Onboard SORT-Based Bird Pest Tracking from Drone Footage for Rice Farm Monitoring

Chisom S. Nwokonko

Department of Electrical and Electronic Engineering,
Imo State University, Owerri

Abstract

This paper presents an onboard Simple Online and Realtime Tracker (SORT)-based approach for bird pest tracking from drone footage in rice farm monitoring. The work addresses the need for a lightweight tracker that can support real-time drone response on resource-constrained platforms. The study used the Distant Bird Detection Dataset and the Roboflow Drone-Bird Dataset as drone-view bird image sources. The preprocessed data were divided into 80% for detector preparation and tracker configuration and 20% for validation. The onboard SORT tracker achieved MOTA of 78.20%, MOTP of 74.10%, drone-bird proximity error of 2.03 m, and Proximity Compliance Rate of 97.80%. Overall, the results show that SORT is suitable for low-latency onboard bird pest tracking, although future work should improve identity consistency under dense flock movement and temporary occlusion.

Keywords: bird pest repellent system; drone footage; Simple Online and Realtime Tracker; SORT; onboard AI; rice farm; multi-object tracking; computer vision.

1. Introduction

Rice is an important food crop, but its production is often affected by bird damage during the vulnerable heading and grain-filling stages. In many rice farms, farmers still depend on scarecrows, manual guarding, noise making, reflective materials, and other traditional deterrent methods. These methods may provide short-term relief, but they are difficult to sustain over large fields and can become less effective when birds become familiar with repeated stimuli.

Drone-based monitoring provides a more flexible option because the drone can observe large farm areas, follow bird movement, and support timely repellent action. A complete drone-based bird pest monitoring system normally requires three connected functions. The first function is detection, which localizes birds in the captured frame. The second function is classification, which helps to separate rice pest birds from harmless birds. The third function is tracking, which maintains the location and identity of the detected birds over time so that the drone can make a meaningful response.

Existing drone-based bird monitoring studies often focus on detection or classification, while the tracking stage is less clearly evaluated for onboard deployment. This creates a research gap because the drone must track birds in real time under limited processing power, limited battery capacity, camera movement, motion blur, and changing altitude. A heavy tracker may improve identity consistency, but it may not be suitable for immediate onboard drone response. A lightweight tracker is therefore needed for practical deployment.

SORT is suitable for this role because it uses Kalman filter prediction and Hungarian assignment without a deep appearance feature extractor. This makes it faster and simpler than appearance-aware trackers such as DeepSORT. However, the absence of deep appearance features can increase identity switches when birds overlap, move in flocks, or leave and re-enter the camera view. This paper evaluates onboard SORT tracking for real-time rice farm bird pest monitoring.

This paper makes three main contributions. First, it presents a lightweight onboard SORT-based tracking framework for drone-assisted bird pest monitoring in rice farms. Second, it formulates the tracking pipeline using Kalman prediction, IoU-based association, and Hungarian assignment for real-time bird trajectory estimation. Third, it evaluates the tracker using multi-object tracking and drone-bird proximity metrics to assess its suitability for low-latency repellent response.

2. Related Work

Bird pest monitoring has moved from static deterrent devices toward automated sensing, computer vision, and drone-assisted monitoring. Sensor-based systems are useful for simple detection and repellent triggering, but they often cannot distinguish rice pest birds from non-target species. Computer vision improves this process by allowing the system to detect birds from image or video frames and provide object-level information for further analysis.

Object detection models such as YOLO, SSD, Faster R-CNN, and EfficientDet have been used in many real-time agricultural monitoring tasks because they can localize objects in complex farm scenes. For drone-based bird monitoring, the detection model must handle small targets, background clutter, lighting changes, camera motion, and distance-related scale variation. The detector output then becomes the input to the tracker.

Multi-object tracking methods link detections across video frames. SORT is a classical real-time tracking-by-detection method. It applies a Kalman filter to predict the next location of each object and uses the Hungarian algorithm to match predicted tracks with current detections. DeepSORT extends SORT by adding appearance embeddings, while ByteTrack improves association by using both high-confidence and low-confidence detections. These methods can improve identity consistency, but they may introduce additional computation. For onboard drone deployment, SORT remains attractive because of its speed and low resource demand. Recent SORT variants such as OC-SORT further show that observation-centred updates can improve robustness under occlusion and non-linear motion while retaining online real-time tracking behaviour. Recent drone-specific MOT research such as DroneMOT also shows that moving-camera tracking requires special attention to small objects, blur, occlusion, drone motion, and changing viewing angles [20].

3. Materials and Methods

3.1. Dataset and Data Preparation

The study used the Distant Bird Detection Dataset and the Roboflow Drone-Bird Dataset as the main drone-based bird image sources. The datasets contain distant and aerial-view bird instances that are relevant to rice farm monitoring because the target birds are small, mobile, and often captured against complex farm backgrounds.

Each annotated bird instance was represented by a bounding box used for tracking-by-detection evaluation. The final validation records used for the multi-object tracking evaluation contained 1,011 ground-truth bird bounding boxes. The validation records used for the reported multi-object tracking evaluation contained 1,011 ground-truth bird bounding boxes. These annotated instances were used to compute missed detections, false positives, identity switches, matched pairs, and proximity-based tracking scores.

The preprocessed dataset was divided into two parts. Eighty percent was used for detector preparation and tracker configuration, while twenty percent was used for validation. This wording is important because SORT is not trained like a neural network. Instead, the detector provides bounding boxes, while SORT is configured and evaluated using tracking parameters such as IoU threshold, maximum track age, and minimum confirmation hits. The validation subset contained 1,011 ground-truth bird boxes used for the reported tracking evaluation.

Table 1. Dataset usage and experimental split

Item	Description
Primary data sources	Distant Bird Detection Dataset and Roboflow Drone-Bird Dataset

Item	Description
Application context	Drone-based bird pest monitoring in rice farms
Data representation	Annotated bird bounding boxes used for tracking-by-detection evaluation
Detector preparation and tracker configuration split	80% of the preprocessed dataset for detector preparation and tracker configuration
Validation split	20% of the preprocessed dataset for performance evaluation
Tracking output	Bird track IDs, smoothed bounding boxes, trajectories, and proximity estimates
Validation ground-truth boxes	1,011 annotated bird boxes used for MOT evaluation

3.2. System Model Architecture

The system model begins with drone video capture over the rice farm. Each frame passes through onboard preprocessing and bird detection. The SORT tracker receives bird bounding boxes from the onboard MobileNetV3-YOLO-Lite/SSD detection module before applying Kalman prediction and IoU-based association. The resulting track IDs, smoothed bounding boxes, and drone-bird proximity estimates are then used for drone response decisions and base-station logging. This arrangement supports fast onboard action while keeping the base station available for heavier review, analytics, and future model updates.

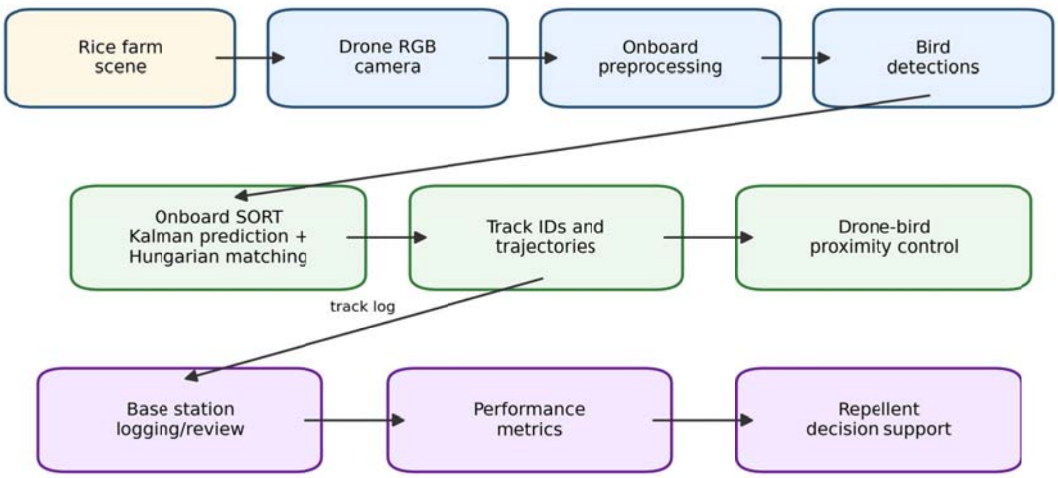

Figure 1. System model architecture for onboard SORT-based rice farm bird tracking.

3.3. Onboard SORT Model Architecture

The onboard SORT architecture has five main stages. The first stage receives frame-level bird detections from the onboard detection module. The second stage predicts the new locations of existing tracks using a Kalman filter. The third stage computes the Intersection over Union (IoU) cost between predicted boxes and detected boxes. The fourth stage uses the Hungarian algorithm to obtain the best assignment between detections and tracks. The final stage updates matched tracks, creates new tracks for unmatched detections, and deletes stale tracks that are not matched for a specified number of frames.

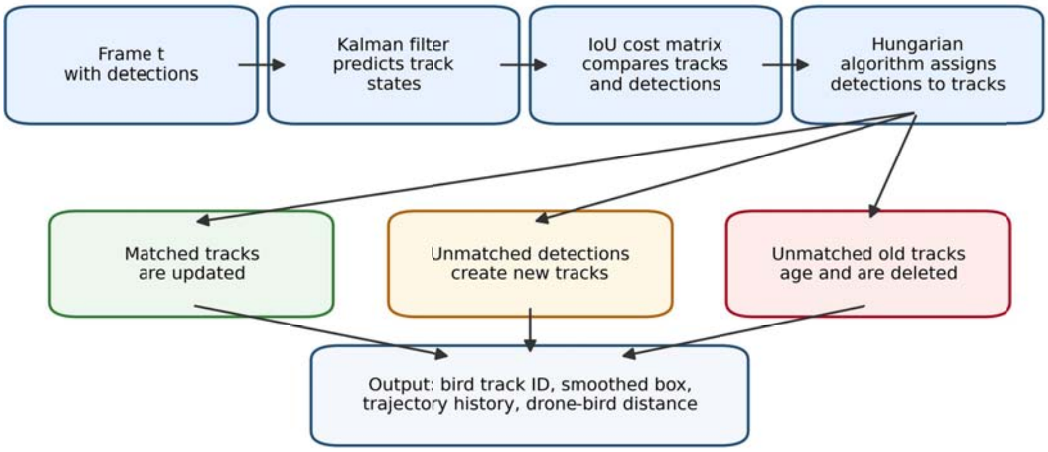

Figure 2. Model architecture of the onboard SORT tracker.

3.4. SORT State Representation and Motion Prediction

For each tracked bird, SORT maintains a state vector that represents the bounding box centre, scale, aspect ratio, and motion components. The state vector used in this paper is written as:

$$\mathbf{x}_k = [u_k,\ v_k,\ s_k,\ r_k,\ \dot{u}_k,\ \dot{v}_k,\ \dot{s}_k]^T \tag{1}$$

where u_k and v_k are the coordinates of the bounding box centre at frame k, s_k is the scale or area of the bounding box, r_k is the aspect ratio, and the dotted terms represent the corresponding velocity components. This representation allows the tracker to predict the next bounding box even when a bird is briefly missed by the detector due to occlusion, motion blur, or rapid movement. The Kalman filter prediction stage is expressed as:

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k \mathbf{x}_{k-1|k-1} + \mathbf{w}_k \tag{2}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q}_k \tag{3}$$

where $\mathbf{F}_k$ is the state transition matrix, $\mathbf{P}_{k|k-1}$ is the predicted state covariance matrix, $\mathbf{Q}_k$ is the process noise covariance matrix, and $\mathbf{w}_k$ is the process noise vector. When a new detection is received, the correction stage is expressed as:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \left(\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k \right)^{-1} \tag{4}$$

$$\mathbf{x}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}) \tag{5}$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} \tag{6}$$

where $\mathbf{H}_k$ maps the hidden state to the observed measurement, $\mathbf{R}_k$ is the measurement noise covariance matrix, $\mathbf{z}_k$ is the observed detection, and $\mathbf{K}_k$ is the Kalman gain. The Kalman gain determines how much the tracker should trust the predicted motion compared with the new detection.

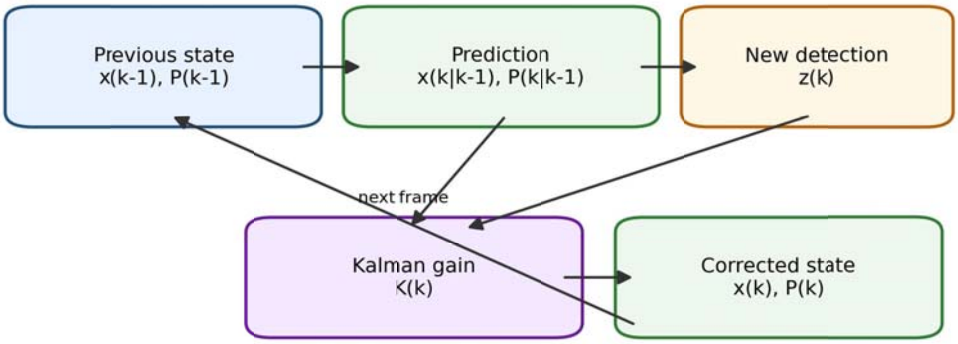

Figure 3. Kalman filter state update cycle used in SORT.

3.5. Data Association with IoU and the Hungarian Algorithm

After prediction, the tracker must decide which current detections belong to the predicted tracks. This is handled as an assignment problem. The IoU between a predicted bounding box and a detected bounding box is computed as:

$$\mathrm{IoU}\left(B_i^p, B_j^d\right) = \frac{\left|B_i^p \cap B_j^d\right|}{\left|B_i^p \cup B_j^d\right|} \qquad (7)$$

where B_i^p is the predicted bounding box of track i, B_j^d is the detected bounding box j, $\cap$ denotes the intersection area, and $\cup$ denotes the union area. The association cost is then computed as:

$$C_{ij} = 1 - \mathrm{IoU}\left(B_i^p, B_j^d\right) \qquad (8)$$

The Hungarian algorithm finds the assignment that minimizes the total association cost:

$$\mathbf{A} = \underset{\mathbf{A}}{\mathrm{argmin}} \sum_{(i,j)\in\mathbf{A}} C_{ij} \qquad (9)$$

A detection is treated as unmatched when its IoU with the predicted track is below the acceptance threshold:

$$\mathrm{IoU}\left(B_i^p, B_j^d\right) < \tau_{\mathrm{IoU}} \qquad (10)$$

where τ_{IoU} is the IoU acceptance threshold. Unmatched detections can create new tracks, while unmatched old tracks are retained for a short period and then deleted when they exceed the maximum track age.

3.6. Algorithmic Procedure

Algorithm 1. Onboard SORT tracking procedure

1. Input video frames, detector bounding boxes, IoU threshold, maximum track age, and minimum confirmation hits.
2. Initialize an empty list of active tracks.
3. For each new frame, obtain bird detections from the onboard detector.
4. Predict the next state of every active track using the Kalman filter.
5. Compute the IoU cost matrix between predicted tracks and current detections.
6. Apply the Hungarian algorithm to associate detections with predicted tracks.
7. Update matched tracks using the assigned detections.
8. Create new tracks from unmatched detections.
9. Increase the age of unmatched tracks and delete tracks older than the maximum age.

10. Output confirmed track IDs, smoothed bounding boxes, bird trajectories, and proximity estimates.

3.7. Evaluation Metrics

The tracker was evaluated with standard multi-object tracking metrics and drone proximity metrics. MOTA measures the overall tracking accuracy by considering false positives, false negatives, and identity switches. It is computed as:

$$\mathrm{MOTA} = \left(1 - \frac{\mathrm{FN} + \mathrm{FP} + \mathrm{IDSW}}{N_{\mathrm{GT}}}\right) \times 100\% \qquad (11)$$

where FN is the number of false negatives, FP is the number of false positives, IDSW is the number of identity switches, and N_{GT} is the total number of ground-truth objects. MOTP measures localization precision for matched objects. In this paper, it is expressed using average IoU over all matched pairs:

$$\mathrm{MOTP} = \frac{1}{N_m} \sum_{m=1}^{N_m} \mathrm{IoU}\left(B_m^p, B_m^g\right) \times 100\% \qquad (12)$$

where N_m is the number of matched pairs, B_m^p is a predicted bounding box, and B_m^g is the corresponding ground-truth bounding box. IDF1 reflects identity consistency across frames and is defined as:

$$\mathrm{IDF1} = \frac{2\mathrm{IDTP}}{2\mathrm{IDTP} + \mathrm{IDFP} + \mathrm{IDFN}} \times 100\% \qquad (13)$$

where IDTP, IDFP, and IDFN denote identity true positives, identity false positives, and identity false negatives, respectively. Fragmentation counts how often continuous ground-truth trajectories are broken. Drone-Bird Proximity Error (DBPE) measures the average drone-bird proximity error in metres and is computed as:

$$\mathrm{DBPE} = \frac{1}{N} \sum_{k=1}^{N} \left| d_k^{\mathrm{ref}} - d_k^{\mathrm{est}} \right| \qquad (14)$$

where d_k^{ref} is the reference drone-bird distance, d_k^{est} is the estimated drone-bird distance, and N is the number of evaluated proximity samples. The Proximity Compliance Rate (PCR) measures the percentage of tracking decisions that meet the desired proximity requirement:

$$\mathrm{PCR} = \frac{1}{N} \sum_{k=1}^{N} \mathbb{I}\left(d_{\min} \le d_k^{\mathrm{est}} \le d_{\max}\right) \times 100\% \qquad (15)$$

where $\mathbb{I}(\cdot)$ is the indicator function, $d_{\min}$ is the minimum acceptable drone-bird distance, and $d_{\max}$ is the maximum acceptable drone-bird distance.

Table 2. Tracking metrics used for evaluation

Metric	Meaning	Preferred direction
MOTA (%)	Overall tracking accuracy considering false positives, false negatives, and identity switches	Higher is better
MOTP (%)	Precision of bounding box localization for matched objects	Higher is better
IDF1 (%)	Consistency of identity assignment across frames	Higher is better
Fragmentation	Number of breaks in continuous trajectories	Lower is better
DBPE (m)	Drone-bird proximity error in metres	Lower is better

Metric	Meaning	Preferred direction
PCR (%)	Proximity compliance rate for drone-bird operating distance	Higher is better

4. Results

The results for the onboard embedded SORT tracker are presented in Table 3. The tracker achieved MOTA of 78.20% and MOTP of 74.10%. The drone-bird proximity error was 2.03 m, while the Proximity Compliance Rate was 97.80%. The validation record contained 1,011 ground-truth boxes, with 919 matched pairs, 92 false negatives, 68 false positives, and 65 identity switches. The 1,011 validation ground-truth boxes provided the basis for computing missed detections, false positives, identity switches, and matched tracking pairs. Although the IDF1 value is low, the proximity metrics show that SORT still provides useful position guidance for real-time drone response.

Table 3. Performance scores of the onboard SORT tracking model

Metric	Value
MOTA (%)	78.20
MOTP (%)	74.10
IDF1 (%)	5.20
Fragmentation (Frag)	88
Drone-Bird Proximity Error (DBPE, m)	2.03
Proximity Compliance Rate (PCR, %)	97.80
Total Ground-Truth Boxes	1,011
False Negatives (FN)	92
False Positives (FP)	68
ID Switches	65
Matched Pairs	919

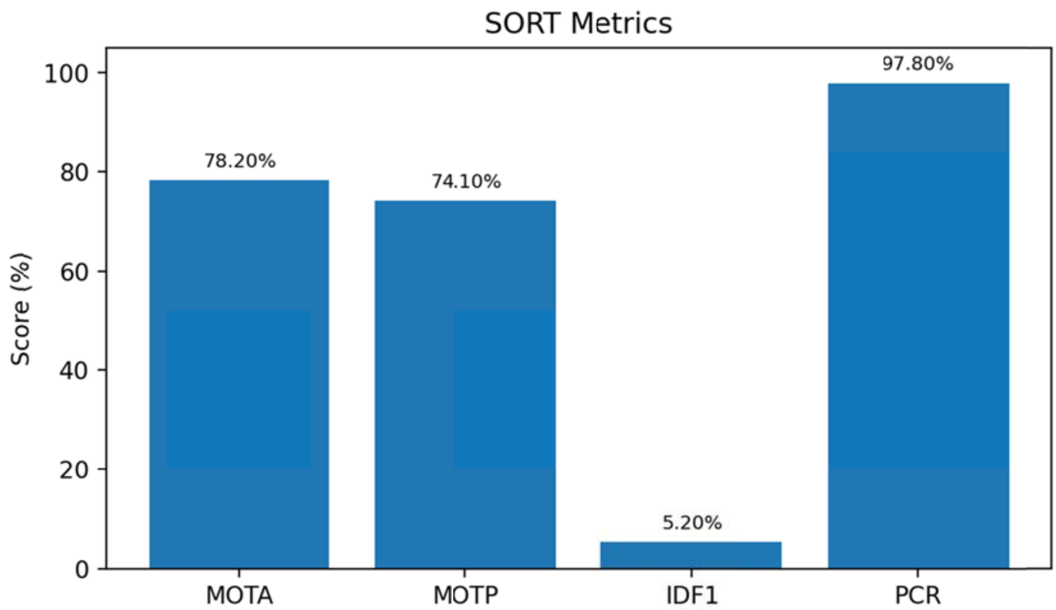

Figure 4. Comparative visualization of selected onboard SORT tracking and proximity metrics.

5. Discussion

The results show that SORT provided a useful balance between tracking performance and computational simplicity. A MOTA of 78.20% indicates that the tracker maintained acceptable overall tracking accuracy despite false negatives, false positives, and identity switches. A MOTP of 74.10% shows that the matched bounding boxes were localized with reasonable precision. These two values support the use of SORT as a practical onboard tracker for drone-based bird pest monitoring.

The proximity-related result is also important for the rice farm use case. A DBPE of 2.03 m indicates that the tracker can provide bird position information with a low distance error for drone response. The PCR of 97.80% shows that the tracker maintained the required drone-bird operating proximity in most evaluated cases. This is important because bird repellent action depends not only on detecting birds, but also on keeping the drone close enough to influence bird movement without unnecessary energy waste.

The main limitation is identity consistency. The IDF1 value is low and 65 identity switches were recorded. This means that SORT sometimes lost or changed bird identities, especially when birds moved close together or crossed paths. The low IDF1 value is expected because SORT does not use appearance embeddings, so it may reassign identities when birds overlap, cross paths, or temporarily leave the detector view. Although the IDF1 value is low, the high PCR and low DBPE show that the tracker still provides useful location guidance for drone-based bird repellent response, where proximity control is more critical than long-term identity preservation.

Despite the acceptable MOTA and proximity performance, the study is limited by the use of SORT without appearance-based re-identification. This affects identity preservation in dense flock movement and occlusion cases. The study also depends on the quality of the upstream detector, meaning that missed detections and false detections directly affect tracking performance.

In practical deployment, a dual-tier strategy is recommended. The drone can use SORT for fast onboard tracking and immediate repellent response, while the base station can apply more complex tracking or review methods after flight. This arrangement preserves real-time action on the drone and allows the base station to improve identity correction, long-term analytics, and model refinement.

6. Conclusion

This paper presented an onboard SORT-based bird pest tracking study for rice farm monitoring. The study clarifies that SORT is a configured tracking-by-detection algorithm rather than a neural network trained in the conventional sense. The paper also clarifies the dataset usage, system model, SORT architecture, Kalman filter prediction, IoU-based association, and evaluation metrics.

The experimental results show that the onboard SORT tracker achieved MOTA of 78.20%, MOTP of 74.10%, drone-bird proximity error of 2.03 m, and Proximity Compliance Rate of 97.80%. These values indicate that SORT is suitable for lightweight onboard tracking and immediate drone response in rice farm bird pest monitoring.

In practical use, the SORT tracker can serve as the first-level onboard tracker for immediate drone response, while more complex base-station models can support post-flight review, identity correction, and model improvement. In practical farm deployment, the tracker should be paired with a robust detector and tested under different altitudes, lighting conditions, wind levels, and flock densities.

Since SORT relies directly on detector outputs, future improvement should also consider detector robustness under small-object, low-light, and motion-blur conditions. Future work should further evaluate DeepSORT, ByteTrack, OC-SORT, and other appearance-aware or motion-aware trackers to improve identity consistency under dense flock movement and temporary occlusion.

References

[1] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, “Simple online and realtime tracking,” in *Proc. IEEE Int. Conf. Image Processing*, 2016, pp. 3464-3468. doi: 10.1109/ICIP.2016.7533003.

[2] N. Wojke, A. Bewley, and D. Paulus, “Simple online and realtime tracking with a deep association metric,” in *Proc. IEEE Int. Conf. Image Processing*, 2017, pp. 3645-3649. doi: 10.1109/ICIP.2017.8296962.

[3] Y. Zhang et al., “ByteTrack: Multi-object tracking by associating every detection box,” in *Proc. European Conf. Computer Vision*, 2022, pp. 1-21. doi: 10.1007/978-3-031-20047-2_1.

[4] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016, pp. 770-778. doi: 10.1109/CVPR.2016.90.

[5] W. Liu et al., “SSD: Single shot multibox detector,” in *Proc. European Conf. Computer Vision*, 2016, pp. 21-37. doi: 10.1007/978-3-319-46448-0_2.

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016, pp. 779-788. doi: 10.1109/CVPR.2016.91.

[7] M. Tan, R. Pang, and Q. V. Le, “EfficientDet: Scalable and efficient object detection,” in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, 2020, pp. 10781-10790. doi: 10.1109/CVPR42600.2020.01079.

[8] A. Howard et al., “Searching for MobileNetV3,” in *Proc. IEEE/CVF Int. Conf. Computer Vision*, 2019, pp. 1314-1324. doi: 10.1109/ICCV.2019.00140.

[9] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, “ShuffleNet V2: Practical guidelines for efficient CNN architecture design,” in *Proc. European Conf. Computer Vision*, 2018, pp. 116-131. doi: 10.1007/978-3-030-01264-9_8.

[10] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137-1149, 2017. doi: 10.1109/TPAMI.2016.2577031.

[11] M. Tan and Q. V. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” in *Proc. Int. Conf. Machine Learning*, 2019, pp. 6105-6114.

[12] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal speed and accuracy of object detection,” arXiv:2004.10934, 2020.

[13] H. Zhang and J. M. Kovacs, “The application of small unmanned aerial systems for precision agriculture: A review,” *Precision Agriculture*, vol. 13, pp. 693-712, 2012. doi: 10.1007/s11119-012-9274-5.

[14] U. R. Mogili and B. B. V. L. Deepak, “Review on application of drone systems in precision agriculture,” *Procedia Computer Science*, vol. 133, pp. 502-509, 2018. doi: 10.1016/j.procs.2018.07.063.

[15] E.-C. Oerke, “Crop losses to pests,” *Journal of Agricultural Science*, vol. 144, no. 1, pp. 31-43, 2006. doi: 10.1017/S0021859605005708.

[16] S. De Mey, M. Demont, and A. Diagne, “Estimating bird damage to rice in Africa: Evidence from the Senegal River Valley,” *Journal of Agricultural Economics*, vol. 63, no. 1, pp. 175-200, 2012.

[17] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35-45, 1960. doi: 10.1115/1.3662552.

[18] H. W. Kuhn, “The Hungarian method for the assignment problem,” *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83-97, 1955. doi: 10.1002/nav.3800020109.

[19] J. Cao, J. Pang, X. Weng, R. Khirodkar, and K. Kitani, "Observation-centric SORT: Rethinking SORT for robust multi-object tracking," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, 2023, pp. 9686-9696.

[20] P. Wang, Y. Wang, and D. Li, "DroneMOT: Drone-based multi-object tracking considering detection difficulties and simultaneous moving of drones and objects," arXiv:2407.09051, 2024.