EPJ Web of Conferences **377**, 02010 (2026) https://doi.org/10.1051/epjconf/202637702010
IPS 2026

Physics-aware vision instrumentation for stingless bee counting at hive entrance using hybrid edge-cloud object detection

Mohd Amri Md Yunus[1*], *Lari Andres* Sanjaya[2], *Celestine* Hiu Shun Yi[1], *Shafishuhaza* Sahlan[1], *Saharudin* Saharudin[3], *Sudrajat* Harris[4], and *Hendri Maja* Saputra[5]

[1]Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 Johor Bahru, Johor, Malaysia
[2]Faculty of Mathematics and Natural Sciences, Universitas Negeri Jakarta, 13220 Jakarta Timur, Jakarta, Indonesia
[3]Departemen of Electrical Engineering, Institut Teknologi Indonesia, 15314 Tangerang Selatan, Banten, Indonesia
[4]Department of Chemical Engineering, Politeknik Negeri Bandung, 40559 Bandung Barat, Indonesia
[5]Research Center for Smart Mechatronics - National Research and Innovation Agency (BRIN), Indonesia

Abstract. Stingless bee entrance monitoring requires a non-invasive tool to measure colony traffic without disrupting foraging. A hybrid edge-cloud object identification system and physics-aware vision instrumentation framework are used to track stingless bees in this paper. The system uses a Raspberry Pi 4 edge node, Sony IMX296 Global Shutter camera, Only Look Once (*YOLO*)-based detection, and Observation-Centric Simple Online and Realtime Tracking (OC-SORT) tracking. Because entry activity implies foraging intensity, the traffic count can be a non-invasive proxy for colony health and yield. The edge-side *YOLO11* and *OC-SORT* tracking pipeline found trajectory fragmentation during high-speed ingress. A frame interval of 0.0667 s was achieved using a 15 FPS edge processing rate. Inter-frame displacement may approach 0.333 m for bee motion exceeding 5 ms^{-1}, producing missed detections and identity switching. Thus, a sampling-based tracking failure situation happens when a bee travels more than the tracker's maximum association distance between two processed frames. This illustrates that frame rate, bee velocity, field-of-view scale, detector recall, tracker association tolerance, and biologically meaningful bee counting interpretation all affect bee tracking reliability. Validation using 14 one-minute Geniotrigona thoracica entrance videos showed an average IN counting accuracy of 85.0%, OUT counting accuracy of 66.7%, and total counting accuracy of 76.3% compared with manual video counting.

1 Introduction

Stingless bees are important tropical pollinators and are widely associated with meliponiculture, crop pollination, honey production and ecological sustainability [1, 2]. In

* Corresponding author: amri@utm.my

practical beekeeping, external hive entrance activity is one of the simplest indicators of colony condition. A healthy colony normally shows active foraging traffic, while abnormal entrance behaviour may indicate stress, predator disturbance, weak colony activity or environmental changes [3]. However, manual observation of bee traffic is labour-intensive, subjective and difficult to repeat over long monitoring periods [4, 5].

Computer vision offers a promising solution because it allows non-invasive monitoring of bees without opening the hive [6, 7]. Nevertheless, bee entrance monitoring is not only an artificial intelligence problem [8]. From the viewpoint of instrumentation and computational physics, the system must measure small and fast-moving objects, maintain temporal continuity of trajectories, and convert image-based position data into interpretable motion quantities [9]. Therefore, tracking reliability depends on the interaction between optical sensing, frame rate, computational latency, motion blur, object detector recall, and tracker association logic [10, 11].

This paper reports the development of a hybrid edge-cloud framework for real-time species identification and directional counting of stingless bees. The edge node was responsible for lightweight object detection and tracking, while the cloud server handled species classification. Moreover, this paper reframes the work for the scope of instrumentation and computational physics by focusing on the physical interpretation of bee movement and the sampling limitation of the tracking instrument.

The main contribution of this paper is not merely to show that a low frame rate affects tracking. Instead, the contribution is to formulate a sampling-based design rule for bee entrance tracking. The paper proposes that tracking failure occurs when the physical inter-frame displacement of a bee exceeds the allowable displacement that the tracker can associate.

2 System concept and instrumentation framework

Fig. 1 illustrates the monitoring system which consists of four main layers: optical sensing, edge inference, trajectory analysis, and cloud server monitoring/communication. This structure follows the general cloud-edge-device concept, where time-sensitive tasks can be processed near the sensing node while more computationally demanding analysis can be handled in the cloud [12].

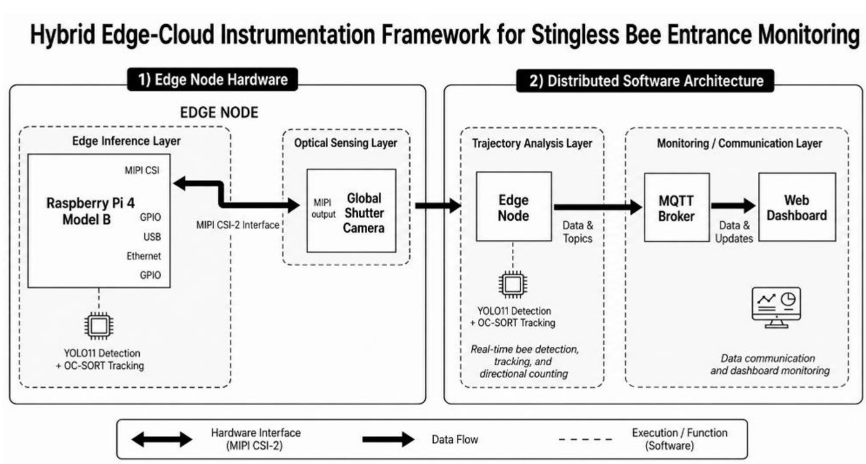

Fig. 1. Hybrid edge-cloud instrumentation framework for stingless bee entrance monitoring. The edge node performs real-time body tracking and directional counting, while the cloud server performs data updates and dashboard visualisation

The optical sensing layer captures video of the hive entrance using Sony IMX296 Global Shutter sensor. The edge inference layer performs real-time bee detection and multi-object tracking. The trajectory analysis layer converts frame-wise object positions into directional movement states. The cloud server monitoring/communication layer performs data updates and dashboard visualisation.

The visual sensing is based on a Raspberry Pi 4 Model B connected to a Sony IMX296 global shutter camera through the MIPI CSI-2 interface. The use of a global shutter camera is important because all pixels are exposed simultaneously. This helps reduce rolling-shutter distortion when fast-moving bees pass through the field of view. This optical design was selected to support high-speed entrance monitoring and reduce motion artefacts in the captured video.

The software architecture follows a distributed edge-cloud design. The Raspberry Pi operates as the edge node and runs the lightweight *YOLO11* detector and *OC-SORT* tracker. Lightweight telemetry, such as total IN/OUT count, is transmitted to a web-based dashboard through an MQTT publisher-subscriber structure.

2.1 Detection, tracking and counting method

The proposed vision system uses a YOLO-based detector and OC-SORT tracker for Geniotrigona thoracica entrance counting. In this paper, all detected bees are treated as a single tracking class. This is appropriate because the objective is not species identification, but reliable detection, identity continuity and directional counting at the hive entrance. The detector localises bees in each processed frame, while OC-SORT links detections across consecutive frames to maintain object identity and trajectory history.

The dataset was created using video frames of Geniotrigona thoracica activity at the hive entrance. A total of 14 hive-entrance videos were selected for validation, and each video was analyzed using a fixed 1-minute counting interval. Frames were extracted from the video sequences and manually annotated using tight bounding boxes around visible bee bodies. The annotation focused on bee body localization rather than species labelling because the edge-side YOLO11 model was designed as two movement classes, namely "IN" and "OUT" detector for tracking and directional counting. The recordings represented practical outdoor hive-entrance conditions, including natural lighting variation, shadow near the entrance, partial occlusion, hovering behavior, fast-moving bees, and bees crossing the field of view at different speeds. This allows the model to prioritise detection and tracking robustness, especially for small, blurred or partially occluded bees near the entrance. Directional counting is performed using a three-zone buffer logic. Two virtual vertical lines divide the image into Hive Zone, Buffer Zone and Air Zone. Line A is placed at 30% of the frame width, while Line B is placed at 60% of the frame width as can be seen from Fig. 2. A valid IN count is recorded only when the tracked bee moves from Air Zone through Buffer Zone into Hive Zone. A valid OUT count is recorded only when the tracked bee moves from Hive Zone through Buffer Zone into Air Zone. This buffer-zone logic acts like a hysteresis filter. It rejects bees that hover near the entrance, turn back inside the buffer zone or perform guarding behaviour without completing a full directional transition. However, the logic depends strongly on identity continuity. If the tracker loses the bee ID during the buffer transition, the state machine resets and the count may be missed.

The YOLO11 detector was implemented using the Ultralytics object-detection framework and configured for lightweight hive-entrance bee detection. The dataset followed the same annotation structure as the previous stingless bee entrance-detection experiment, consisting of 165 original images with 206 bounding-box annotations across two movement classes, namely "*IN*" and "*OUT*". Augmentation using rotation, flipping, and exposure adjustment increased the dataset to 387 images. The augmented dataset was divided into 336 training images, 17 validation images, and 34 testing images. Training was performed for 100 epochs using an input image size of 640 × 640 pixels and batch size of 16. The confidence threshold and non-maximum suppression IoU threshold were set to 0.25 and 0.70, respectively, during inference following the default Ultralytics prediction configuration. The detector achieved an overall precision of 0.959, recall of 0.938, F1-score of 0.948, mAP@50 of 0.918, and mAP@50–95 of 0.511. These values indicate that the detector was suitable for small-object bee movement detection, although additional motion-blurred and high-speed samples are still required to improve tracking continuity during rapid entrance crossing.

For validation, each video sample was analyzed using a fixed 1-minute counting interval. Within each 1-minute interval, the automated system produced the *YOLO + OC-SORT* predicted *IN* and *OUT* counts, while the same video segment was manually reviewed to obtain the Manual in and Manual Out counts. The automated counts were then compared against the manual counts to calculate IN accuracy, OUT accuracy and total counting accuracy.

Table 1 shows the Tracking and instrumentation parameters used in the proposed framework. The edge-side tracking model was configured for lightweight bee detection and real-time trajectory association. The objective of training was to support robust localisation of bees in entrance videos, so that the tracker could preserve object identity during Air-Buffer-Hive or Hive-Buffer-Air transitions. The counting performance can be evaluated by comparing automated *IN/OUT* counts with manual counting from video observations.

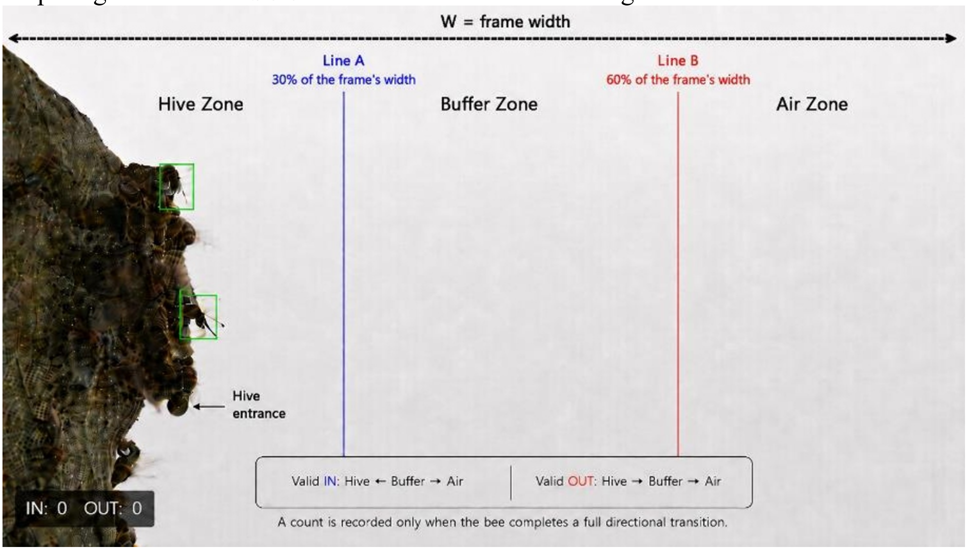

Fig. 2. Three-zone buffer logic for bee entrance tracking. Line A is positioned at 30% of the frame width and Line B is positioned at 60% of the frame width. A count is recorded only when the bee completes a full directional transition.

2.2 Physics-aware trajectory formulation

The object detector and tracker produce the centroid position of each detected bee in every processed frame. The image position of bee, *i* at time, *t* can be written as:

$$p_i(t)=\{x_i(t),y_i(t)\} \quad (1)$$

where $x_i(t)$ and $y_i(t)$ are the image coordinates in pixels. If the camera is calibrated using a spatial scale factor k_s in metres per pixel, the projected displacement between two consecutive frames, Δs_i can be estimated as:

$$\Delta s_i = k_s\sqrt{[x_i(t+\Delta t)-x_i(t)]^2+[y_i(t+\Delta t)-y_i(t)]^2} \tag{2}$$

where Δt is the frame interval is second. Then, the projected velocity, U_i is:

$$U_i = \frac{\Delta s_i}{\Delta t} \tag{3}$$

The frame interval, Δt depends on the effective processing frame rate and can be calculated from:

$$\Delta t = \frac{1}{f_s} \tag{4}$$

where f_s is the operational frame rate, and the inter-frame displacement, d can be calculated from

$$d = \frac{U_i}{f_s} \tag{5}$$

These calculations show that frame rate should not be interpreted only as a computational performance metric. It is also a physical sampling parameter.

Table 1. Tracking and instrumentation parameters used in the proposed framework.

Parameter	Value
Target species for counting	Geniotrigona thoracica
Edge node	Raspberry Pi 4 Model B
Edge camera	Sony IMX296 global shutter sensor
Detection model	YOLO11
Tracking algorithm	OC-SORT
Counting logic	Three-zone buffer logic
Frame-width boundary Line A	30% of the frame width
Frame-width boundary Line B	60% of the frame width
Edge operating frame rate	~15 FPS
Reference video frame rate	60 FPS
High-speed ingress condition	5 ms^{-1}
Main output	IN/OUT directional bee count
Validation method	Comparison with manual video counting
Counting interval per sample	1 minute

3 Results

Each video sample was evaluated over a fixed 1-minute counting interval, during which *YOLO + OC-SORT* predicted counts were compared with *Manual In* and *Manual Out* counts obtained from the same video segment. The manual counts are denoted as *Manual In*, *Manual Out,* and *Manual Total* (*Manual In* + *Manual Out*), while the predicted counts are denoted as *OC-SORT In*, *OC-SORT Out*, and *OC-SORT Total* (*OC-SORT In* + *OC-SORT Out*).

The accuracy percentage was computed from the absolute counting error relative to the corresponding manual count. The equations report the percentage agreement between the *YOLO + OC-SORT* count and the corresponding *Manual* video count are:

$$IN\ accuracy = 100 - \left(\left|\frac{OC\text{-}SORT\ IN - Manual\ In}{Manual\ In}\right| \times 100\right) \tag{6}$$

$$OUT\ accuracy = 100 - \left(\left| \frac{OC{-}SORT\ OUT - Manual\ OUT}{Manual\ OUT} \right| \times 100 \right) \quad (7)$$

$$TOTAL\ accuracy = 100 - \left(\left| \frac{OC{-}SORT\ Total - Manual\ Total}{Manual\ Total} \right| \times 100 \right) \quad (8)$$

The average accuracy across from all video samples was calculated as:

$$Average\ accuracy\ (\%) = \frac{sum\ of\ video\ accuracy\ percentages}{number\ of\ videos} \quad (9)$$

The edge-side counting algorithm was evaluated using 14 Geniotrigona thoracica entrance videos. Manual video counting produced 687 *Manual In* events and 638 *Manual Out* events, giving 1325 total manual events. The *YOLO + OC-SORT* pipeline predicted 604 *IN* events and 429 *OUT* events, giving 1033 total predicted events. This corresponds to aggregate errors of -83 for *IN* counting, -209 for *OUT* counting and -292 for *TOTAL* counting.

From Fig 3. and using the manual counts as the reference, the average *IN* counting accuracy was 85.0%, the average *OUT* counting accuracy was 66.7% and the average total counting accuracy was 76.3%. The best total counting accuracy was 86.4% for video sample *V4*, while the lowest total counting accuracy was 53.5% for video sample *V2*. The *IN* accuracy ranged from 50.0% to 100.0%, whereas the *OUT* accuracy ranged from 50.0% to 80.0%. The lower *OUT* counting accuracy indicates an important limitation of the current system. Exiting bees are more difficult to count because they often emerge from the dark or partially occluded hive entrance and accelerate rapidly toward the Air Zone. This condition can produce short visibility duration, partial body appearance, high-speed motion blur, missed bounding boxes during boundary crossing, and temporary identity switching in the *OC-SORT* tracker. Since the three-zone buffer logic requires a continuous tracking identity from Hive Zone to Buffer Zone and then Air Zone, any missed detection during this transition can reset the state machine and produce an undercount. Therefore, the lower *OUT* accuracy is not only caused by the counting logic, but by the combined effects of detector recall, temporal sampling, motion blur, field-of-view placement, and tracker association tolerance.

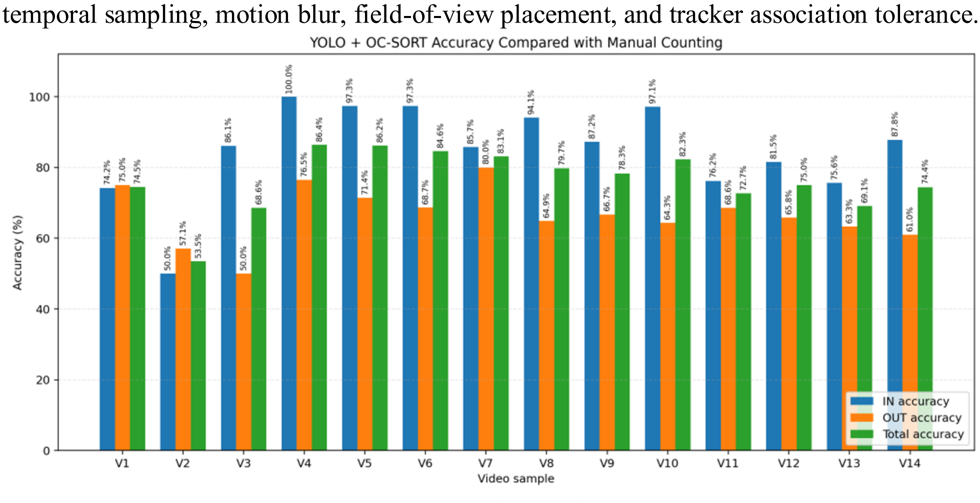

Fig. 3. *YOLO + OC-SORT* counting accuracy compared with manual video counting across 14 video samples. Each sample represents a 1-minute counting interval. *IN* accuracy, *OUT* accuracy and *TOTAL* accuracy were calculated by comparing the automated counts with the corresponding *Manual In*, *Manual Out* and *TOTAL* manual counts.

3.1 Sampling-based tracking failure criterion

For a vision-based tracker, the physical distance travelled by a bee between two processed frames can be represented by equation (5) in section 2.2. Tracking is expected to remain reliable only when this inter-frame displacement, d is smaller than the maximum displacement that the tracker can associate between consecutive frames, i.e. $d \leq d_{allow}$.

Where d_{allow} is the allowable association distance of the tracker. Therefore, tracking failure becomes likely when $d$exceeds d_{allow}. In practical terms, this means that a bee moving too far between two processed frames may not be associated with the same identity, causing missed detections, trajectory fragmentation or identity switching. To quantify the sampling limitation, the maximum ingress velocity was taken as 5 m s^{-1}. This value was used as a conservative high-speed design assumption because published honeybee flight studies have reported bee flight-related testing and natural flight speeds within this range or higher. Nachtigall measured honeybee aerodynamic conditions at velocities between 0.5 and 5 ms^{-1} [13]. Although direct stingless bee velocity measurement was not performed in the present study, the 5 m s^{-1} assumption provides a reasonable upper-bound reference for evaluating sampling-related tracking failure in small, fast-moving bees. Table 2 summarizes the expected inter-frame displacement at different effective frame rates. The calculation shows that increasing the frame rate reduces the distance travelled by the bee between two processed frames, which improves the probability that *OC-SORT* can maintain the same bee identity.

Table 2. Tracking and instrumentation parameters used in the proposed framework.

Frame rate, f_s	Frame interval, ($\Delta t=1/f_s$)	Inter-frame displacement, ($d=U/f_s$)	Tracking implication
15 FPS	0.0667 s	0.333 m	High risk of missed detection and ID switching
60 FPS	0.0167 s	0.083 m	More suitable, but still detector-dependent
100 FPS	0.0100 s	0.050 m	More reliable if tracker tolerance is around 0.05 m

The numerical trend in Table 2 is illustrated schematically in Fig. 4. At 15 FPS, the inter-frame displacement is approximately 0.333 m, which may exceed the allowable association distance of the tracker. This condition increases the likelihood of trajectory fragmentation and identity switching. At 60 FPS, the displacement decreases to approximately 0.083 m, producing a borderline tracking condition. At 100 FPS and above, the displacement becomes smaller, improving the likelihood that the tracker can associate the same bee across consecutive frames.

The same bee-tracking algorithm may succeed or fail depending on field-of-view size, camera placement and tracker tuning. A close-up view with a small physical field of view may require much higher frame rate because the bee crosses the buffer region quickly. A wider field of view may tolerate larger displacement but may reduce bee image size and detection confidence. Therefore, the design problem involves a trade-off between spatial resolution, field coverage, frame rate and detector accuracy.

It also suggests that 60 FPS alone is not a complete solution. Even at 60 FPS, identity switching may occur if the detector misses the bee for several frames due to motion blur, occlusion or poor generalization to fast-moving objects. Therefore, reliable bee tracking requires four conditions: sufficient temporal sampling, motion-blur-aware training data, tracker memory that can survive short detection gaps, and a buffer-zone logic that is tolerant to realistic insect motion.

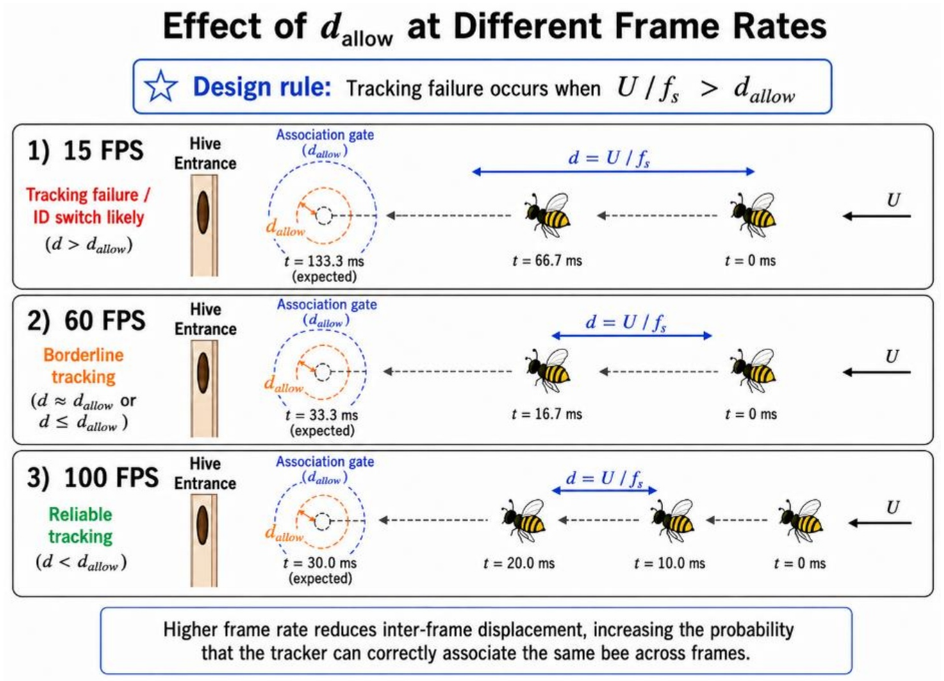

Fig. 4. Effect of allowable inter-frame displacement, d_{allow}, at different frame rates. Higher frame rate reduces the inter-frame *displacement* $d = U/f_s$, increasing the probability that the tracker can correctly associate the same bee across frames.

4 Conclusion

This paper presented a physics-aware vision instrumentation framework for Geniotrigona thoracica entrance tracking and counting using edge-based object detection and *YOLO* + *OC-SORT* tracking. The system combines a Raspberry Pi 4 edge node, Sony IMX296 global shutter camera, *YOLO11* detector, OC-SORT tracker, three-zone buffer counting logic and MQTT-based dashboard monitoring. The counting validation across 14 video samples showed average accuracies of 85.0% for *IN* counting, 66.7% for *OUT* counting and 76.3% for total counting, when compared with manual video counts. The edge-side tracking result also revealed trajectory fragmentation during fast entrance motion, especially when the detector failed to maintain continuous bounding-box localization during line-crossing events.

The main contribution of the paper is the sampling-based interpretation of tracking failure. At the measured edge-processing rate of approximately 15 *FPS*, the frame interval is 0.0667 s. For bee velocity above 5 ms^{-1}, the bee may travel more than 0.333 m between processed frames. This can exceed the allowable association distance, d_{allow} of the tracker, leading to missed detections, identity switching and undercounting. The tracking failure condition is expressed as $U/f_s > d_{allow}$.

This condition provides a practical design rule for future bee-tracking instruments. It shows that reliable tracking requires not only higher FPS, but also suitable camera placement, calibrated field of view, motion-blur-aware detection, robust tracker memory and appropriate buffer-zone design. Future research work should calibrate the image scale, measure bee velocity distribution, tune *OC-SORT* association parameters, add motion-blurred training samples, and measure bee morphology to support bee flight analysis.

This publication was funded by the Indonesia Endowment Fund for Education (LPDP) on behalf of the Indonesian Ministry of Higher Education, Science and Technology, and managed under the EQUITY Program (Contract Number 4308/B3/DT.03.08/2025 and B/284/UN39/HK.07.00/2025).

References

1. T.A. Heard, Annu. Rev. Entomol. 44, 183 (1999). https://doi.org/10.1146/annurev.ento.44.1.183
2. A. Chauhan, L. Imtinaro, Int. J. Farm Sci. 14, 64 (2024). https://doi.org/10.5958/2250-0499.2024.00027.X
3. Y. Lejing, et al., Int. J. Adv. Comput. Sci. Appl. 16(8) (2025). https://thesai.org/Downloads/Volume16No8/Paper_103-From_Review_to_Refinement.pdf
4. N.H.K. Anuar, M.A.M. Yunus, N. Kasuan, et al., Proc. 2023 IEEE Int. Conf. Agrosyst. Eng. Technol. Appl. (AGRETA), 80 (2023). https://doi.org/10.1109/AGRETA57740.2023.10262636
5. N.H.K. Anuar, M.A.M. Yunus, M.A. Baharudin, et al., Proc. 2021 IEEE Int. Conf. Power Eng. Appl. (ICPEA), 149 (2021). https://doi.org/10.1109/ICPEA51500.2021.9417758
6. B.J. Spiesman, C. Gratton, R.G. Hatfield, et al., Sci. Rep. 11, 7580 (2021). https://doi.org/10.1038/s41598-021-87210-1
7. A.M.B. Ratnayake, H.M. Yasin, A.G. Naim, et al., Appl. Syst. Innov. 7, 62 (2024). https://doi.org/10.3390/asi7040062
8. M.H.C. Leme, V.S. Simm, D.R. Tanno, et al., Lect. Notes Comput. Sci., 730 (2023). https://doi.org/10.1007/978-3-031-49018-7_52
9. G. Vdoviak, T. Sledevič, A. Serackis, et al., Agriculture 15, 1019 (2025). https://doi.org/10.3390/agriculture15101019
10. C. Lei, Y. Lu, Z. Xing, et al., Insects 15, 974 (2024). https://doi.org/10.3390/insects15120974
11. W. Hua, Q. Chen, Artif. Intell. Rev. 58, 162 (2025). https://doi.org/10.1007/s10462-025-11150-9
12. P. Yu, F. Teng, W. Zhu, et al., Front. Plant Sci. 16, 1668545 (2025). https://doi.org/10.3389/fpls.2025.1668545
13. W. Nachtigall, U. Hanauer-Thieser, J. Comp. Physiol. B 162, 267 (1992). https://doi.org/10.1007/BF00357534