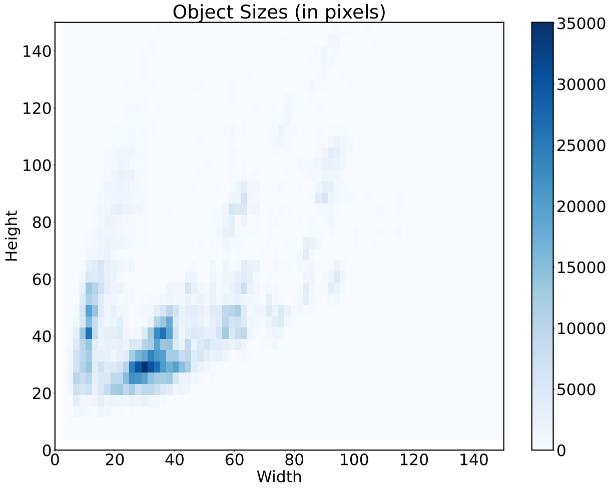


FIGURE 5. Heatmap of object sizes in SurvTrack dataset.

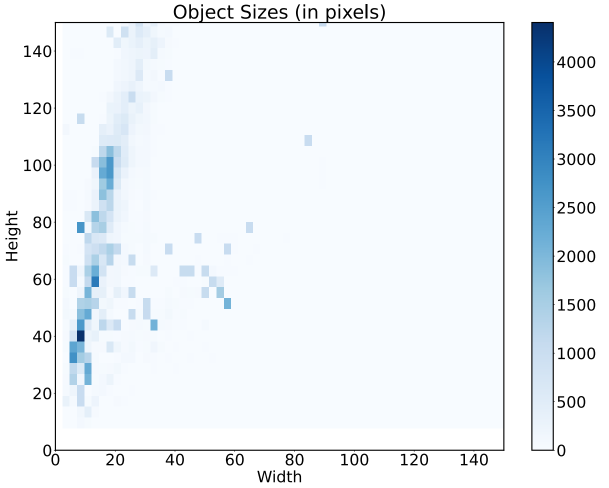


FIGURE 6. Heat map of object sizes in the MOT17 dataset.

D. EXPERIMENTAL SETTING

YOLOv8m model is used for performing the detection stage of all trackers. Unless otherwise mentioned, throughout the experiment, the input image resolution in the YOLO model was 640 × 640, and the minimum detection confidence was set to 0.5. Deep learning models were executed for inference with FP32 precision.

All experiments were conducted on a system equipped with the following hardware specifications:

- CPU:12th-Gen Intel Core i9-12900K at 5.2 GHz
- GPU: NVIDIA GeForce RTX 3090 with 24 GB VRAM
- RAM: 64 GB, DDR4 at 3200 MHz
- Storage: 1 TB SSD

The experiments were performed using the following software:

- Ubuntu 22.04.4 LTS

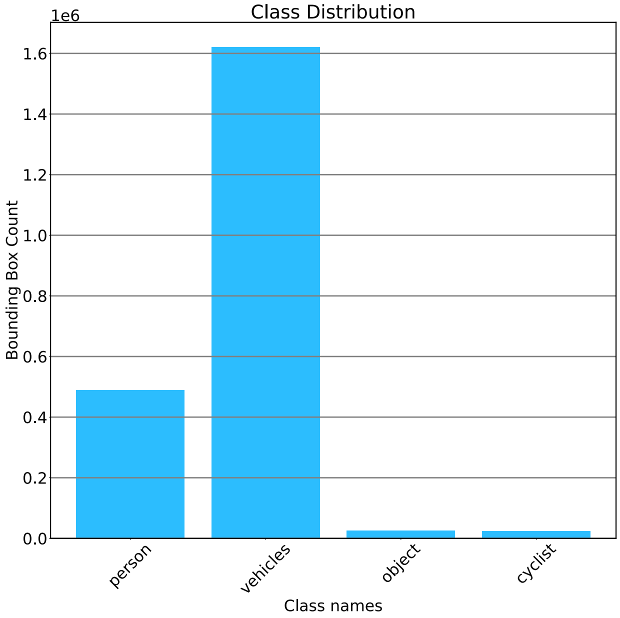


FIGURE 7. Annotated bounding boxes for each class in the SurvTrack dataset. Note, object and cyclist classes are eliminated in the experiment due to having 20× less samples in comparison to person class.

- Python Version 3.8.10
- PyTorch Version 1.10.0
- CUDA Version 11.5
- cuDNN Version 8.3.1

IV. EXPERIMENTS AND RESULTS

A. COMPARISON OF TRACKERS WITH DEFAULT SETTINGS

In this experiment, we compared the performance of the selected state-of-the-art (SOTA) MOT trackers using their default settings [14]. The evaluation was based on commonly used metrics such as MOTP, MOTA, IDF1, and FPS.

The results presented in Fig. 8 show that the overall performance of trackers on the SurvTrack dataset using these default settings was much worse than results with other benchmark datasets [19], [20], [21], [22], [23]. Our second observation is that ByteTrack had significantly poorer performance on SurvTrack compared to its performance on the MOT17 and MOT20 datasets, where it was on par with OC-SORT [22]. Third, there is no significant difference in terms of performance for trackers that exploit re-identification ReID modules in addition to motion modeling. For instance, OC-SORT had performance on par with StrongSORT and Deep OC-SORT.

Since we have 145 videos in the dataset, it is useful to provide error bars to indicate how performance varied across different videos. Recall that in a normal distribution, approximately 68% of the data lies within one standard deviation (1σ) to the left and right of the mean (μ). This is a property of the normal distribution, also known as the Gaussian distribution or bell curve.

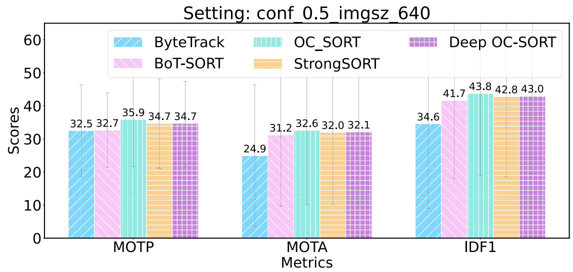


FIGURE 8. Accuracy comparison of SOTA trackers on the SurvTrack (subset of VIRAT) dataset.

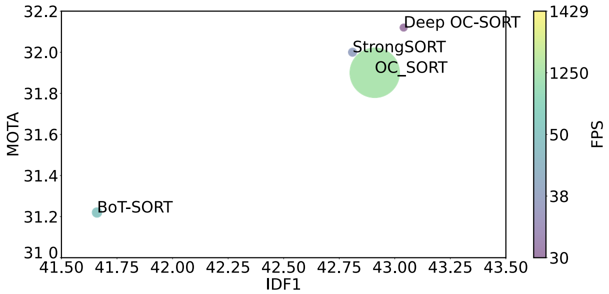


FIGURE 9. Accuracy versus speed for SOTA trackers on the SurvTrack dataset. Bubble size indicates the number of frames per second the tracker processed. The detector's processing time is not included because all trackers used the same detector.

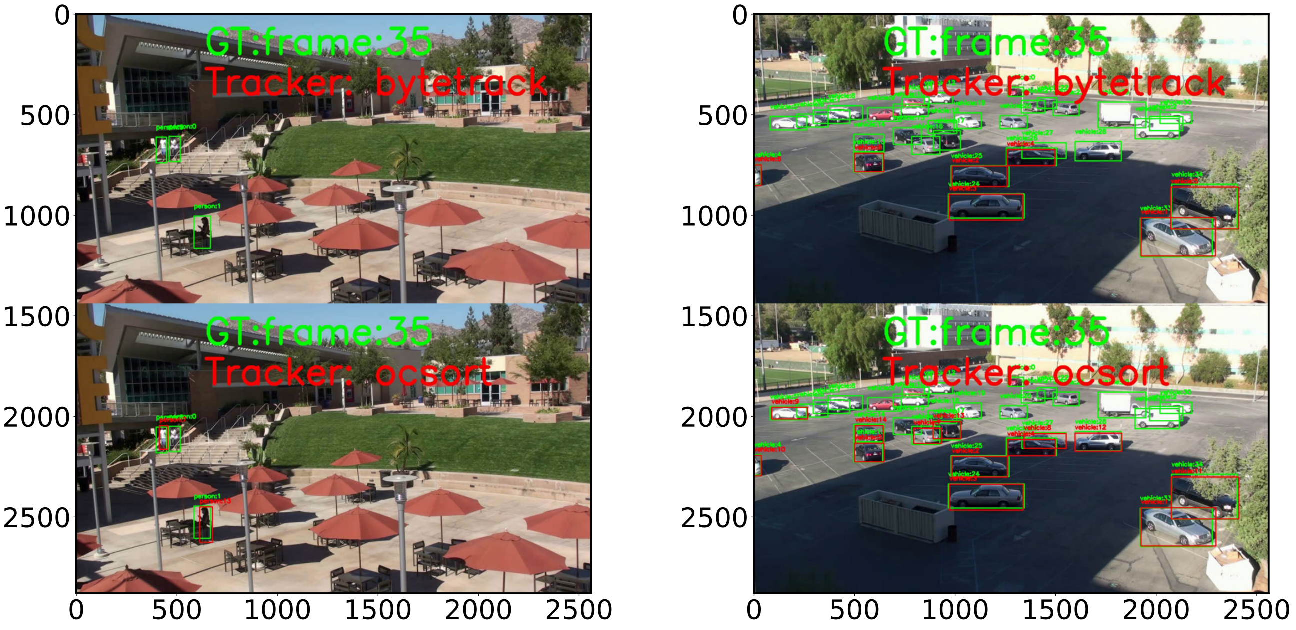
1) QUALITATIVE ANALYSIS

Results in the previous section discovered that ByteTrack has a notable performance gap (an IDF1 score of 34%) compared to OC-SORT (42.9%). This is significant because both consist of only motion modeling based on the Kalman filter. Below, we compare these two trackers qualitatively. We selected a representative video that had the highest performance difference: an IDF1 score of 11.8% for ByteTrack and IDF1 of 53.4% for OC-SORT.

Observations from the figure are, Bytetrack is missing the objects where the object scale is small and/or has illumination. In such cases, confidence of the detected object maybe low and due to the two-stage matching nature of ByteTrack, they use larger confidence for the first stage. Consequently, such low confidence detections may never contribute for instantiating a tracklet.

B. FAILURE CASE ANALYSIS

The previous section demonstrated that the trackers had significantly lower performance on the SurvTrack dataset compared to other benchmarks from the literature. In this section, we investigate why. In Table 4, we present the performance of five different multi-object tracking methods - BoT-SORT, ByteTrack, Deep OC-SORT, OC-SORT, and StrongSORT - on five selected videos from the VIRAT dataset. Table 4 reports IDF1 scores for each tracker-and-video combination. It can be observed that most of the trackers have relatively low IDF1 scores, indicating that they



(a) ByteTrack

(b) OC-SORT

FIGURE 10. Qualitative comparisons between ByteTrack and OC-SORT. Each top of the frame shows tracks from ByteTrack compared to the bottom frame from OC-SORT. For convenience, we only showed the sample frames from the video. A link to the full video is provided in Supplemental Materials.

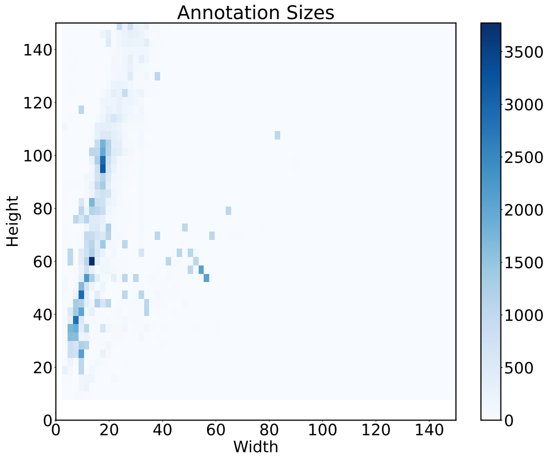


FIGURE 11. Object size distributions of the 10 most difficult videos for each tracker. Some of the videos were difficult for more than one tracker.

are not making many correct predictions. In particular, ByteTrack did not make any predictions at all in five cases. After further investigation we confirmed that there wasnt any tracking prediction is made owing to high confidence threshold and ByteTrack configuration.

Because not many predictions were made, a potential reason for low performance is the high confidence threshold of the YOLOv8 object detector, which was set at 0.5. With such a high threshold, the object detector may fail to produce a sufficient number of bounding box predictions, which, in turn, affects the trackers' ability to associate and track objects accurately. This issue is particularly evident in cases where the trackers have an IDF1 score of 0.0, indicating no predictions. To improve performance from these trackers, one possible solution is to adjust the confidence threshold of the YOLOv8 object detector, allowing for more bounding box predictions and consequently improving the ability to associate and track objects in the videos.

Another observation from Fig. 11 is that object scales in those videos are much smaller (around 20 pixel wide) than the average object scale (around 50 pixels wide) shown in Fig. 5 in Section II-E. This is especially troublesome for the ReID component of the trackers because at such a small scale, it is difficult to obtain descriptive embeddings for the given object. Therefore, another possible solution is to increase the resolution of input images so object scales become larger. We include links to the visualization of the worst and best-case scenarios in Supplementary Material.

C. EFFECT OF THE CONFIDENCE THRESHOLD

In this section, we investigate the impact of the detector confidence threshold on the performance of the multi-object trackers. The results are shown in Fig. 12, where each chart illustrates a specific confidence threshold. Additionality, for clarity, we also provide IDF results in Table 5. The input resolution used is 640 × 640.

The minimum threshold of the YOLOv8m detector model is kept at its default value of 0.5. While we provide all three metrics, we focus on the primary metric, IDF1.

TABLE 4. The 10 most difficult videos for each tracker based on IDF1 score.

Video ID	BoT-SORT	ByteTrack	Deep OC-SORT	OC-SORT	StrongSORT
VIRAT_S_010000_06_000728_000762	3.08	0.07	3.57	3.08	2.73
VIRAT_S_010000_07_000827_000860	6.02	0.65	7.54	7.14	6.31
VIRAT_S_010001_08_000826_000893	1.42	0.00	1.98	1.51	1.23
VIRAT_S_010002_07_000522_000547	2.37	0.00	3.73	3.15	1.78
VIRAT_S_010004_02_000191_000237	6.10	1.43	7.75	7.09	6.64
VIRAT_S_010005_04_000299_000323	6.42	0.00	8.79	8.04	7.35
VIRAT_S_010204_10_001372_001395	0.00	0.00	0.00	0.00	0.00
VIRAT_S_010206_02_000414_000439	4.37	0.00	4.53	4.37	3.15
VIRAT_S_010207_04_000929_000954	1.56	0.00	2.48	2.18	1.87
VIRAT_S_010207_05_001013_001038	0.00	0.00	0.00	0.00	0.00

TABLE 5. IDF1 scores of trackers at different detector confidence thresholds.

Detector Confidence Thresholds	ByteTrack	OC-SORT	BoT-SORT	Deep OC-SORT	StrongSORT
0.5	34.44	43.27	41.88	43.28	43.03
0.4	34.91	48.42	47.17	48.65	48.74
0.3	35.80	52.28	49.96	53.13	53.61
0.2	36.60	56.43	49.96	63.97	59.14
0.1	37.42	58.44	49.96	58.32	65.19

From the figure, it is clear that lower confidence thresholds produce better results in general. An exception occurs when switching the threshold from 0.2 to 0.1 where some trackers such as StrongSORT obtain better performance while others such as Deep OC-SORT lose performance. This indicates, even though all trackers use the same detector, the minimum confidence threshold affects them differently and should be set carefully depending on which tracker is being used.

The experiments showed that adjusting the confidence threshold of the YOLOv8 object detector can significantly impact the performance of multi-object trackers. While each tracker may have different sensitivities to the parameter, exploring its effect can help optimize a tracker's performance for a given application or dataset.

D. EFFECT OF IMAGE RESOLUTION

In this section, we conducted experiments by varying the input image resolution for both the detector and the tracker. The input resolutions were resized to a square aspect ratio, such as 640 × 640. We tested resolutions at 480, 640, 720, 1000, and 1280. Although we provide other metrics for completeness, our focus is on the IDF1 score. The results are illustrated in Fig. 13 and Fig. 15.

The first observation is that, in general, larger resolutions lead to better performance by trackers. The second observation is that sensitivity to image resolution is higher when the confidence threshold is set to 0.1. Third, different trackers favored different image resolutions. For instance, in Fig. 15, Deep OC-SORT achieved its best IDF1 performance of 80.1% with an image resolution of 720 × 720, while StrongSORT obtained its peak performance of 73.3% from the 1280 × 1280 resolution. It is interesting to see how an increase in input resolution affects the inference speed. In detection-based tracking pipeline, there are two components which are detector, and tracker. Figures 16 and 15 depict the effect of input resolution on inference speed of detector and update time of tracker.

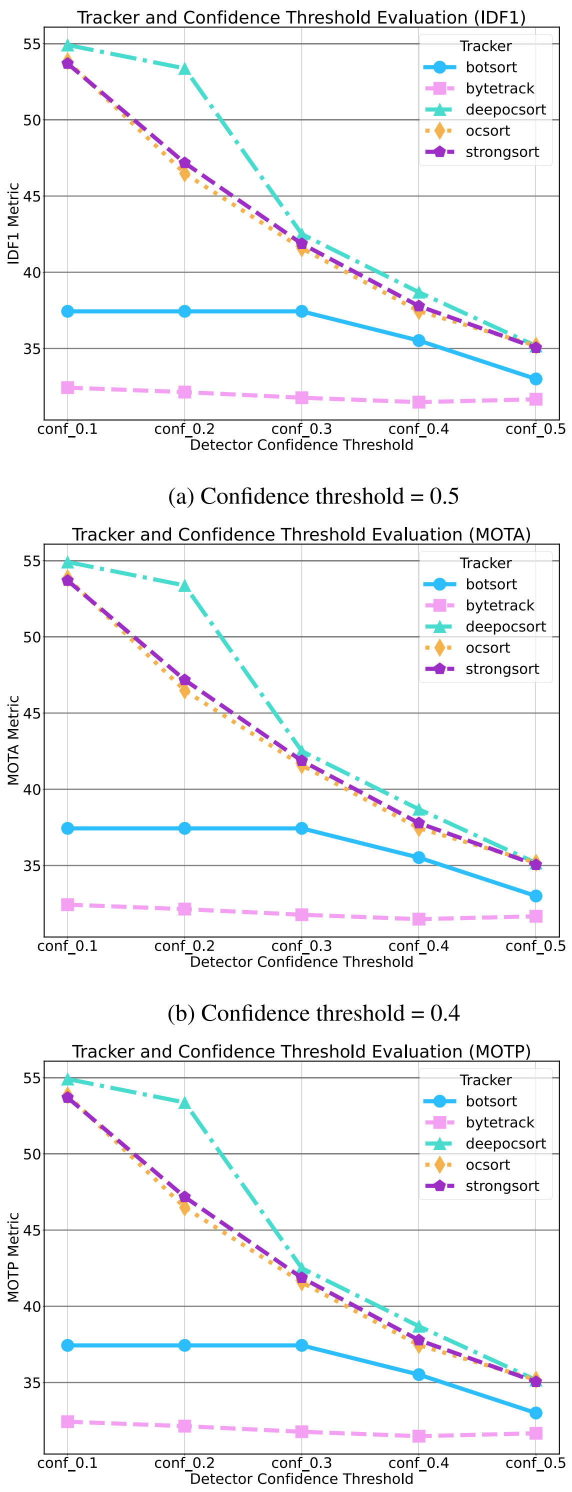
The main insight from this experiment is that by careful tuning of the affecting factors we can improve IDF1 score of 43% in a default scenario to an IDF1 score of 80.1% at a resolution of 720 × 720 pixels when using a detection confidence threshold of 0.1. Second, in terms of resolution, 720 × 720 could be considered the most suitable for SurvTrack because it provided a good trade-off between accuracy and computational complexity. Third, although increasing input resolution reduces speed of the pipeline, it is still good idea to use large resolution. This is because increasing input resolution only increases the inference time of the detector and has little effect on tracker update time. However, this depends on the speed of the tracker as well. For instance, for pure motion based trackers such as OC-SORT and ByteTrack detector inference is the main bottleneck for speed.

V. DISCUSSION

A. THE IMPORTANCE OF THE DATASET

Our results underscore the significance of selecting an appropriate dataset when evaluating multi-object trackers, particularly for surveillance scenarios. The performance of state-of-the-art trackers on the SurvTrack dataset, which closely represents real-world surveillance conditions, was substantially lower than performance on other benchmark datasets reported in the literature. This observation highlights the fact that a tracker's performance on one dataset may not necessarily generalize to other datasets or scenarios.

Choosing a suitable dataset is crucial for understanding the limitations of multi-object trackers and identifying areas where improvements can be made. With the SurvTrack dataset, we noted that object scale, object detection

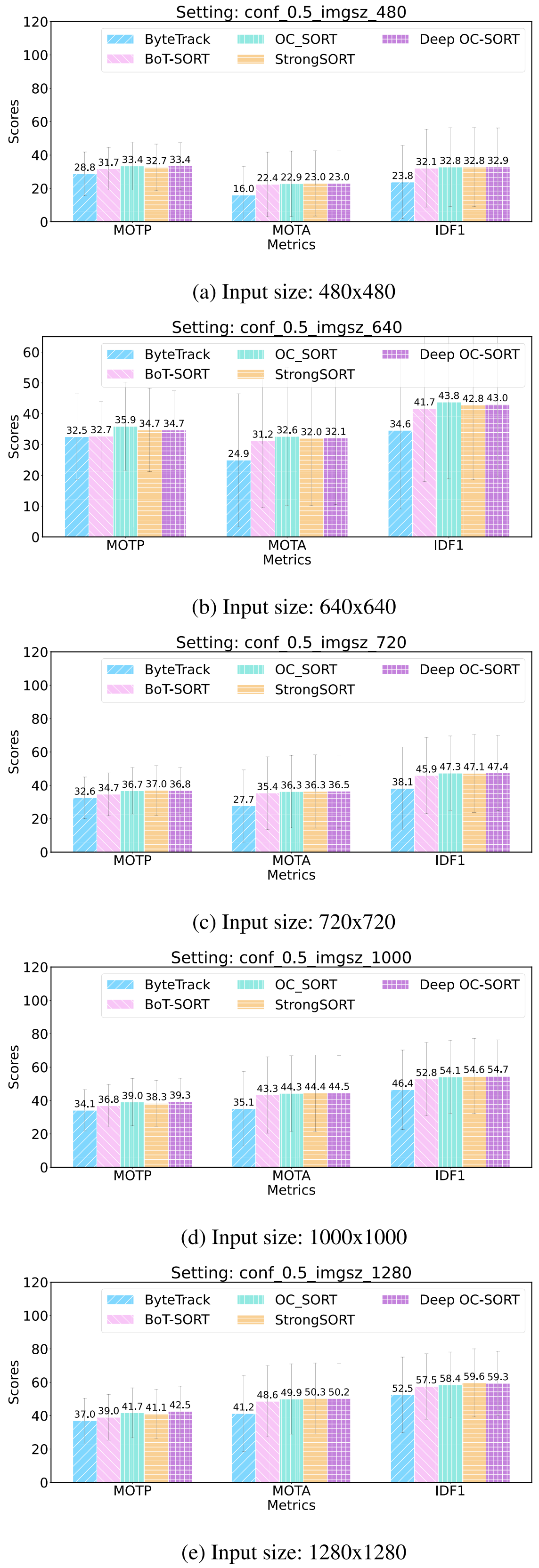


(a) Confidence threshold = 0.5

(b) Confidence threshold = 0.4

(c) Confidence threshold = 0.3

FIGURE 12. Comparison of results from different confidence thresholds.



(a) Input size: 480x480

(b) Input size: 640x640

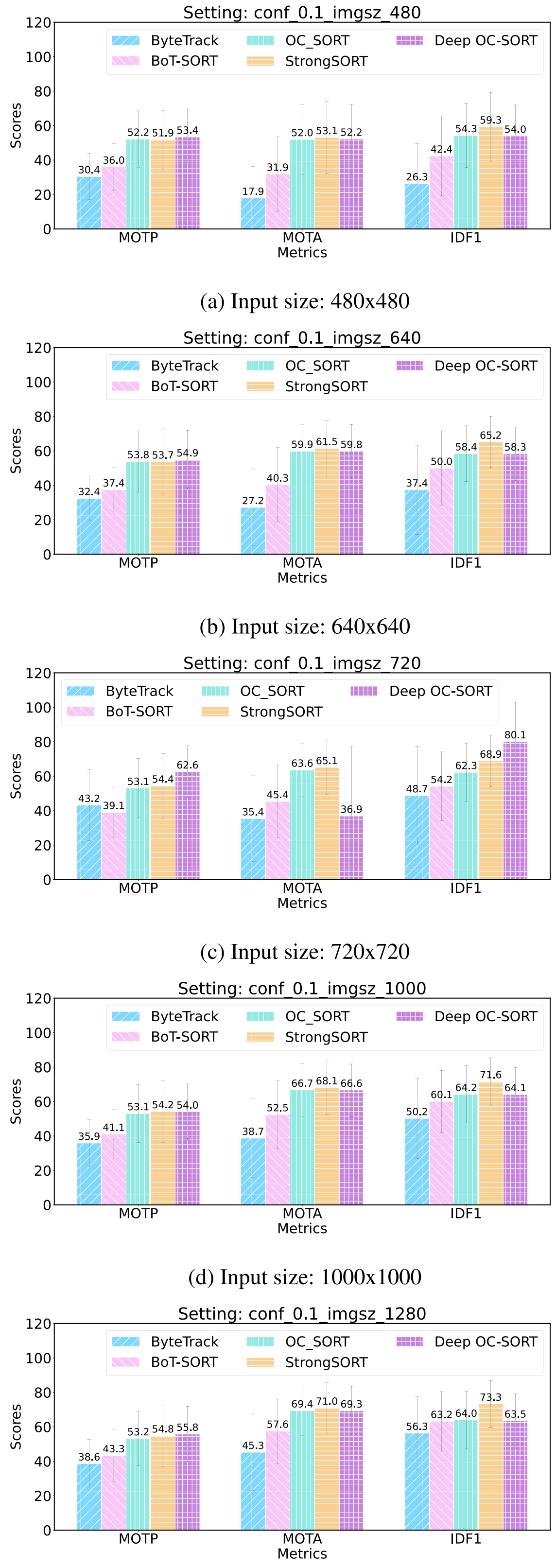
(c) Input size: 720x720

(d) Input size: 1000x1000

(e) Input size: 1280x1280

FIGURE 13. Confidence threshold: 0.5.

confidence threshold, and image resolution were factors contributing to lower performance from the trackers. By carefully analyzing these factors, we gain valuable insights into how to enhance tracker performance in real-world surveillance scenarios.



(a) Input size: 480x480

(b) Input size: 640x640

(c) Input size: 720x720

(d) Input size: 1000x1000

(e) Input size: 1280x1280

FIGURE 14. **Confidence threshold: 0.1**

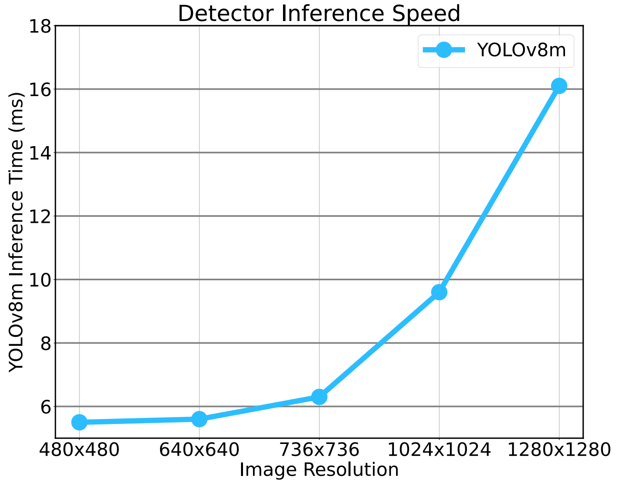


FIGURE 15. **Effect of input resolution on detector model inference speed. YOLOv8m architecture is used. Time is measured in ms for a single frame.**

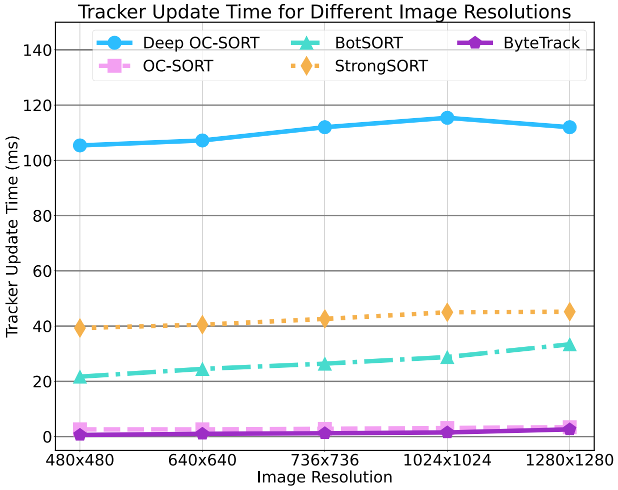


FIGURE 16. **Effect of input resolution on tracker update time. Time is measured in ms for a single frame.**

Moreover, the SurvTrack dataset emphasizes the challenges multi-object trackers face under real-world surveillance conditions, such as low-quality camera views, and small-scale objects, and the need for online tracking. By evaluating trackers on a dataset that closely resembles the target application, researchers can focus on developing methods that are more robust and adaptable to these specific challenges.

In summary, the choice of dataset plays a vital role in evaluating and understanding the performance of multi-object trackers, especially in surveillance scenarios. By using a dataset like SurvTrack that closely resembles the target application (e.g., action detection in surveillance scenarios), we can ensure that the research community is developing and evaluating trackers that are more likely to succeed in real-world action detection applications.

B. THE IMPORTANCE OF DETECTOR SETTINGS

Our experiments demonstrate that the settings of the object detector play a critical role in the performance of multi-object trackers. Adjusting the confidence threshold of the detector can significantly impact overall tracking accuracy, as evidenced by the different performance results obtained from varying confidence thresholds. It is essential to carefully tune the confidence threshold to strike a balance between detecting enough objects for accurate tracking and avoiding too many false positives. The results suggest that the optimal confidence threshold may differ for each tracker, emphasizing the need for a tailored approach when choosing detector settings for specific multi-object tracking methods.

C. THE IMPORTANCE OF INPUT RESOLUTION

The input resolution of images fed into the object detector and tracker also significantly affects tracking performance. As demonstrated in our experiments, trackers generally achieve better performance with higher resolutions, providing more detailed visual information for both detection and tracking. However, the sensitivity to image resolution may vary across different trackers and confidence thresholds. It is essential to consider the trade-offs between increased performance and computational cost because higher resolutions can also lead to slower processing. Identifying the optimal input resolution for each tracker while considering computational constraints is crucial for achieving the best possible results.

D. FUTURE WORK

Our analysis of the SurvTrack dataset from multiple aspects reveals that, when properly configured with suitable confidence thresholds and image resolutions, ReID-based models can outperform pure motion-based models. However, ReID-based models often suffer from slower inference speeds due to the additional ReID component. To address this issue in the future, one possible approach is to develop a methodology that extracts appearance models directly from the object detector without requiring pretraining of the ReID module. In this way, the tracker can maintain the speed advantages of pure motion-based methods while benefiting from the additional ReID component. Such an approach would allow for more accurate and efficient multi-object tracking solutions. Additionally, the development of adaptive methods to automatically adjust confidence thresholds and input resolutions according to the scene or object type could lead to more robust and versatile tracking systems capable of handling various challenging scenarios. Lastly, employing color information in addition to ReID embeddings may potentially enhance performance while adding insignificant computational expense.

VI. CONCLUSION

The findings of this study provide valuable guidance for researchers and practitioners in designing effective action detection pipelines, particularly for surveillance scenarios. By carefully selecting the most appropriate tracker and configuration from this study, the performance of action detection systems can be significantly improved. Furthermore, this research highlights a critical issue in the multiple object tracking (MOT) community, where methods are often evaluated in narrow domains such as pedestrian tracking, while surveillance scenarios require more comprehensive evaluation. It is crucial to evaluate state-of-the-art methods in surveillance settings to ensure accurate performance assessment.

In addition, the importance of dataset selection, detector settings, and input resolution is emphasized in evaluating multi-object trackers for real-world surveillance applications. These factors significantly impact the performance of the trackers and should be carefully considered during the evaluation process. It is essential to choose datasets that are representative of surveillance scenarios and to optimize detector settings and input resolution for accurate tracking results.

By incorporating these insights and applying the recommendations provided in this study, researchers and practitioners can continue to advance the performance of online action detection systems in surveillance scenarios. The knowledge gained from this research will contribute to the development of more robust and reliable tracking methods, ultimately enhancing the effectiveness of surveillance systems.

To summarize:

- Selecting the appropriate tracker and configuration is crucial for designing an effective action detection pipeline, especially in surveillance scenarios. It is essential to evaluate trackers in a comprehensive range of domains, including surveillance settings, to ensure accurate performance assessment.
- Dataset selection, detector settings, and input resolution play significant roles in evaluating multi-object trackers for real-world surveillance applications. Careful consideration and optimization of these factors are necessary to achieve accurate and reliable tracking results.
- The findings from this research provide valuable insights and guidance for improving the performance of online action detection systems in surveillance scenarios. By incorporating these recommendations, researchers and practitioners can enhance the effectiveness of surveillance systems and contribute to the advancement of the field.

SUPPLEMENTARY MATERIALS

We provide a link to the failure cases of the VIRAT dataset for each of the five trackers evaluated in our study. The link to the failure cases is available in the supplementary material at the following URL: https://drive.google.com/drive/folders/11H5reEyvelh050xfAS7310gSAooQe6IR?usp=sharing Most of these cases involve small objects, occlusion, or illumination issues.

Additionally, we provide visualizations of the most successful cases for each tracker in the following link: https://drive.google.com/drive/folders/1kvkK3vtmUIzJbW5 × 2zKnljOy-do7yLHe?usp=sharing. It is worth noting that many of the successful cases involve static objects, where tracking is relatively easier.

Full video for qualitative analysis shown in Fig 10 can be found here https://drive.google.com/file/d/1bZamxpleRyq-zxrZmA9ovjkxoIntdsmo/view?usp=share_link

REFERENCES

[1] G. Ciaparrone, F. Luque Sánchez, S. Tabik, L. Troiano, R. Tagliaferri, and F. Herrera, "Deep learning in video multi-object tracking: A survey," *Neurocomputing*, vol. 381, pp. 61–88, Mar. 2020.

[2] J. Chen, J. Liu, J. Liang, T. Hu, W. Ke, W. Barrios, D. Huang, and A. G. Hauptmann, "Minding the gaps in a video action analysis pipeline," in *Proc. IEEE Winter Appl. Comput. Vis. Workshops (WACVW)*, Jan. 2019, pp. 41–46.

[3] I. Dave, Z. Scheffer, A. Kumar, S. Shiraz, Y. S. Rawat, and M. Shah, "GabriellaV2: Towards better generalization in surveillance videos for action detection," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. Workshops (WACVW)*, Jan. 2022, pp. 122–132.

[4] L. Yu, Y. Qian, W. Liu, and A. G. Hauptmann, "Argus++: Robust real-time activity detection for unconstrained video streams with overlapping cube proposals," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. Workshops (WACVW)*, Jan. 2022, pp. 112–121.

[5] K. Corona, K. Osterdahl, R. Collins, and A. Hoogs, "MEVA: A large-scale multiview, multimodal video dataset for activity detection," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 1059–1067.

[6] S. K. Challa, A. Kumar, and V. B. Semwal, "A multibranch CNN-BiLSTM model for human activity recognition using wearable sensor data," *Vis. Comput.*, vol. 38, no. 12, pp. 4095–4109, Dec. 2022.

[7] I. U. Khan, S. Afzal, and J. W. Lee, "Human activity recognition via hybrid deep learning based model," *Sensors*, vol. 22, no. 1, p. 323, Jan. 2022.

[8] X. Shu, J. Yang, R. Yan, and Y. Song, "Expansion-squeeze-excitation fusion network for elderly activity recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 8, pp. 5281–5292, Aug. 2022.

[9] S. Juraev, A. Ghimire, J. Alikhanov, V. Kakani, and H. Kim, "Exploring human pose estimation and the usage of synthetic data for elderly fall detection in real-world surveillance," *IEEE Access*, vol. 10, pp. 94249–94261, 2022.

[10] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger, and B. Leibe, "MOTS: Multi-object tracking and segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7934–7943.

[11] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "BDD100K: A diverse driving dataset for heterogeneous multitask learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2633–2642.

[12] Y. Cui, Z. Cao, Y. Xie, X. Jiang, F. Tao, Y. V. Chen, L. Li, and D. Liu, "DG-labeler and DGL-MOTS dataset: Boost the autonomous driving perception," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 3411–3420.

[13] P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler, and L. Leal-Taixé, "MOT20: A benchmark for multi object tracking in crowded scenes," 2020, *arXiv:2003.09003*.

[14] M. Broström. (2022). *YOLOv8 Tracking*. GitHub. Accessed: Mar. 31, 2023. [Online]. Available: https://github.com/mikel-brostrom/YOLOv8_tracking

[15] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," 2016, *arXiv:1603.00831*.

[16] Y. Wu, J. Lim, and M. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2411–2418.

[17] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, and T.-K. Kim, "Multiple object tracking: A literature review," *Artif. Intell.*, vol. 293, Apr. 2021, Art. no. 103448.

[18] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, p. 13, 2006.

[19] N. Aharon, R. Orfaig, and B.-Z. Bobrovsky, "BoT-SORT: Robust associations multi-pedestrian tracking," 2022, *arXiv:2206.14651*.

[20] Y. Du, Z. Zhao, Y. Song, Y. Zhao, F. Su, T. Gong, and H. Meng, "StrongSORT: Make DeepSORT great again," *IEEE Trans. Multimedia*, early access, Jan. 31, 2023, doi: 10.1109/TMM.2023.3240881.

[21] G. Maggiolino, A. Ahmad, J. Cao, and K. Kitani, "Deep OC-SORT: Multi-pedestrian tracking by adaptive re-identification," 2023, *arXiv:2302.11813*.

[22] J. Cao, J. Pang, X. Weng, R. Khirodkar, and K. Kitani, "Observation-centric SORT: Rethinking SORT for robust multi-object tracking," 2022, *arXiv:2203.14360*.

[23] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang, "ByteTrack: Multi-object tracking by associating every detection box," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 1–21.

[24] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *Computer Vision—ECCV*. Amsterdam, The Netherlands: Springer, 2016, pp. 17–35.

[25] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *EURASIP J. Image Video Process.*, vol. 2008, pp. 1–10, Jan. 2008.

[26] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C. Chen, J. T. Lee, S. Mukherjee, J. K. Aggarwal, H. Lee, and L. Davis, "A large-scale benchmark dataset for event recognition in surveillance video," in *Proc. CVPR*, Jun. 2011, pp. 3153–3160.

[27] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks for action recognition in videos," 2017, *arXiv:1705.02953*.

[28] S. Shao, Z. Zhao, B. Li, T. Xiao, G. Yu, X. Zhang, and J. Sun, "CrowdHuman: A benchmark for detecting human in a crowd," 2018, *arXiv:1805.00123*.

[29] S. Zhang, R. Benenson, and B. Schiele, "CityPersons: A diverse dataset for pedestrian detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4457–4465.

[30] L. Leal-Taixé, A. Milan, K. Schindler, D. Cremers, I. Reid, and S. Roth, "Tracking the trackers: An analysis of the state of the art in multiple object tracking," 2017, *arXiv:1704.02781*.

[31] N. Sarafianos, X. Xu, A. Mahmood, G. Pavlakos, D. Tzionas, and M. J. Black, "DanceTrack: An evaluation set for the study of human motion tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 475–491.

JUMABEK ALIKHANOV (Student Member, IEEE) received the B.S. degree from the Tashkent University of Information Technology, in 2014, and the M.E. degree from Inha University, in 2017, where he is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering. His research interests include machine learning and its applications to computer vision, sensor data science, and natural language processing.

HAKIL KIM (Member, IEEE) received the M.Sc. and Ph.D. degrees in electrical and computer engineering from Purdue University, in 1985 and 1990, respectively. In 1990, he joined the College of Engineering, Inha University, Incheon, South Korea, where he is currently a Full Professor with the Department of Information and Communication Engineering. In order to retain the balance between academic research and commercial development, he founded Vision Inc., in 2014, where he is also the CEO of the company. His research interests include biometrics, intelligent video surveillance, and embedded vision for autonomous vehicles. Since 2003, he has been actively involved as a Project Editor of the International Standardization of Biometrics at ISO/IEC JTC1/SC37.