http://doi.org/10.35784/iapgos.7691

received: 21.01.2026 | revised: 08.03.2026 | accepted: 16.03.2026 | available online: 30.03.2026

COMPARATIVE ANALYSIS OF DEEPSORT, BYTETRACK AND STRONGSORT ALGORITHMS FOR MULTI-OBJECT TRACKING IN UAV-BASED VIDEO SURVEILLANCE

Andrii Safonyk, Viktor Podvyshennyi, Oleksandr Naumchuk
National University of Water and Environmental Engineering, Department of Automation, Computer-Integrated Technologies and Robotics, Rivne, Ukraine

Abstract. *This paper presents a comparative analysis of state-of-the-art multi-object tracking algorithms applied in UAV-based video surveillance systems. The performance results of three advanced tracking methods – DeepSORT, ByteTrack, and StrongSORT – integrated with the YOLOv8 object detector are presented. A mathematical description and experimental simulations were conducted to evaluate the accuracy, stability, and computational performance of the algorithms in dynamic and complex scenes. The obtained results indicate that the StrongSORT + YOLOv8 combination provides the best balance between accuracy and robustness, whereas the ByteTrack method demonstrates high track continuity in high-density environments. The proposed approach can be utilized to enhance the efficiency of UAV-based autonomous monitoring systems.*

Keywords: object tracking, computer vision, unmanned aerial vehicle, object detector, tracking methods

ANALIZA PORÓWNAWCZA ALGORYTMÓW DEEPSORT, BYTETRACK I STRONGSORT W ZAKRESIE ŚLEDZENIA WIELU OBIEKTÓW W SYSTEMACH MONITORINGU WIZYJNEGO OPARTYCH NA BEZZAŁOGOWYCH STATKACH POWIETRZNYCH

Streszczenie. *W tym artykule przedstawiono analizę porównawczą najnowocześniejszych algorytmów śledzenia wielu obiektów stosowanych w systemach monitoringu wizyjnego opartych na bezzałogowych statkach powietrznych (UAV). Przedstawiono wyniki działania trzech zaawansowanych metod śledzenia – DeepSORT, ByteTrack i StrongSORT – zintegrowanych z detektorem obiektów YOLOv8. Przeprowadzono opis matematyczny i symulacje eksperymentalne w celu oceny dokładności, stabilności i wydajności obliczeniowej algorytmów w dynamicznych i złożonych scenach. Uzyskane wyniki wskazują, że połączenie StrongSORT + YOLOv8 zapewnia najlepszą równowagę między dokładnością a odpornością, podczas gdy metoda ByteTrack wykazuje wysoką ciągłość śledzenia w środowiskach o dużej gęstości. Proponowane podejście może być wykorzystane do zwiększenia wydajności autonomicznych systemów monitorowania opartych na bezzałogowych statkach powietrznych.*

Słowa kluczowe: śledzenie obiektów, wizja komputerowa, bezzałogowy statek powietrzny, detektor obiektów, metody śledzenia

Introduction

Unmanned Aerial Vehicles (UAVs) are finding increasingly broad applications in addressing diverse tasks, ranging from traffic monitoring and search-and-rescue operations to the security of strategic facilities and state borders. The effectiveness of such systems is largely determined by their ability not only to detect moving objects but also to ensure their continuous and accurate tracking under dynamic conditions.

Classical tracking algorithms, such as the Kalman filter, optical flow methods, and traditional detection association approaches, have long been utilized in computer vision systems. However, their performance notably deteriorates in scenarios involving a high number of objects, occlusions, or abrupt changes in movement trajectories. The increasing complexity of visual data obtained from UAV cameras creates a need for the development of more robust and adaptive tracking approaches.

Modern advancements in the fields of computer vision and deep learning have opened new possibilities for enhancing tracking accuracy and reliability. Current approaches combine object detectors, specifically YOLOv8, with advanced association and re-identification algorithms such as DeepSORT, ByteTrack, and StrongSORT. This integration enables multi-object tracking in real-time even within complex aerial scenes. Advanced models incorporate object appearance features and detection confidence scores, while utilizing prediction mechanisms to maintain stable tracks during partial occlusions or temporary object loss.

Despite the progress achieved, ensuring an optimal balance between tracking accuracy, robustness, and computational speed remains a pertinent scientific challenge, particularly for UAV-based systems with limited computational resources. Therefore, this paper conducts a comparative analysis of three modern tracking algorithms – DeepSORT, ByteTrack, and StrongSORT – integrated with the YOLOv8 detector. The objective of this study is to evaluate their efficiency under various dynamic conditions and to determine optimal configurations for application in UAV-based video surveillance systems.

1. Literature review

In recent years, the evolution of Multi-Object Tracking (MOT) has progressed from simple geometric association models to complex hybrid architectures that integrate object detection, appearance features, and trajectory prediction. The shift towards deep learning-based approaches has significantly enhanced tracking robustness and reduced identification errors within dynamic environments.

In parallel with tracking algorithm development, advances in object detection architectures – particularly YOLO-based models – have played a crucial role in improving end-to-end tracking performance. High-speed detectors enable real-time tracking when integrated with modern association algorithms, which is especially important for UAV-based applications [1].

One of the key milestones in the development of MOT was the introduction of the DeepSORT algorithm, which marked a transition from traditional tracking-by-detection methods to approaches utilizing deep appearance feature descriptors. This advancement enabled the effective distinction of similar objects and the preservation of their identities even under conditions of partial occlusion [3]. Subsequent research focused on improving feature extraction mechanisms, employing attention-based models, and optimizing re-identification pipelines.

Further progress was achieved with the development of StrongSORT, which enhances the original DeepSORT framework by incorporating more powerful re-identification models, trajectory filtering, and adaptive motion prediction. The algorithm demonstrated superior performance on public MOT benchmarks, particularly in complex scenes involving occlusions and rapid object motion [4].

A significant conceptual shift was introduced by the ByteTrack algorithm, which proposed associating both high- and low-confidence detections during the data association stage. This "full association" strategy significantly reduced track fragmentation and improved tracking stability in dense or noisy environments. As a result, ByteTrack became the basis for many subsequent tracking frameworks seeking a balance between accuracy and computational efficiency [5].

This work is licensed under a Creative Commons Attribution 4.0 International License.
Utwór dostępny jest na licencji Creative Commons Uznanie autorstwa 4.0 Międzynarodowe.

Contemporary research increasingly focuses on deploying multi-object tracking systems on embedded and edge platforms, including Raspberry Pi and NVIDIA Jetson devices. Techniques such as model pruning, weight quantization, and hardware-specific acceleration are widely used to reduce computational latency while maintaining acceptable accuracy [2].

Thus, the current state of research confirms that hybrid tracking architectures combining YOLO-based detectors with advanced association and re-identification modules represent one of the most promising directions for reliable UAV-based object tracking systems.

2. Problem statement

This study addresses the following primary tasks: a) reducing the number of identity switches (ID switches) and track fragmentations during partial object disappearance from the field of view or occlusions; b) enhancing tracking accuracy under conditions of unstable motion and imaging noise; c) optimizing computational performance to ensure real-time operation on embedded UAV platforms.

To achieve these objectives, three tracking algorithms utilizing deep association – DeepSORT, ByteTrack, and StrongSORT – were employed and integrated with the YOLOv8 detector. Each of these algorithms combines an object motion model (based on the Kalman filter) with visual object identification relying on appearance features, thereby ensuring track stability and reducing association errors.

The motion of an object is described by the Kalman filter equations:

$$\begin{gathered}\hat{x}_{k|k-1} = F\hat{x}_{k-1|k-1} + B_{uk}\\ P_{k|k-1} = FP_{k-1|k-1}F^T + Q\\ K_k = P_{k|k-1}H^T(HP_{k|k-1}H^T + R)^{-1}\\ \hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\big(z_k - H\hat{x}_{k|k-1}\big)\\ P_{k|k} = (I - K_kH)P_{k|k-1}\end{gathered} \tag{1}$$

where $\hat{x}_k$ – estimated state vector (position, velocity, size of bounding box), F – state transition matrix, H – observation matrix, P – covariance matrix, Q, R – process and measurement noise covariance matrices, K_k – Kalman gain, z_k – observation (detected position from YOLOv8).

For the data association between detections and predicted tracks, a cost matrix is constructed using both motion and appearance metrics. The association cost $C(i, j)$ between the i-th track and j-th detection is defined as:

$$C(i,j) = \alpha d_{IOU}(i,j) + (1-\alpha)d_{app}(i,j) \tag{2}$$

where d_{IOU} – distance based on Intersection over Union, d_{app} – cosine distance between appearance embeddings, $\alpha \in [0,1]$ – weighting coefficient determining the contribution of motion and appearance similarity.

Thus, this study formulates the problem of determining the optimal combination of DeepSORT, ByteTrack, and StrongSORT algorithms integrated with YOLOv8 that ensures the best balance between tracking accuracy, robustness, and real-time performance for UAV-based video surveillance systems.

3. Materials and methods

The developed experimental system integrates the YOLOv8 object detector with three modern multi-object tracking algorithms: DeepSORT, ByteTrack, and StrongSORT. The general system workflow is presented below:

1. Object detection – each video frame captured by the UAV camera is processed using the YOLOv8 network to generate bounding boxes (x, y, w, h) and confidence scores p_c.
2. Feature extraction – appearance embeddings $f_i \in \mathbb{R}^n$ are generated for each detection using a CNN-based re-identification model.
3. Motion prediction – for each active track, a Kalman filter predicts the expected state in the next frame.
4. Data association – detections and predictions are matched using a cost function that combines spatial overlap (IoU) and visual similarity.
5. Track update and management – confirmed tracks are updated, while unmatched detections initialize new tracks or are marked as lost after several frames.

Formally, each object track T_i is represented as a tuple:

$$T_i = \{x_i^k, P_i^k, f_{i,}s_i\}, \quad k = 1,2,\ldots,N \tag{3}$$

where x_i^k – estimated state vector (position and velocity), P_i^k – covariance matrix of uncertainty, f_i – appearance descriptor, s_i – current tracking state (active, lost, or deleted).

The process of multi-object tracking can be formulated as a sequential state estimation problem:

$$x_t = Fx_{t-1} + w_t, z_t = Hx_t + v_t \tag{4}$$

where $w_t \sim N(0,Q)$ and $v_t \sim N(0,R)$ represent process and measurement noise.

The goal of the tracking module is to find the optimal set of trajectories:

$$\hat{X} = arg\max_X P(X|Z) \tag{5}$$

where $Z = \{z_1, z_2, \ldots, z_T\}$ is the sequence of detections, and X represents the latent object states over time.

Data association determines which detections belong to which existing tracks. In all three algorithms, this process is modelled as a minimum-cost assignment problem based on a combination of geometric and appearance cues.

The association cost function is defined as:

$$C_{ij} = \alpha(1 - IoU_{ij}) + (1-\alpha)di_{j_{cos}} \tag{6}$$

where IoU_{ij} – intersection-over-union between the predicted bounding box of track i and detection j; d_{cos} – cosine distance between the feature embeddings; α – weighting coefficient balancing geometry and appearance (typically α = 0.7).

The optimal assignment matrix $A = [a_{ij}]$ is obtained using the Hungarian algorithm, minimizing the global association cost:

$$\min_A \sum_{i,j} C_{ij}a_{ij},\ a_{ij} \in \{0,1\} \tag{7}$$

subject to the constraints that each detection and each track can be assigned at most once.

The DeepSORT (Deep Simple Online and Realtime Tracking) algorithm extends the original SORT framework by integrating appearance-based features extracted using a convolutional neural network (CNN). This enhancement significantly improves robustness in multi-object tracking, particularly under conditions of occlusion or partial visibility. The tracking process in DeepSORT consists of four main stages:

1. Motion prediction using a Kalman filter.
2. Feature extraction to compute appearance embeddings.
3. Data association via a two-stage matching process.
4. Track management to initialize, confirm, and delete tracks.

Each object is represented by a state vector

$$x = [u, v, \gamma, h, u^{\cdot}, v^{\cdot}, \gamma^{\cdot}, h^{\cdot}]^T \tag{8}$$

where u, v denote the centre coordinates of the bounding box, γ – aspect ratio, h – height, and the dotted terms correspond to their velocities.

The Kalman filter predicts the next state and uncertainty as:

$$\hat{x}_{k|k-1} = F\hat{x}_{k-1|k-1}, P_{k|k-1} = FP_{k-1|k-1}F^T + Q \tag{9}$$

where F is the state transition matrix and Q is the process noise covariance. The predicted bounding box is then used for spatial matching with new detections.

For each detected object, a feature vector $f_i \in R^n$ is extracted using a pre-trained CNN trained on person or object re-identification datasets. This vector encodes the appearance signature of the object and remains relatively stable over time, allowing the tracker to distinguish between visually similar targets.

The cosine distance between two embeddings is computed as:

$$d_{cos}(f_i, f_j) = 1 - \frac{f_i \cdot f_j}{\|f_i\|\|f_j\|} \quad (10)$$

where $d[0,2]_{cos}$ measures visual dissimilarity between the previous and current detections.

DeepSORT uses a two-stage matching cascade to associate detections with predicted tracks:

1. First stage (appearance-based): Matches are prioritized for confirmed tracks using both motion and appearance metrics.
2. Second stage (IoU-based): Remaining unmatched detections are assigned to unconfirmed or newly initiated tracks based on Intersection over Union (IoU) distance.

The total cost function for each pair (i, j) of track and detection is defined as:

$$C(i,j) = \lambda d_{maha}(i,j) + (1-\lambda)d_{cos} \quad (11)$$

where d_{maha} is the Mahalanobis distance between the predicted state and detected position:

$$d_{maha}(i,j) = \sqrt{(z_i - Hx_i)^T S^{-1}(z_j - Hx_i)} \quad (12)$$

and $S = HP_iH^T + R$ is the innovation covariance matrix. The Hungarian algorithm is then applied to minimize the global cost across all detections and tracks.

Each track passes through three life-cycle states:

a) Tentative: A new detection is considered as a potential track and requires N_{init} consecutive matches to be confirmed.

b) Confirmed: Actively tracked and updated at each frame.

c) Deleted: Removed if unmatched for more than N_{max} frames.

This logic ensures that temporary false detections do not create persistent tracks and that lost objects are properly terminated. This combination of motion and appearance modelling enables DeepSORT to achieve reliable tracking performance in UAV-based video surveillance systems, even under dynamic conditions.

The ByteTrack algorithm represents a robust and efficient approach to multi-object tracking (MOT) that directly associates both high-confidence and low-confidence detections instead of discarding the latter, as done in most traditional trackers. By retaining these additional detections and applying a two-stage association strategy, ByteTrack significantly improves track continuity and reduces identity switches (IDSW), particularly in complex UAV-based surveillance scenes with partial occlusions.

Traditional tracking-by-detection frameworks rely on filtering out low-confidence detections to reduce false positives. However, this practice also removes valid but uncertain detections, which leads to track fragmentation. ByteTrack redefines this paradigm by processing all detections in two separate sets:

$$D = D_{high} \cup D_{low}, D_{high} \cap D_{low} = \emptyset \quad (13)$$

where D_{high} and D_{low} denote detections above and below a confidence threshold τ, respectively.

ByteTrack performs data association in two main stages using the IoU distance metric.

Stage 1: High-confidence association**.** Predicted tracks are matched with high-confidence detections D_{high}using IoU-based distance:

$$d_{IoU}(b_i, b_j) = 1 - \frac{|b_i \cap b_j|}{|b_i \cup b_j|} \quad (14)$$

A track is updated when the best match satisfies $d_{IoU}(b_i, b_j) < \delta_1$.

Stage 2: Low-confidence association. Unmatched tracks from Stage 1 are re-evaluated using detections from D_{low}. This secondary association recovers partially occluded or momentarily uncertain targets that were missed in the first stage. A lower threshold $\delta_2 > \delta_1$ is used to control association sensitivity. The Hungarian algorithm optimally solves both assignment problems.

As in DeepSORT, a Kalman filter is employed to predict the state of each track:

$$\hat{x}_{k|k-1} = F\hat{x}_{k-1|k-1}, P_{k|k-1} = FP_{k-1|k-1}F^T + Q \quad (15)$$

The prediction smooths the trajectory and provides motion continuity between frames. Unlike DeepSORT, ByteTrack does not require visual features, making it computationally lighter and more suitable for real-time UAV applications.

Each track T_i is categorized as:

a) Active: successfully matched with a detection;

b) Lost: temporarily unmatched, but still stored for potential reassociation;

c) Removed: deleted if unmatched for more than T_{lost} frames.

The track update rule can be expressed as:

$$T_i^{(k)} = \begin{cases} update(T_i^{(k-1)}, b_j) \\ predict(T_i^{(k-1)}) \\ remove(T_i) \end{cases} \quad (16)$$

This structure ensures stable and continuous tracking, even when detections fluctuate in confidence. ByteTrack achieves a balance between accuracy and real-time performance**,** as it avoids heavy CNN-based feature extraction. Its efficiency is particularly beneficial for edge devices and embedded UAV systems with limited computing power. Moreover, the integration of low-confidence detections increases MOTA and reduces ID switches, as confirmed in multiple benchmark datasets. This structure enables ByteTrack to maintain high tracking stability without the need for appearance embeddings, making it an ideal choice for lightweight aerial video analytics.

The StrongSORT algorithm is an enhanced and more robust variant of DeepSORT, designed to overcome its main limitations such as track fragmentation, identity switches, and instability in dynamic camera scenes. It integrates camera motion compensation (CMC), appearance enhancement, and re-identification refinement into the traditional tracking-by-detection framework.

The architecture of StrongSORT can be summarized in five functional components: a) motion prediction and compensation; b) appearance feature extraction and refinement; c) re-identification enhancement via EMA smoothing; d) data association using fused metrics; e) track state management and smoothing.

Unlike DeepSORT, which assumes a static camera model, StrongSORT explicitly estimates and compensates for camera motion between consecutive frames. This correction reduces artificial displacement of tracked objects caused by UAV movement or vibration. The global motion between frame $t-1$ and t is modeled by an affine transformation matrix M_t:

$$M_t = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \quad (17)$$

where a_{ij}represent rotation and scaling, and t_x, t_y represent translation.

Each predicted bounding box $b_{i,t-1}$ is updated as:

$$\tilde{b}_{i,t} = M_t b_{i,t-1} \quad (18)$$

This transformation stabilizes predicted positions before applying the Kalman filter, thus maintaining tracking accuracy in aerial or moving camera scenarios. StrongSORT employs a stronger ReID model (e.g., OSNet or ResNet-IBN) to extract higher-quality appearance embeddings f_i. To further improve feature stability, an Exponential Moving Average (EMA) update is used:

$$f_i^{(t)} = \alpha f_i^{(t-1)} + (1-\alpha)\hat{f}_i^{(t)} \quad (19)$$

where $\alpha \in [0,1]$ is the smoothing factor and $\hat{f}_i^{(t)}$is the current embedding from the CNN.

This smoothing reduces noise from illumination or pose changes. The cosine similarity remains the basis for appearance matching:

$$d_{cos}(f_i, f_j) = 1 - \frac{f_i \cdot f_j}{\|f_i\|\|f_j\|} \quad (20)$$

As in DeepSORT, StrongSORT uses a combined cost function integrating motion and appearance cues, but with additional weighting from CMC-corrected predictions:

$$C(i,j) = \lambda_1 d_{maha}^{CMC}(i,j) + \lambda_2(i,j) \quad (21)$$

where d_{maha}^{CMC} is the Mahalanobis distance computed using motion-compensated predictions. The Hungarian algorithm again provides the optimal matching between detections and tracks. After data association, the algorithm performs temporal smoothing of bounding box trajectories using:

$$\tilde{x}_t = \beta x_1 + (1-\beta)\tilde{x}_{t-1} \quad (22)$$

where β controls the inertia of the smoothed trajectory. This step minimizes abrupt position jumps and produces visually stable tracks even in turbulent UAV motion. Track management logic remains similar to DeepSORT, with states: *tentative*, *confirmed*, and *deleted*, but StrongSORT maintains an additional buffer of "lost" tracks that can be recovered if re-identified in subsequent frames.

StrongSORT achieves higher robustness compared to its predecessors due to: compensation of global motion (CMC); smoothed feature embeddings via EMA; temporal smoothing of trajectories; improved ReID networks and adaptive association weighting. These improvements collectively enhance MOTA, reduce IDSW, and ensure higher stability under camera movement and occlusion conditions, making StrongSORT particularly effective for UAV-based object tracking.

The comparative analysis of the DeepSORT, ByteTrack, and StrongSORT algorithms was performed to evaluate their tracking performance and runtime efficiency under identical experimental conditions. The evaluation utilized a publicly available UAV video dataset with verified ground-truth annotations, eliminating the need for manual labelling and ensuring objective quantitative assessment.

The performance evaluation followed a standardized experimental protocol consisting of three main stages:

1. Dataset selection. UAV-based video sequences with diverse flight altitudes, illumination conditions, and object densities were selected from a ground-truth annotated dataset designed for multi-object tracking tasks. Each frame of the dataset contains manually verified bounding boxes and unique object identifiers, which were used as a baseline for comparison.
2. Algorithm execution. The three tracking algorithms – DeepSORT, ByteTrack, and StrongSORT – were independently applied to the same video sequences. Each method processed identical detection outputs from the YOLOv8 model, ensuring consistency in the experimental setup. The resulting sets of tracks are denoted as $T = \{t_1, t_2, \ldots, t_n\}$.
3. Metric computation. Quantitative comparison was carried out using standard multi-object tracking metrics – MOTA (Multiple Object Tracking Accuracy), IDSW (Identity Switches), and FPS (Frames Per Second) – according to the MOTChallenge evaluation framework.

This methodology ensures reproducibility, objectivity, and comparability of results across different tracking paradigms. The overall tracking accuracy is measured by the MOTA metric:

$$MOTA = 1 - \frac{\sum_t (FN_t + FP_t + IDSW_t)}{\sum_t GT_t} \quad (23)$$

where FN_t, FP_t and $IDSW_t$ denote the number of false negatives, false positives, and identity switches at time *t*, respectively, and GT_t represents the total number of ground truth objects. Higher MOTA values correspond to better performance. The number of identity switches reflects the stability of object identification:

$$IDSW = \sum_{i=1}^{N} \delta(ID_{i,t} \neq ID_{i,t-1}) \quad (24)$$

where $\delta(\cdot)$ is an indicator function that equals 1 when an identity change occurs.

A smaller number of IDSW indicates higher temporal consistency. The FPS metric quantifies computational efficiency:

$$FPS = \frac{N_f}{t_{proc}} \quad (25)$$

where N_f is the total number of processed frames and t_{proc} is the overall processing time in seconds. It is critical for assessing suitability for real-time UAV applications.

4. Results and discussion

The objective of the conducted experimental research was to evaluate the performance of the DeepSORT, ByteTrack, and StrongSORT algorithms, integrated with the YOLOv8 detector, for multi-object tracking tasks on video data. The evaluation was carried out using a dataset with ground truth annotations based on three primary metrics: MOTA, ID Switches (IDSW), and Frames Per Second (FPS). The research results are presented in Table 1.

Table 1. Comparative evaluation of tracking algorithms

Algorithm	MOTA (%)	ID Switches	FPS
DeepSORT + YOLOv8	73–80	400–700	25–30
ByteTrack + YOLOv8	80–84	350–600	35–40
StrongSORT + YOLOv8	84–88	200–450	28–32

The obtained results demonstrate a clear trade-off between tracking accuracy and computational speed. Notably, ByteTrack achieved the highest FPS due to its simplified two-stage association mechanism; however, it exhibited an increased number of identity switches (IDSW) during partial occlusion scenarios. In contrast, DeepSORT demonstrated more stable identification through the utilization of visual features (ReID), although its accuracy degraded during active camera motion. The superior result was provided by StrongSORT, which incorporates Camera Motion Compensation (CMC) and feature smoothing (EMA). This integration resulted in a reduction of IDSW by approximately 35% and an increase in MOTA by 4–5% compared to ByteTrack. Consequently, StrongSORT achieves an optimal balance between accuracy and stability in dynamic UAV scenarios, whereas ByteTrack remains the preferred choice for real-time systems, and DeepSORT is suitable for environments with stationary or slow-moving cameras where identification stability is the priority.

To visually illustrate the relationship between these performance metrics, Figure 1 depicts the comparative efficiency of the analysed algorithms under identical conditions.

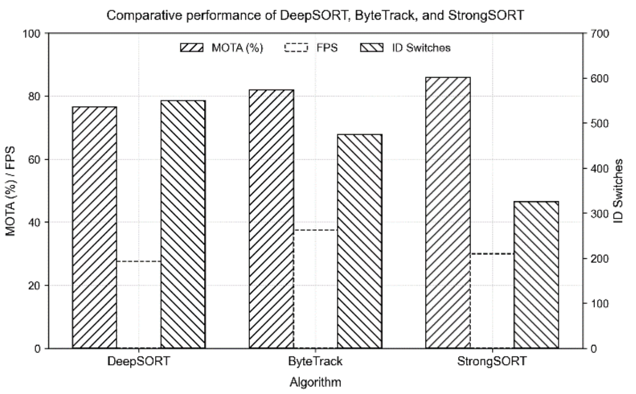

Fig. 1. Comparative performance of the DeepSORT, ByteTrack, and StrongSORT algorithms evaluated on the UAV ground-truth dataset using MOTA, IDSW, and FPS metrics

Thus, the conducted research indicates that the selection of an algorithm depends on the specific application domain:

- ByteTrack is recommended for real-time systems operating on resource-constrained devices;
- StrongSORT is suitable for analytical or high-precision surveillance systems;
- DeepSORT is appropriate for stationary or semi-static environments where accurate object identification is critical.

5. Conclusions

This study conducted a comparative analysis of three modern multi-object tracking algorithms – DeepSORT, ByteTrack, and StrongSORT – integrated with the YOLOv8 detector for the analysis of UAV video footage. Experimental results demonstrate that StrongSORT achieves the highest accuracy (MOTA = 84–88%) and track stability due to the application of Camera Motion Compensation (CMC) and feature smoothing (EMA). Conversely, ByteTrack provides the highest processing speed (FPS = 35–40), making it suitable for real-time embedded systems. DeepSORT exhibited balanced performance but proved less robust to dynamic scene changes.

The comparative analysis confirmed that hybrid approaches combining motion modelling and visual features ensure superior robustness and tracking continuity in complex aerial scenes. The research findings hold practical significance for aerial monitoring systems, territory security, and traffic flow analysis, where maintaining a balance between accuracy, stability, and computational speed is essential.

Future research should focus on:

- optimizing StrongSORT and ByteTrack for low-power platforms (such as Raspberry Pi and Jetson Nano) through quantization and model pruning;
- integrating Transformer-based models to enhance ReID quality during partial occlusions;
- developing adaptive tracking systems that dynamically adjust operating parameters based on flight conditions and computational resources.

Thus, the conducted research establishes a scientific and practical foundation for the selection and adaptation of object tracking algorithms in UAV-based computer vision systems, ensuring reliability, accuracy, and efficiency in real-time.

References

[1] Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). *YOLOv4: Optimal Speed and Accuracy of Object Detection* (arXiv:2004.10934). arXiv. https://doi.org/10.48550/arXiv.2004.10934

[2] McKinney, W. (2010). *Data Structures for Statistical Computing in Python*. SciPy 2010, 56–61. https://doi.org/10.25080/Majora-92bf1922-00a

[3] Nahrstedt, F., Karmouche, M., Bargieł, K., Banijamali, P., Nalini Pradeep Kumar, A., & Malavolta, I. (2024). An Empirical Study on the Energy Usage and Performance of Pandas and Polars Data Analysis Python Libraries. *Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering*, 58–68. https://doi.org/10.1145/3661167.3661203

[4] Wojke, N., Bewley, A., & Paulus, D. (2017). *Simple Online and Realtime Tracking with a Deep Association Metric* (arXiv:1703.07402). arXiv. https://doi.org/10.48550/arXiv.1703.07402

[5] Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., & Wang, X. (2022). *ByteTrack: Multi-Object Tracking by Associating Every Detection Box* (arXiv:2110.06864). arXiv. https://doi.org/10.48550/arXiv.2110.06864

Prof. Andrii Safonyk
e-mail: a.p.safonyk@nuwm.edu.ua

Director of the Institute of Energy, Automation and Water Engineering, National University of Water and Environmental Engineering, Rivne, Ukraine.
Engaged in scientific mathematical modelling of natural and technological processes, computer techniques and computer technologies, programming.

https://orcid.org/0000-0002-5020-9051

M.Sc. Podvyshennyi Viktor
e-mail: v.s.podvyshenyi@nuwm.edu.ua

Ph.D. student of the Department of Automation, Computer-Integrated Technologies and Robotics, National University of Water and Environmental Engineering, Rivne, Ukraine.
Research interests include computer vision and machine learning methods for UAV-based object detection and tracking, aerial image analysis, and the development of edge AI systems.

https://orcid.org/0009-0007-5037-7315

Ph.D. Oleksandr Naumchuk
e-mail: o.m.naumchuk@nuwm.edu.ua

Associate professor of the Department of Automation, Computer-Integrated Technologies and Robotics, of the of the Institute of Energy, Automation and Water Engineering, National University of Water and Environmental Engineering, Rivne, Ukraine.
Engaged in scientific research of effects of electro-magnetic radiation from the mobile communication equipment on the environment, and automation, mathematical modelling of technological processes.

https://orcid.org/0000-0003-2483-4141