Advance Sustainable Science, Engineering and Technology (ASSET)
Vol. 8, No.3, July 2026, pp. 02603023-01 ~ 02603023-018
ISSN: 2715-4211 DOI: https://doi.org/10.26877/asset.v8i3.2879

Optimization of Tracking Algorithm on Mouse Movement Monitoring Platform in Medical Testing

Sutrisno Ibrahim[1*], Rahmat Rohmani[1], Joko Hariyono[1], Faisal Rahutomo[1], Nanang Wiyono[2], and Ratih Yudhani[2]

[1]Department of Electrical Engineering, Faculty of Engineering, Sebelas Maret University, Ir. Sutami 36A Kentingan, Jebres, Surakarta, Central Java, 57126, Indonesia.

[2]Department of Medicine, Faculty of Medicine, Sebelas Maret University, Ir. Sutami 36A Kentingan, Jebres, Surakarta, Central Java, 57126, Indonesia.

*sutrisno@staff.uns.ac.id

Abstract. Accurate monitoring of mouse behavior in the Elevated Plus Maze (EPM) is essential for anxiety-related biomedical research, yet manual observation is time-consuming, subjective, and prone to human error. This study proposes an optimized automated tracking framework that integrates YOLOv8 detection with tracking methods including an adaptive Kalman Filter and DeepSORT, and compares them with conventional trackers such as CSRT and GOTURN. System performance was evaluated using Intersection over Union (IoU), Center Location Error (CLE), and Frames Per Second (FPS), with the Weighted Scoring Method (WSM) used for overall performance comparison. Experimental results show that the proposed YOLOv8 with adaptive Kalman filtering (frame interval = 5) provides the best balance between accuracy and computational efficiency. The approach achieved an IoU of 0.89 and CLE of 2.34 while increasing processing speed from 10.44 FPS to 22.55 FPS, representing an improvement of approximately 116% compared with the baseline configuration. Despite a slight increase in failure rates, the framework maintained stable real-time tracking performance under laboratory conditions. These results demonstrate that the proposed system improves both tracking efficiency and robustness, offering a reliable automated solution for high-throughput behavioral monitoring. The framework is particularly suitable for laboratory automation environments, supporting more objective behavioral assessment and improved data integrity in preclinical biomedical research.

Keywords: elevated plus maze, mouse tracking, biomedical, YOLO, DeepSORT, kalman filtering

(Received 2025-10-19, Revised 2026-03-16, Accepted 2026-05-06, Available Online by 2026-06-17)

1. Introduction

Technological advances in biomedical research demand experimental systems that are accurate, efficient, and reproducible. In pre-clinical studies, rodents such as mice and rats remain indispensable for evaluating therapeutic interventions before clinical trials in humans [1]. Monitoring their movement provides valuable insights into neurological, physiological, and behavioral responses [2]. However, in many laboratories—including those in developing countries—observations are still performed manually or semi-automatically, leading to subjectivity, high error rates, and inefficiency [3]. Accurate behavioral monitoring is particularly critical in assays such as the Elevated Plus Maze (EPM), a validated standard for assessing anxiety-related behavior in rodents [4].

The emergence of artificial intelligence (AI) and computer vision has revolutionized automated monitoring platforms. YOLO (You Only Look Once) and its successors are widely recognized for combining high accuracy with real-time detection performance [5], [6]. Behavioral monitoring platforms leveraging computer vision have been increasingly applied in neuroscience and pharmacology [7]. Despite this, EPM assessments are still largely conducted manually, which is time-consuming and prone to inter-observer variability [8]. Automated methods offer significant advantages but face challenges in dynamic environments such as irregular movement, occlusion, and variable lighting conditions[9].

Several studies have addressed animal tracking with diverse strategies [10], proposed a customized tracking algorithm (CTA) for livestock under occlusion using Detectron2, [11] implemented YOLOv8 with DeepSORT to track small animals such as butterflies and squirrels. [12] developed DAMM, a Mask R-CNN and SORT-based model for mouse tracking, combined YOLOv4, SORT, and RFID for multi-mouse behavioral identification. Although effective, CNN-based methods such as Faster R-CNN, Mask R-CNN, and classification of animal behavior. Extensions of these frameworks now support multi-animal pose tracking [13], [14], and identity preservation under social interactions [15]. Nonetheless, these methods are often data-intensive, require extensive training, and remain computationally demanding, restricting their real-time applicability in standard laboratory environments [16], [17]. The rapid evolution of YOLO models from YOLOv3 to YOLOv8 has brought significant improvements in balancing accuracy and efficiency [18]–[19] Lightweight architectures and hardware accelerators have been investigated to overcome computational barriers [20], [21]. Software toolkits such as DeepPoseKit [28] and DeepFly3D [22] also contribute to animal tracking, but challenges remain in scalability and generalization. Pose-estimation pipelines such as DeeperCut [23], AlphaTracker [24], and SLEAP [25] provide alternatives, yet they too require extensive computational resources. Recent reviews emphasize the need for efficient, explainable, and scalable AI frameworks in behavioral neuroscience [26], [27].

The evolution of real-time object detection has been significantly accelerated by advanced frameworks such as YOLOv4 and YOLOv5, which offer an optimal balance between computational speed and detection accuracy [28]. For continuous trajectory monitoring, these detection models are frequently paired with efficient data association metrics like Simple Online and Realtime Tracking (SORT) to reliably maintain object identities across consecutive frames [29]. Beyond basic bounding-box tracking, deep learning systems such as SLEAP have further advanced the field by enabling precise multi-animal pose estimation and behavioral mapping [30]. Furthermore, the application of computer vision and Convolutional Neural Networks (CNNs) has expanded widely into broader veterinary and biomedical contexts, demonstrating robust performance in automating the complex behavior recognition of various animals, including livestock [31], [32].

YOLO continues to dominate real-time detection benchmarks, with YOLOv5, YOLOv6, and YOLOv7 demonstrating stepwise improvements in training efficiency and detection accuracy [32]–[33],[37],[38]. In rodent studies, vision-based tracking is being integrated with multimodal data such as RFID [34] and neural recordings [35], opening opportunities for richer behavioral analysis. However, the gap remains between offline high-accuracy approaches and real-time, computationally efficient solutions adaptable to standard laboratory setups [36].

This study aims to optimize a tracking algorithm for a mouse movement monitoring platform used in biomedical behavioral testing. Recent studies have applied deep learning–based detectors, particularly YOLO models combined with trackers such as DeepSORT, CSRT, or GOTURN, for automated rodent behavior analysis. While these approaches improve detection accuracy, many suffer from high computational cost or unstable tracking under laboratory conditions, such as rapid movements, occlusion, and varying illumination. In addition, common YOLO + DeepSORT pipelines are primarily designed for general multi-object tracking and often introduce unnecessary complexity for single-animal behavioral monitoring. To address this gap, this study proposes an optimized YOLOv8-based tracking framework tailored for rodent movement monitoring. The proposed approach integrates YOLOv8 detection with adaptive filtering strategies to improve tracking stability while maintaining computational efficiency. Its performance is compared with established trackers, including CSRT, GOTURN, and DeepSORT, under real laboratory conditions.

The main contributions of this study are as follows. **(1)** The development of an optimized YOLOv8-based tracking framework for rodent monitoring in Elevated Plus Maze (EPM) assays. **(2)** A comparative evaluation of several tracking methods using standardized performance metrics, including Intersection over Union (IoU), Center Location Error (CLE), Frames Per Second (FPS), and the Weighted Scoring Model (WSM). **(3)** Demonstration of improved computational efficiency, enabling real-time behavioral video analysis while maintaining reliable tracking accuracy. **(4)** The provision of a practical automated monitoring system that reduces manual observation workload and improves reproducibility in pre-clinical biomedical research. This research is a continuation of our previous work, which focused on rodent movement monitoring in the T-Maze paradigm, while the present study extends the framework to the EPM experimental setup [37].

2. Research Methods

2.1. Dataset

2.1.1. Data Source

Data were collected from Elevated Plus Maze (EPM) experiments conducted under controlled laboratory conditions. To capture video data, several components are required. Fig. 1 shows the arrangement of hardware components for data acquisition. Each video was recorded at a resolution of 1280×720 (HD) with a frame rate of 30 frames per second (fps). As in other computer vision systems [38], before conducting behavioral analysis, the most important stages that must be carried out include: data collection with a camera, segmentation, object detection, tracking and finally behavioral analysis.

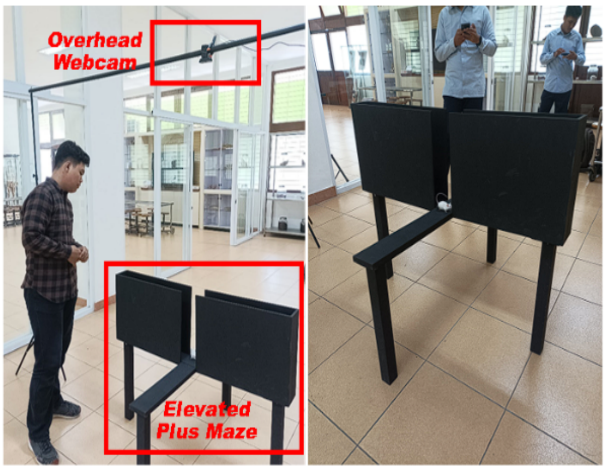

Figure 1. Hardware setup

Integration of a USB camera with a monitor as a video capture or recording tool is carried out which is connected to the monitor so that it is integrated with the object detection system. Apart from that, a design was made for the elevated plus maze with the following specifications [15]:

2.1.1. *Annotation Process*

Collected video data were converted into an annotated dataset to train the YOLOv8-based object detection model and to serve as a baseline for evaluating the tracking algorithms. Each video was decomposed into individual frames at its original frame rate to preserve temporal consistency for subsequent tracking analysis.

YOLOv8 annotation was performed by drawing bounding boxes around the target object (mouse) in each frame. The annotation process focused on accurately capturing the spatial extent of the mouse body, excluding shadows and environmental artifacts to minimize labeling noise. All annotations were saved in YOLO format, consisting of normalized bounding box coordinates relative to the image dimensions. The annotated dataset was then divided into training (70%), validation (20%), and testing (10%) subsets to support detector training and unbiased performance evaluation.

Tracking ground truth annotation was performed by selecting video sections focused on segments with high levels of subject activity to ensure diversity in movement patterns in the test data. To obtain location data as a baseline, object position annotations are required. The annotation process was performed semi-automatically, utilizing the YOLOv8 model to automatically generate annotations, which were then refined manually using the OpenCV-based ROI selection method. To validate the data result, a visual check is performed for each frame and the corresponding ground truth to ensure that the target position data corresponds to the actual situation.

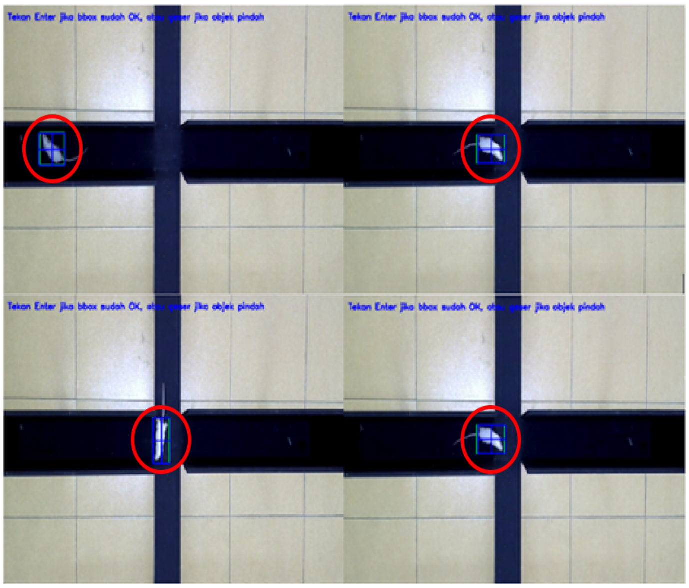

Figure 2. Ground truth annotation process as baseline

2.2. *Model Architecture*

2.2.1. *Proposed Method Pipeline*

Fig. 4. Shows the illustration of the proposed tracking approach. Initial object detection is performed using the YOLOv8-nano algorithm. This algorithm is a lightweight variant of YOLOv8 that has high interference speed and low resource consumption, making it suitable for real-time implementation with limited computing resources. The detection process using YOLO is known to require quite high

computational resources, especially to achieve low inference time [14]. In systems with hardware limitations, this can trigger lagging, which is a delay in processing video data in real-time, thus impacting overall system performance. To overcome this, a frame skipping strategy is applied by skipping a number of frames before performing the next detection. This approach aims to reduce the frequency of use of the YOLO model, thereby reducing the computational load and speeding up the system response time. In its implementation, a skipping interval is used with a value of N = {3, 5, 10}.

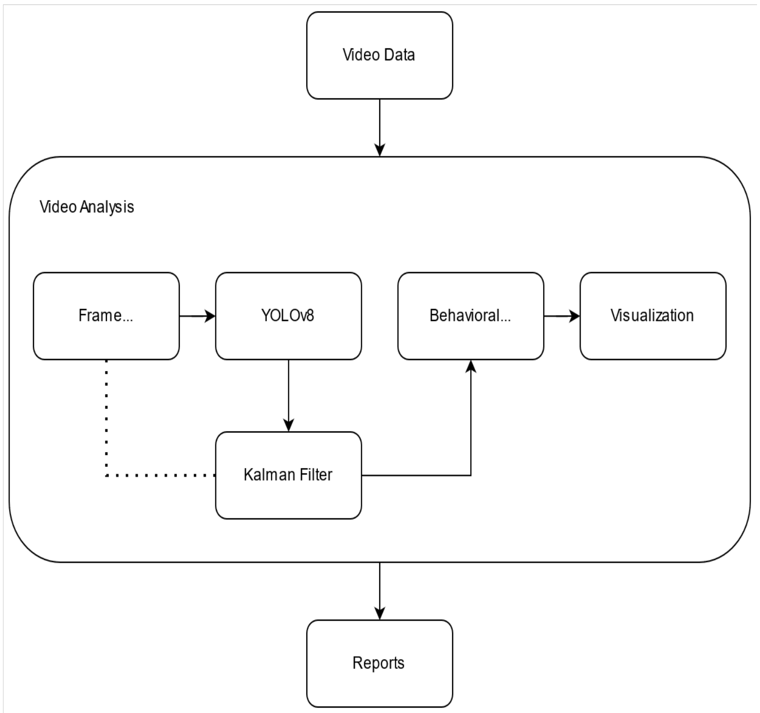

Figure 3. *illustrates the architecture of the proposed tracking approach.*

The application of the frame skipping technique results in the loss of target location data on frames that do not apply YOLO detection. This results in discontinuous tracking data. This is overcome by applying a Kalman filter to estimate the position of the object on frames that are not directly detected by YOLO. In order for the Kalman filter to produce accurate predictions, initial adjustments are required to the estimated parameters, such as the position and velocity of the object. This adjustment is carried out through a direct observation mechanism using YOLO detection on a number of N initial frames, the number of which is adjusted to the value of the skipping interval.

2.2.2. Kalman Filter-based Motion Modeling

The state model used in this filter consists of x, y, w, h, Δx, Δy, Δw, and Δh. In the prediction process, the Kalman filter uses the transition matrix (F) to estimate the target location based on the previous condition variables. In the observation process using YOLO, changes in state are based on the observation matrix (H) which takes into account information from the bounding box. State model at time step k is defined as:

$$x_k = \left[x, y, w, h, \dot{x}, \dot{y}, \dot{w}, \dot{h}\right]^T \quad (1)$$

The estimation of state is modeled using a constant-velocity assumption. Initiation of the transition matrix (F) and observation matrix (H) values is carried out by applying 4x4 identity matrix and 4x4 zero matrix, respectively, as shown below:

$$x_{k|k-1} = Fx_{k-1} \tag{2}$$

$$F = [\, I\ I\ 0\ I\ ], H = [\, I\ 0\ ] \tag{3}$$

In update step, measurement value are obtained from YOLOv8 detections, detections result is transformed from (x_{min}, y_{min}, x_{max}, y_{max}) format to (x,y,w,h) format defined as:

$$z_k = \left[\, x_k^{obs}, y_k^{obs}, w_k^{obs}, h_k^{obs} \right]^T \tag{4}$$

$$z_k = Hx_k + v_k \tag{5}$$

Prior to incorporating the measurement at time k, the state and covariance are predicted as:

$$x_{k|k-1} = Fx_{k-1} \tag{6}$$

$$Q_k = \alpha Q_{k-1} + \beta I \tag{7}$$

$$P_{k|k-1} = FP_{k-1}F^T + Q_k \tag{8}$$

Errors of prediction and measurement are calculated as innovation defined as:

$$y_k = z_k - Hx_{k|k-1} \tag{9}$$

Based on the innovation magnitude, the measurement noise covariance is adapted as:

$$R_k = diag(|y_k| + \varepsilon) \tag{10}$$

So, the Kalman Gain is calculated as:

$$K_k = P_{k|k-1} H^T \left(HP_{k|k-1}H^T + R_k \right)^{-1} \tag{11}$$

Then, the state and covariance are updated according to:

$$x_k = x_{k|k-1} + K_k y_k \tag{12}$$

$$P_k = (I - K_k H)P_{k|k-1} \tag{13}$$

After estimate the state, estimated state is converted back to bounding box coordinates:

$$Convert: \{x_{min} = x_k - \frac{w_k}{2}\ y_{min} = y_k - \frac{h_k}{2}\ x_{max} = x_k + \frac{w_k}{2}\ y_{max} = y_k + \frac{h_k}{2} \tag{14}$$

2.3. Optimization

The test results were analyzed using the weighted scoring method (WSM) to determine the most optimal method. Weighted scoring method (WSM) is a decision-making method that takes into account various criteria including qualitative and quantitative measurements [16]. The most optimal method was then implemented in the tracking system to evaluate the effect of tracking algorithm implementation on software processing speed. The evaluation results are normalized to a scale of 0-1 and then the degree of importance for each metric is determined [17]. The following is the formula used to calculate the overall performance score:

$$Score = (w_1 \times v_1) + (w_2 \times v_2) + \cdots + (w_n + v_n) \tag{15}$$

2.4. Evaluation Metrics

To assess the accuracy and consistency of the system tracking, performance metrics are taken. Metrics used to measure algorithm performance include Intersection over Union (IoU), Center Location Error (CLE), Frames per Seconds (FPS), Failure Rate (FR). Failure rate (FR) is a metric that measures the percentage of failed detections from all test frames. Tracking success is determined using predefined threshold values to ensure consistent and objective performance evaluation. A successful tracking was defined when the IoU and CLE between the predicted bounding box and the ground-truth annotation exceed 0.75 and less than 15 pixels, reflecting acceptable localization precision under the given video resolution. FR value measurement is performed on IoU and CLE results by producing IoU FR (%) and CLE FR (%) values.

The related metrics are calculated as below:

$$IoU = \frac{|B_p \cap B_g|}{|B_p \cup B_g|} \tag{16}$$

$$CLE\ (x,y) = \sqrt{\sum_{i=1}^{n} \quad (x_i - y_i)^2} \tag{17}$$

$$FPS = \frac{N_{frames}}{T_{total}} \tag{18}$$

$$FR = 1 - \frac{\sum \quad Success}{N} \tag{19}$$

$$Success_{IoU} = \{ 1, if\ IoU \geq 0.75\ 0, otherwise \tag{20}$$

$$Success_{CLE} = \{ 1, if\ CLE \geq 15\ 0, otherwise \tag{21}$$

3. **Results and Discussion**

The trials were performed on the Windows 11 operating system using Python environment with an AMD Athlon Silver 3050U processor (2 cores, 2 threads, base clock 2.3 GHz, up to 3.2 GHz) and 8 GB RAM. No discrete GPU was used; computations were executed on CPU.

3.1. Optimal Tracking Method Selection

Algorithm testing is carried out with various approaches including Kalman filter with frame skipping, DeepSORT, GOTURN and CSRT. The GOTURN and CSRT algorithms are SOT methods, this method uses a template matching approach through deep learning and discriminative correlation filters. The GOTURN and CSRT algorithms have high speed but often lose track and are unable to recover. Both methods are less ideal to apply in a dynamic testing environment, so YOLO detection has better advantages. In the case of SOT such as anxiety testing using elevated plus maze, there is no need for ID-based identification.

The DeepSORT algorithm was implemented with max IoU distance is 0.7 and max age of tracking is 200 frames. The DeepSORT algorithm based on MOT has an association feature that is not needed in this case. So the Kalman filter approach with frame skipping has a significant effect. In the application of frame skipping, the interval change affects the tracking accuracy based on the IoU and CLE failure rate metrics. Based on Table 1 related to the comparison of IoU and CLE failure rates against interval changes, there is an increase in error with higher interval values. The higher the interval used, the number of predictions made is much greater than YOLO detection. Prediction has a high error potential because it is based on position estimation from the previous frame, in contrast to YOLO detection which has a low error potential that detects the position in each frame directly.

Table 1. Average performance for each algorithm

Method	IoU FR	CLE FR	Mean IoU	Mean CLE	Time	FPS
KF-Backup	2.37 ± 2.53	0.55 ± 1.23	0.94 ± 0.02	1.48 ± 0.69	1.05 ± 0.18	9.78 ± 1.42

KF-Interval-3	3.82 ± 2.78	0.86 ± 1.20	0.92 ± 0.02	1.60 ± 0.39	0.45 ± 0.04	22.19 ± 1.73
KF-Interval-5	9.54 ± 4.66	1.22 ± 1.60	0.89 ± 0.03	2.35 ± 0.57	0.33 ± 0.05	31.30 ± 4.38
KF-Interval-10	26.60 ± 8.79	4.87 ± 2.54	0.81 ± 0.04	4.42 ± 1.15	0.25 ± 0.02	39.83 ± 3.65
UKF-Interval-3	3.06 ± 2.51	0.95 ± 1.16	0.93 ± 0.02	1.52 ± 0.39	0.47 ± 0.08	21.82 ± 3.32
UKF-Interval-5	8.80 ± 3.85	1.25 ± 1.58	0.89 ± 0.02	2.26 ± 0.56	0.33 ± 0.02	30.14 ± 2.05
UKF-Interval-10	26.03 ± 8.27	5.24 ± 2.18	0.82 ± 0.04	4.30 ± 1.11	0.26 ± 0.03	39.66 ± 4.07
AKF-Interval-3	3.24 ± 2.57	1.08 ± 1.18	0.92 ± 0.02	1.69 ± 0.47	0.45 ± 0.02	22.36 ± 1.10
AKF-Interval-5	8.97 ± 3.88	1.38 ± 1.61	0.89 ± 0.03	2.34 ± 0.59	0.33 ± 0.03	30.70 ± 2.70
AKF-Interval-10	26.02 ± 8.26	5.26 ± 2.18	0.82 ± 0.04	4.31 ± 1.09	0.25 ± 0.03	41.37 ± 4.64
DEEPSORT	8.73 ± 1.81	0.08 ± 0.07	0.88 ± 0.01	1.45 ± 0.31	1.06 ± 0.06	9.48 ± 0.54
GOTURN	99.99 ± 0.02	99.09 ± 1.04	0.01 ± 0.00	217.10 ± 85.64	0.70 ± 0.19	16.37 ± 8.11
CSR-DCF	98.35 ± 1.87	41.30 ± 36.57	0.37 ± 0.19	50.91 ± 69.74	0.55 ± 0.09	18.59 ± 2.51

IoU FR threshold : 0.75, CLE FR threshold : 15

Determination of the most optimal method in the case of anxiety analysis using EPM is determined using the weighted scoring method (WSM) based on five performance metrics: IoU FR, CLE FR, Mean IoU, Mean CLE and FPS (see fig. 5-7). The values are normalized on a scale of 0 to 1 based on measurements with the lowest and highest values. When evaluating performance, we consider that certain metrics are better when their values are higher (FPS, Mean IoU), while others indicate superior performance when their values are lower (IoU FR, CLE FR, Mean CLE).

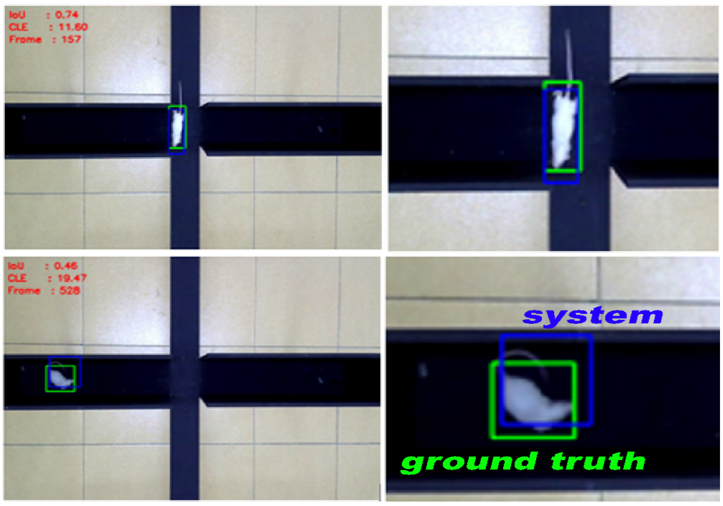

Figure 5. Performance metrics measurement between ground truth and system

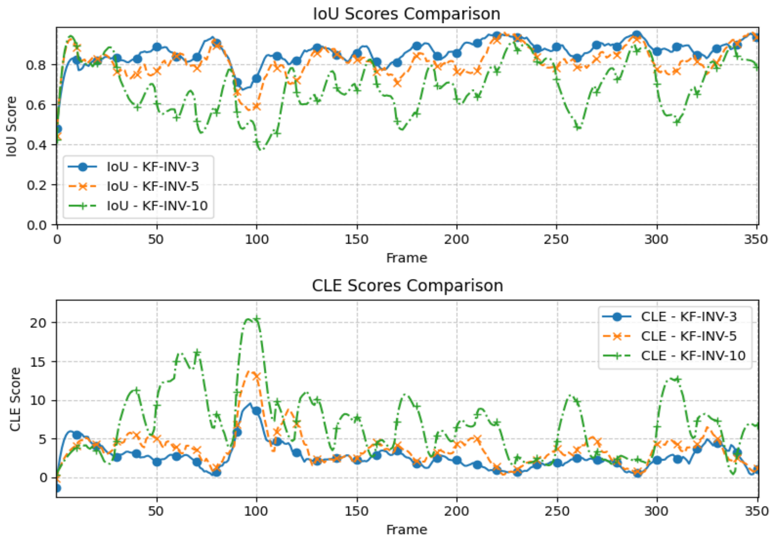

Figure 6. Comparison of Kalman filter implementation for target tracking in different intervals

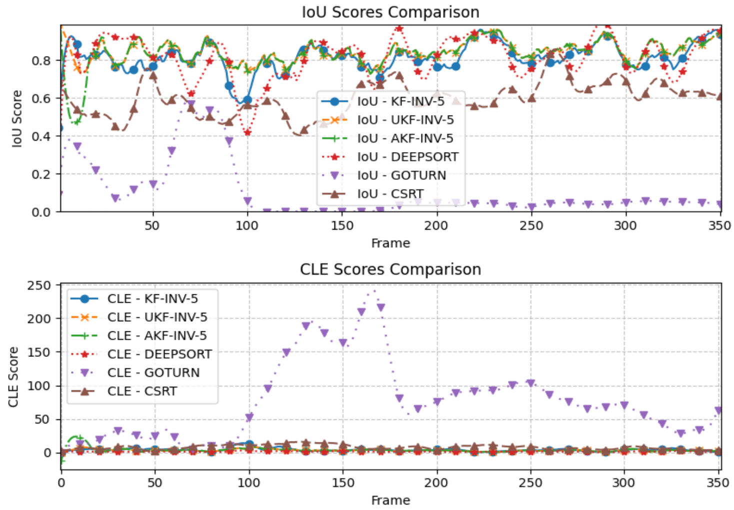

Figure 7. Comparison of IoU and CLE of each tracking method

Equal weight is implemented with an equal weight of 0.2 for each metric. The calculation results using WSM with equal weights can be seen in Table 2, the Kalman filter approach with intervals achieves the most satisfactory performance compared to the DeepSORT, GOTURN and CSRT algorithms. This indicates that combination of frame skipping and kalman filter successfully increase overall performance of the system, increase the FPS by reducing time inference from YOLOv8 detection and maintain tracking accuracy. The most optimal method based on the highest score is the AKF-INV-10 method or the adaptive Kalman filter approach with an interval of 10 frames with a value of 0.9127.

Table 2. Weighted scoring method with equal weights

Method	IoU FR	CLE FR	Mean IoU	Mean CLE	Time	FPS	Score
KF-Backup	1.000	0.995	1.000	1.000	0.012	0.009	0.8009
KF-Interval-3	0.985	0.992	0.984	0.999	0.744	0.399	0.8718
KF-Interval-5	0.927	0.988	0.946	0.996	0.900	0.684	0.9081
KF-Interval-10	0.752	0.952	0.868	0.986	0.990	0.952	0.9019
UKF-Interval-3	0.993	0.991	0.989	1.000	0.723	0.387	0.8720
UKF-Interval-5	0.934	0.988	0.951	0.996	0.891	0.648	0.9035
UKF-Interval-10	0.758	0.948	0.872	0.987	0.988	0.947	0.9021
AKF-Interval-3	0.991	0.990	0.986	0.999	0.750	0.404	0.8739
AKF-Interval-5	0.932	0.987	0.949	0.996	0.898	0.665	0.9060
AKF-Interval-10	0.758	0.948	0.872	0.987	1.000	1.000	**0.9127**
DEEPSORT	0.935	1.000	0.944	1.000	0.000	0.000	0.7757
GOTURN	0.000	0.000	0.000	0.000	0.436	0.216	0.0432
CSR-DCF	0.017	0.584	0.389	0.771	0.625	0.286	0.4091

Applying equal weights (0.2) across the five metrics may not yield an optimal solution. While statistically neutral, this approach fails to account for specific system objectives and the inherent trade-offs between the metrics in a real-world context. Considering the objectives and impacts on the system, different weight configurations are used: IoU FR (0.3), CLE FR (0.3), Mean IoU (0.15), Mean CLE (0.15) and FPS (0.1). Referring to the results of the optimal method calculation with custom weights in Table 3, the highest value obtained is 0.9341 with the AKF-INV-05 and KF-INV-05 method. Both KF-INV-5 and AKF-INV-5 achieve identical WSM scores, indicating comparable overall performance. As the primary objective of this study, AKF-INV-5 was selected due to its lower variability across test video.

Table 3. Weighted scoring method with custom weights

Method	IoU FR	CLE FR	Mean IoU	Mean CLE	Time	FPS	Score
KF-Backup	1.000	0.995	1.000	1.000	0.012	0.009	0.8995
KF-Interval-3	0.985	0.992	0.984	0.999	0.744	0.399	0.9305
KF-Interval-5	0.927	0.988	0.946	0.996	0.900	0.684	**0.9341**
KF-Interval-10	0.752	0.952	0.868	0.986	0.990	0.952	0.8843
UKF-Interval-3	0.993	0.991	0.989	1.000	0.723	0.387	0.9323
UKF-Interval-5	0.934	0.988	0.951	0.996	0.891	0.648	0.9336
UKF-Interval-10	0.758	0.948	0.872	0.987	0.988	0.947	0.8850
AKF-Interval-3	0.991	0.990	0.986	0.999	0.750	0.404	0.9324
AKF-Interval-5	0.932	0.987	0.949	0.996	0.898	0.665	**0.9341**
AKF-Interval-10	0.758	0.948	0.872	0.987	1.000	1.000	0.8904
DEEPSORT	0.935	1.000	0.944	1.000	0.000	0.000	0.8720
GOTURN	0.000	0.000	0.000	0.000	0.436	0.216	0.0216
CSR-DCF	0.017	0.584	0.389	0.771	0.625	0.286	0.3826

Based on the comparison between the optimal method on equal weights and custom weights configurations in Table 4, the optimal method by prioritizing certain aspects of the system (custom weights) produces a method that is better in terms of accuracy and meets the image speed to run in real time based on the minimum FPS requirement of 30 fps or close to that based on the video or camera fps used to minimize information loss.

Table 4. Comparison of equal weights and custom weights

Metriks	**AKF-Interval-5** ***(custom weight)***	**AKF-Interval-10** ***(equal weight)***	**Explanation**
IoU FR (%)	8.97	26.02	AKF-INV-5 is better
CLE FR (%)	1.38	5.26	AKF-INV-5 is more accurate

Mean IoU	0.89	0.82	AKF-INV-5 is more consistent
Mean CLE	2.34	4.31	AKF-INV-5 is more consistent
FPS	30.70	41.37	both are realtime

3.2. Implementation

Implementation of selected method is needed to know its performance in real condition environment. The software implementation was carried out as a monitoring, tracking display, and as processing the results of analysis of the movement of mice in the research elevated plus maze (see Fig. 8). To enhance the user experience and ensure efficient data management, a Graphical User Interface (GUI) was developed using PyQt, as presented in Figure 8. The system performance in detecting, tracking objects to visualize the results is evaluated by measuring the fps value on two types of video data, files and cameras for each interval. The measured speed value consists of two types, system process speed and tracking process speed (track-only) (see Fig. 9).

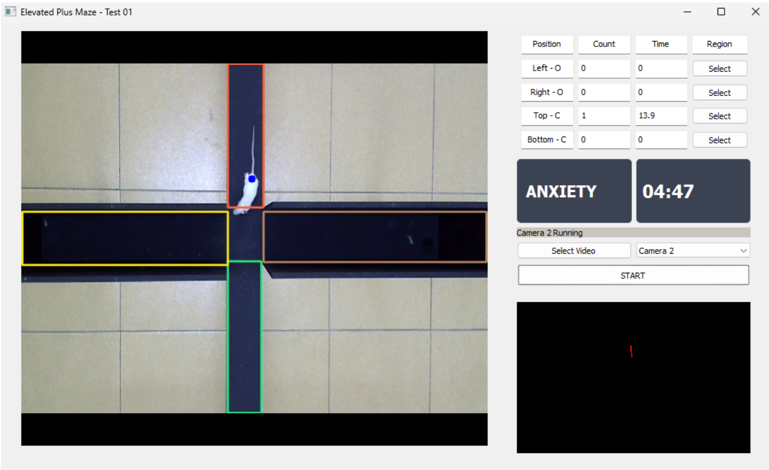

Figure 8. View of the real-time analysis process

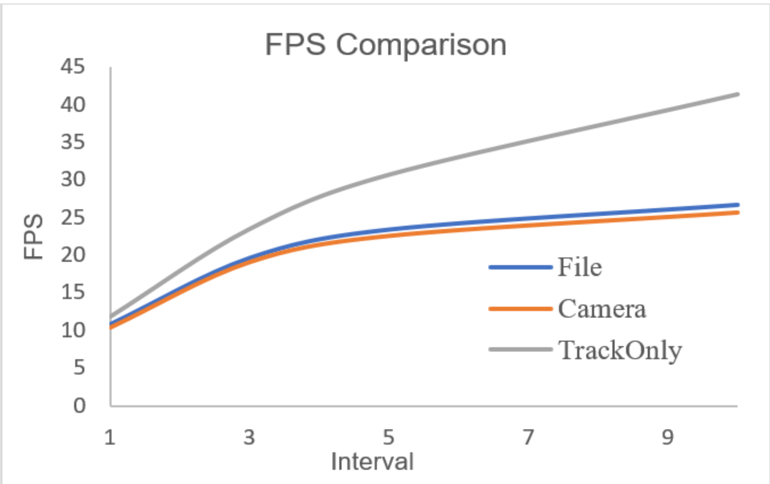

Figure 9. FPS comparison of the deployed system

Based on Table 5, there is an increase in fps value against the change of method from Full-YOLO or interval 1 to AKF-INV-5 with an increase in fps in track-only conditions of 158.81% from 11.86 fps to 30.70 fps. The application of AKF-INV-5 into the system resulted in an increase in fps of 116% from 10.44 fps to 22.55 fps. This decrease occurred because the system had an additional burden in displaying the GUI and other components outside the tracking system. Thus, the application of the Kalman filter tracking algorithm with frame skipping has a significant impact on the speed of the process, especially in the case of the mouse monitoring system with an increase in system fps reaching 116%.

Table 5. System speed performance

Interval	FPS			Improvement (%)	
	File	Cam	Track Only	System	Track Only
AKF- Backup	10.88	10.44	11.86	-	-
AKF-Interval-3	19.67	19.09	23.49	82.85	98.10
AKF-Interval-5	23.43	22.55	30.70	116.00	158.81
AKF-Interval-10	26.71	25.64	41.37	145.59	248.79

The results indicate that the integration of frame skipping reduces inference time, while the use of Kalman-based filtering maintains tracking continuity during skipped frames. Despite the improvement in inference time, the use of frame skipping introduces a trade-off between computational cost and accuracy.

In calculation of WSM with custom weight, we found that classic Kalman filters with fixed covariance value have comparable performance with adaptive Kalman filters that have adaptive covariance value. But, the main difference is coming from its accuracy and computational cost. Classic Kalman filters have lower computational cost for each frame due to its fixed-parameter structure, which reduces complexity and per-frame computation. Although adaptive Kalman filters have higher spatial accuracy, the improvement is limited because the detector output remains unchanged, indicating diminishing returns from adaptive filters under stable tracking conditions.

The advantage of adaptive Kalman filter may be diminished when frame skipping is done at intervals of five frames since the motion dynamic between observed frames stays comparatively smooth. Greater speed gains are achieved with longer frame-skipping intervals, although accuracy may be marginally impacted by the increased prediction uncertainty. Higher frame-skipping intervals enhance FPS but tend to raise CLE, IoU, and failure rate, especially for rapid motion patterns. This trade-off is represented in the evaluation metrics.

It should be mentioned that the suggested method does not explicitly simulate complex non-linear dynamics and instead depends on linear motion assumptions, which could limit effectiveness in situations involving extremely abrupt motion. During testing with different frame skip intervals, there were areas where the CLE or IoU deviation values were significantly higher. This occurs when the target's direction and speed change during the frame skip period. This limitation cannot be adequately addressed using a constant-velocity assumption. Therefore, future implementations must integrate complex non-linear dynamic models to manage rapid changes in object speed and direction, particularly during skipped frames.

In biomedical experiments such as the elevated plus maze, automating the measurement of anxiety behavior can reduce inter-observer bias because all parameters are consistently extracted from the same spatial data, rather than from subjective human interpretation. This approach ensures uniform evaluation standards across sessions and between subjects, thereby improving the reproducibility and reliability of experimental results.

Accurate behavioral interpretation strongly depends on the quality of locomotion tracking. Improved tracking stability and reduced computational cost allow the recorded subject position to better represent real movement, thereby reducing errors in zone classification. This leads to more reliable anxiety analysis while enabling the system to operate on low-cost hardware, making it more accessible for laboratories with limited computational resources.

4. **Conclusion**

This research successfully demonstrates that computational cost can be significantly reduced with minimal degradation in tracking accuracy by integrating frame skipping into a YOLO-based detection framework. By reducing the frequency of object detection, the computational load per frame is substantially lowered, resulting in an FPS enhancement up to 116% at intervals of 5 frames. The continuity of object trajectories is preserved through Kalman filter prediction, which compensates for skipped detections by propagating motion estimates across frames. The YOLO with adaptive Kalman filter (AKF) configuration maintains acceptable spatial and positional reliability, with failure rates limited to 8.97% and 1.38%, respectively, according to quantitative evaluation based on Intersection over Union (IoU) metrics, failure rate, and processing speed. These findings imply that vital tracking stability is not sacrificed in order to achieve the reported speed advantage.

This system opens up great opportunities in the application of real-time automatic analysis using computer vision. Future research should focus on scaling the system toward multi-animal and high-throughput behavioral experiments to support larger cohorts and more statistically robust biomedical studies, enhancing detection accuracy and speed using improved architectures or diverse, representative datasets. Future enhancements should explore state-of-the-art object detection architectures, such as YOLOv10, to further optimize detection accuracy and processing speed [48]. Testing under varied conditions and movement speeds is essential for real-world reliability. Additionally, deeper behavioral analysis—such as classifying freezing, grooming, rearing, and exploration—should be developed through advanced feature extraction.

Declaration of AI and AI assisted technologies in the writing process

During the preparation of this work the author(s) used ChatGPT and other in order to find references and write the article. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research is funded by the Directorate of Research and Community Service, Directorate General of Research and Development of the Ministry of Higher Education, Science, and Technology with contract number: 105/C3/DT.05.00/PL/2025 and the research assignment agreement letter number of LPPM UNS: 1186.1/UN27.22/PT.01.03/2025.

References

[1] Z. Nikolay, K. Nikolay, X. Gao, Q. Wei Li, G. Peng Mi, and Y. Xiang Huang, “Design and testing of novel seed miss prevention system for single seed precision metering devices,” *Comput. Electron. Agric.*, vol. 198, no. May, p. 107048, 2022, doi:

https://doi.org/10.1016/j.compag.2022.107048

[2] C. Ari, D. P. D'Agostino, D. M. Diamond, M. Kindy, C. Park, and Z. Kovács, "Elevated plus maze test combined with video tracking software to investigate the anxiolytic effect of exogenous ketogenic supplements," *J. Vis. Exp.*, vol. 2019, no. 143, pp. 1–10, 2019, doi: https://doi.org/10.3791/58396

[3] P. González-Gaspar *et al.*, "Analixity: An open source, low-cost analysis system for the elevated plus maze test, based on computer vision techniques," *Behav. Processes*, vol. 193, no. November, 2021, doi: https://doi.org/10.1016/j.beproc.2021.104539.

[4] Z. Liu *et al.*, "Development and experimental validation of a system for agricultural grain unloading-on-the-go," *Comput. Electron. Agric.*, vol. 198, no. August 2021, p. 107005, 2022, doi: https://doi.org/10.1016/j.compag.2022.107005.

[5] A. Auguste, W. Kaddah, M. Elbouz, G. Oudinet, and A. Alfalou, "Behavioral analysis and individual tracking based on kalman filter: Application in an urban environment," *Sensors*, vol. 21, no. 21, 2021, doi: https://doi.org/10.3390/s21217234.

[6] G. Sun *et al.*, "DeepBhvTracking: A Novel Behavior Tracking Method for Laboratory Animals Based on Deep Learning," *Front. Behav. Neurosci.*, vol. 15, no. October, pp. 1–10, 2021, doi: https://doi.org/10.3389/fnbeh.2021.750894

[7] M. Sourioux *et al.*, "3-D motion capture for long-term tracking of spontaneous locomotor behaviors and circadian sleep/wake rhythms in mouse," *J. Neurosci. Methods*, vol. 295, no. November 2017, pp. 51–57, 2018, doi: https://doi.org/10.1016/j.jneumeth.2017.11.016.

[8] T. Fong, H. Hu, P. Gupta, B. Jury, and T. H. Murphy, "PyMouseTracks: Flexible Computer Vision and RFID-Based System for Multiple Mouse Tracking and Behavioral Assessment," *eNeuro*, vol. 10, no. 5, pp. 1–17, 2023, doi: https://doi.org/10.1523/ENEURO.0127-22.2023.

[9] F. Wu, J. Duan, P. Ai, Z. Chen, Z. Yang, and X. Zou, "Rachis detection and three-dimensional localization of cut off point for vision-based banana robot," *Comput. Electron. Agric.*, vol. 198, no. May, p. 107079, 2022, doi: https://doi.org/10.1016/j.compag.2022.107079.

[10] M. Chen, C. Jin, Y. Ni, T. Yang, and G. Zhang, "Online field performance evaluation system of a grain combine harvester," *Comput. Electron. Agric.*, vol. 198, no. May, p. 107047, 2022, doi: https://doi.org/10.1016/j.compag.2022.107047.

[11] G. Kaul, J. McDevitt, J. Johnson, and A. Eban-Rothschild, "DAMM for the detection and tracking of multiple animals within complex social and environmental settings," *Sci. Rep.*, vol. 14, no. 1, pp. 1–15, 2024, doi: https://doi.org/10.1038/s41598-024-72367-2.

[12] A. Affan, H. A. Nasir, T. Manzoor, and A. Muhammad, "Mobile sensing for estimation of hydro-dynamic parameters for minimally gauged open channels," *Comput. Electron. Agric.*, vol. 198, no. May, p. 107072, 2022, doi: https://doi.org/10.1016/j.compag.2022.107072.

[13] U. F. Rahim, T. Utsumi, and H. Mineno, "Deep learning-based accurate grapevine inflorescence and flower quantification in unstructured vineyard images acquired using a mobile sensing platform," *Comput. Electron. Agric.*, vol. 198, no. May, p. 107088, 2022, doi: https://doi.org/10.1016/j.compag.2022.107088.

[14] D. Held, S. Thrun, and S. Savarese, "Learning to Track at 100 FPS with Deep," *Comput. Vis. – ECCV 2016 Lect. Notes Comput. Sci.*, pp. 749–765, 2016, [Online]. Available: http://davheld.github.io/GOTURN/GOTURN.html

[15] R. G. Freitas, F. R. S. Pereira, A. A. Dos Reis, P. S. G. Magalhães, G. K. D. A. Figueiredo, and L. R. do Amaral, "Estimating pasture aboveground biomass under an integrated crop-livestock

system based on spectral and texture measures derived from UAV images," *Comput. Electron. Agric.*, vol. 198, no. May, 2022, doi: https://doi.org/10.1016/j.compag.2022.107122.

[16] S. Isik and G. Unal, "Open-source software for automated rodent behavioral analysis," *Front. Neurosci.*, vol. 17, no. April, 2023, doi: https://doi.org/10.3389/fnins.2023.1149027.

[17] C. Chen, J. Lu, M. Zhou, J. Yi, M. Liao, and Z. Gao, "A YOLOv3-based computer vision system for identification of tea buds and the picking point," *Comput. Electron. Agric.*, vol. 198, no. March, p. 107116, 2022, doi: https://doi.org/10.1016/j.compag.2022.107116

[18] S. Zhao *et al.*, "A lightweight dead fish detection method based on deformable convolution and YOLOV4," *Comput. Electron. Agric.*, vol. 198, no. May, p. 107098, 2022, doi: https://doi.org/10.1016/j.compag.2022.107098.

[19] Y. Wang, G. Yan, Q. Meng, T. Yao, J. Han, and B. Zhang, "DSE-YOLO: Detail semantics enhancement YOLO for multi-stage strawberry detection," *Comput. Electron. Agric.*, vol. 198, no. March, p. 107057, 2022, doi: https://doi.org/10.1016/j.compag.2022.107057.

[20] Z. Yu, Y. Li, Z. Liang, and Z. Tang, "Development of single measuring point overall balancing method based on multi-cylinder dynamic balance detection system," *Comput. Electron. Agric.*, vol. 198, no. June, p. 106968, 2022, doi: https://doi.org/10.1016/j.compag.2022.106968.

[21] A. Du *et al.*, "Automatic livestock body measurement based on keypoint detection with multiple depth cameras," *Comput. Electron. Agric.*, vol. 198, no. May, 2022, doi: https://doi.org/10.1016/j.compag.2022.107059.

[22] C. N. Vong, L. S. Conway, A. Feng, J. Zhou, N. R. Kitchen, and K. A. Sudduth, "Corn emergence uniformity estimation and mapping using UAV imagery and deep learning," *Comput. Electron. Agric.*, vol. 198, no. May, p. 107008, 2022, doi:https:/doi.org/10.1016/j.compag.2022.107008.

[23] J. Zhao *et al.*, "A deep learning method for oriented and small wheat spike detection (OSWSDet) in UAV images," *Comput. Electron. Agric.*, vol. 198, no. May, p. 107087, 2022, doi: https://doi.org/10.1016/j.compag.2022.107087.

[24] N. Sakamoto, H. Kakeno, N. Ozaki, Y. Miyazaki, K. Kobayashi, and T. Murata, "Marker-less tracking system for multiple mice using Mask R-CNN," *Front. Behav. Neurosci.*, vol. 16, no. January, pp. 1–10, 2023, doi: https://doi.org/10.3389/fnbeh.2022.1086242.

[25] T. Del Rosario Hernández, N. R. Joshi, S. V. Gore, J. A. Kreiling, and R. Creton, "An 8-cage imaging system for automated analyses of mouse behavior," *Sci. Rep.*, vol. 13, no. 1, pp. 1–14, 2023, doi: https://doi.org/10.1038/s41598-023-35322-1.

[26] R. Sholehurrohman, M. R. Habibi, I. S. Ilman, R. Taufiq, and M. Muhaqiqin, "Analisis Metode Kalman Filter, Particle Filter dan Correlation Filter Untuk Pelacakan Objek," *Komputika J. Sist. Komput.*, vol. 12, no. 2, pp. 21–28, 2023, doi: https://doi.org/10.34010/komputika.v12i2.9567.

[27] C. Guo, Y. Chen, C. Ma, S. Hao, and J. Song, "A Survey on AI-Driven Mouse Behavior Analysis Applications and Solutions," *Bioengineering*, vol. 11, no. 11, pp. 1–39, 2024, doi: https://doi.org/10.3390/bioengineering11111121.

[28] Y. Li, W. Gao, J. Jia, S. Tao, and Y. Ren, "Developing and evaluating the feasibility of a new spatiotemporal fusion framework to improve remote sensing reflectance and dynamic LAI monitoring," *Comput. Electron. Agric.*, vol. 198, no. 17, p. 107037, 2022, doi: https://doi.org/10.1016/j.compag.2022.107037.

[29] T. D. Pereira *et al.*, "SLEAP: A deep learning system for multi-animal pose tracking," *Nat. Methods*, vol. 19, no. 4, pp. 486–495, 2022, doi: https://doi.org/10.1038/s41592-022-01426-1.

[30] Y. Bai, Y. Fang, B. Zhang, and S. Fan, "Model robustness in estimation of blueberry SSC using

NIRS," *Comput. Electron. Agric.*, vol. 198, no. May, p. 107073, 2022, doi: https://doi.org/10.1016/j.compag.2022.107073.

[31] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," *Proc. - Int. Conf. Image Process. ICIP*, vol. 2017-Septe, pp. 3645–3649, 2017, doi: https://doi.org/10.1109/ICIP.2017.8296962 .

[32] H. Wang *et al.*, "Fast detection of cannibalism behavior of juvenile fish based on deep learning," *Comput. Electron. Agric.*, vol. 198, no. May, p. 107033, 2022, doi: https://doi.org/10.1016/j.compag.2022.107033.

[33] J. Lee and K. il Hwang, "YOLO with adaptive frame control for real-time object detection applications," *Multimed. Tools Appl.*, vol. 81, no. 25, pp. 36375–36396, 2022, doi: https://doi.org/10.1007/s11042-021-11480-0.

[34] A. Stratmann, A. Kashev, M. Rice, and M. J. Toscano, "A first approach to investigate characteristics of piling behaviour in individual laying hens," *Appl. Anim. Behav. Sci.*, vol. 299, no. February, p. 106967, 2026, doi: https://doi.org/10.1016/j.applanim.2026.106967.

[35] Y. Zhang, W. Zhang, J. Yu, L. He, J. Chen, and Y. He, "Complete and accurate holly fruits counting using YOLOX object detection," *Comput. Electron. Agric.*, vol. 198, no. May, p. 107062, 2022, doi: https://doi.org/10.1016/j.compag.2022.107062.

[37] R. Hardianto, "Implementation of Object Detection With You Only Look Once Algorithm in Limited Face-To-Face Times in Pandemic," *Journal of Applied Engineering and Technological Science (JAETS)*, vol. 4, no. 1, pp. 400-404, 2022, doi: https://doi.org/10.37385/jaets.v4i1.1161.

[38] B. Soewito, "Multi-Object Detection Using YOLOv7 Object Detection Algorithm on Mobile Device," *Journal of Applied Engineering and Technological Science (JAETS)*, vol. 5, no. 1, pp. 305-320, 2023, doi: https://doi.org/10.37385/jaets.v5i1.3207.

[36] D. E. F. Adz Zahra, S. Ibrahim, J. Hariyono, F. Rahutomo, S. Pramono, M.H. Ibrahim, M. E. Sulistyono and N. Wiyono, "Video Processing and Monitoring Animal Movement Models to Measure Learning and Memory in the T Maze Spontaneous Alternation Test with YOLO Implementation," in *2024 FORTEI-International Conference on Electrical Engineering (FORTEI-ICEE)*, Badung, Indonesia, 2024, pp. 41-46.

[37] S. Ibrahim,"A comprehensive review on intelligent surveillance systems", Communications in Science and Technology, 1 (1) (2016)

[38] S. Bala, "Improving Kidney Stone Detection with YOLOV10 and Channel Attention Mechanisms in Medical Imaging", The Journal of Electronics, Electromedical Engineering, and Medical Informatics, vol. 7, no. 3, pp. 951-963, Jul. 2025.