



ARTICLE

An Efficient Battery-Powered Wild Animal Tracking Device with a PTZ Camera

Shogo Semba^{*}, Hiroshi Saito, Yoichi Tomioka and Yukihide Kohira

School of Computer Science and Engineering, The University of Aizu, Aizu-Wakamatsu, Japan

^{*}Corresponding Author: Shogo Semba. Email: shogo-s@u-aizu.ac.jp

Received: 08 May 2026; Accepted: 09 September 2026; Published: 22 October 2026

ABSTRACT: In this paper, we propose an efficient battery-powered wild animal tracking device with a Pan-Tilt-Zoom (PTZ) camera. The proposed device integrates object detection, tracking, and PTZ camera control to enable autonomous wild animal tracking in outdoor environments without stable power supplies. To improve efficiency, it uses different object detection models for before and during tracking, performs parallel object detection, and applies dynamic Pan-Tilt (PT) control. In the experiments, we implemented the proposed device and conducted field tests. The use of different object detection models for before and during tracking improved the detection success rate before tracking by 29% and reduced the energy consumption during tracking by 64%. The parallel object detection during tracking improved frames per second (FPS) by 62% and FPS per watt (FPS/W) by 49% compared with a non-parallel approach. The dynamic PT control reduced root mean square error (RMSE) compared with a fixed control method (22 vs. 79–163 pixels in the x -direction and 48 vs. 31–65 pixels in the y -direction). In addition, the field tests showed that the proposed device tracked wild animals with an 89% success rate in maintaining the target within the field of view after PTZ control. The power consumption during tracking was 10.14 W during the daytime and 13.47 W during the nighttime, while the idle power consumption was 6.95 and 7.06 W, respectively, demonstrating that the proposed device can operate using a solar panel and a battery.

KEYWORDS: Animal tracking; PTZ camera; deep learning; energy-efficient; edge computing

1 Introduction

Japan has faced serious problems such as human injuries and crop damage caused by wild animals. According to Japanese government reports, in fiscal year (FY) 2023, a total of 304 human injuries were caused by bears and wild boars [1], and the cost of crop damage caused by bears, wild boars, and deer was estimated at 11,326 million Japanese yen [2].

To monitor the movement of wild animals, tag-based devices using technologies such as the Global Positioning System, Radio Frequency Identification, and Bluetooth Low Energy have been developed in [3–5]. These devices are attached to animals to record their locations and detection histories. However, attaching devices is difficult and dangerous, as it requires capturing wild animals. In addition, device size and battery life restrict which animals can be tagged.

Vision-based detection and tracking approaches using deep learning have been presented in [6–8]. In [6], a battery-powered detection device was developed to detect wild animals in outdoor environments. However, it performs only detection. In [7,8], deep learning-based tracking systems assuming fixed camera viewpoints were developed. Animals are tracked by detecting their positions and poses and associating

the results between frames. However, tracking is limited to the camera's field of view because the camera's direction is fixed.

Vision-based wide-area tracking systems using Pan-Tilt-Zoom (PTZ) cameras have been presented in [9–14]. These systems typically rely on continuous video streaming and object detection, and are mainly designed for environments with stable power supplies. However, for applications such as preventing human injuries and crop damage, early detection and wide-area tracking are required in outdoor environments without stable power supplies.

In this paper, we propose an efficient battery-powered wild animal tracking device using a PTZ camera and deep learning. The proposed device tracks wild animals through object detection with You Only Look Once (YOLO) [15], tracking with Deep Simple Online and Realtime Tracking (DeepSORT) [16], and PTZ camera control. To enable operation in outdoor environments without stable power supplies, it is powered by a battery and a solar panel, and performs object detection only when motion is detected to avoid unnecessary detection processes. To achieve efficient wild animal tracking, it uses different object detection models for before and during tracking, improving detection reliability while reducing unnecessary operations. In addition, it executes object detection in parallel during tracking to improve frame rate and performs dynamic Pan-Tilt (PT) control to enhance responsiveness to animal movements.

This paper focuses on the integration of object detection, tracking, PTZ camera control, and power-efficient operation into a battery-powered autonomous wild animal tracking device. The proposed device enables autonomous wild animal tracking in outdoor environments without stable power supplies. The contributions of this paper are as follows:

- A battery-powered device for automatic wild animal tracking in outdoor environments without stable power supplies
- A two-stage detection method using different object detection models before and during tracking, achieving a 29% higher detection success rate before tracking and 64% lower energy consumption during tracking
- A parallel object detection method during tracking, improving frames per second (FPS) by 63% and FPS per watt (FPS/W) by 49%
- A dynamic PT control method reducing root mean square error (RMSE) compared with a fixed control method

The rest of this paper is organized as follows. [Section 2](#) describes related work. [Section 3](#) describes the proposed device. [Section 4](#) describes the experimental results. [Section 5](#) describes the discussion. Finally, [Section 6](#) describes the conclusion and future work.

2 Related Work

PTZ camera-based tracking devices for indoor environments have been presented in [9,10]. In [9], the camera periodically rotates among preset positions to detect cows using YOLO in livestock barns. In [10], the target object and floor grid vertices are detected using YOLO, and the target's world coordinates are estimated to control the PTZ camera and keep the target centered in the image. These devices target indoor environments, where objects are consistently present and stable power supplies are available.

For outdoor environments, PTZ camera-based tracking devices have been presented in [11–14]. In [11], tracking is performed using DeepSORT, while the PTZ camera is controlled to keep the target in view. In [12–14], reinforcement learning is used to determine PTZ actions based on raw images or object detection results. These methods assume outdoor environments with stable power supplies that enable continuous video streaming, object detection, and PTZ control.

Compared with [9–14], the proposed device is powered by a battery, enabling wild animal tracking in outdoor environments without stable power supplies. Since wild animal appearances are unpredictable, periodic camera rotation and continuous video streaming with object detection are inefficient, increasing energy consumption. In contrast, the proposed device reduces unnecessary operations by activating tracking only when motion is detected using motion sensors and frame differencing, improving battery life.

In this work, the proposed device is based on our preliminary conference version presented in [17]. It tracks animals by sequentially performing object detection with YOLO, tracking with DeepSORT, and controlling the PTZ camera with fixed values. However, this sequential structure limits the frame rate and reduces PTZ responsiveness. In addition, fixed PTZ control values do not adapt well to wild animal movements. Moreover, the same object detection model is used before and during tracking despite the different requirements in accuracy and processing speed.

To address these issues, this paper proposes an efficient battery-powered wild animal tracking device with three optimization methods. The proposed device improves tracking efficiency by performing object detection in parallel and dynamically controlling the PTZ camera, enabling a higher frame rate and better adaptation to animal movements. In addition, it uses different object detection models for before and during tracking to meet the different requirements, thereby improving energy efficiency.

3 Proposed Tracking Device

We propose an efficient battery-powered wild animal tracking device using a PTZ camera and deep learning. The proposed device is based on our preliminary conference version presented in [17], with additional optimization methods that improve tracking and energy efficiency. In this section, we describe its overview, device structure, operational flow, optimization methods, and implementation.

3.1 Overview

Fig. 1 shows the overview of the proposed device. To detect and track wild animals, it is placed near farmland, forests, rivers, etc. Since stable power supplies are unavailable, it is powered by a battery and a solar panel.

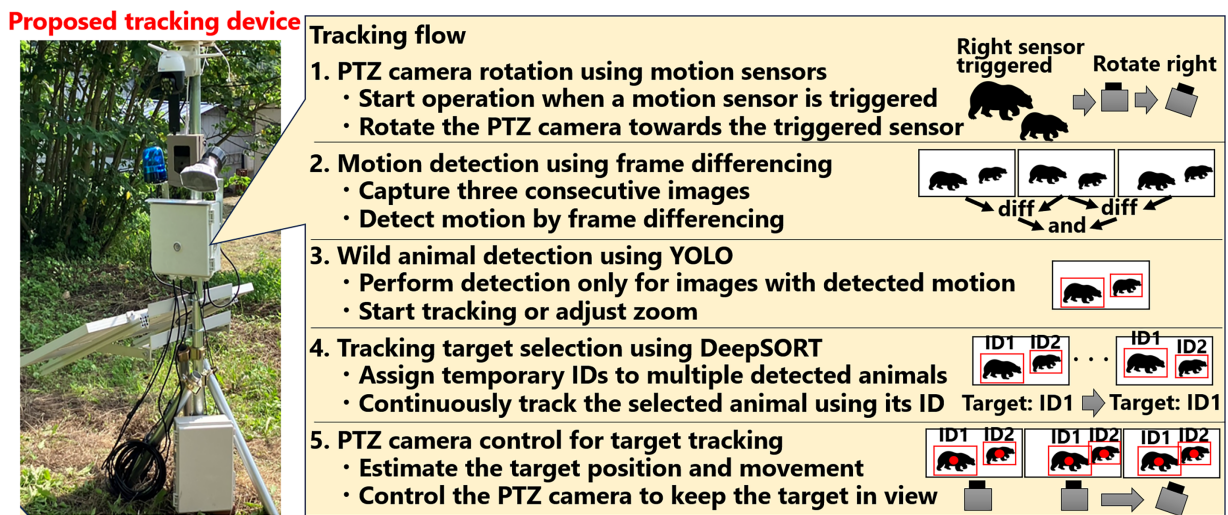


Figure 1: Overview of the proposed tracking device.

The proposed device tracks wild animals through PTZ camera rotation, motion detection, animal detection, target selection, and PTZ camera control. When animals are detected, it generates sound and light alerts and sends detection results to a server, which notifies users by email.

PTZ camera rotation using motion sensors. The proposed device rotates the PTZ camera towards the triggered motion sensor, enabling wide-area detection without periodic rotation as in [9] and reducing unnecessary power consumption. When multiple sensors are triggered, the PTZ camera rotates in the direction with the highest predefined priority.

Motion detection using frame differencing. The proposed device performs frame differencing as a preprocessing step for object detection using YOLO. Unlike [9–14] that continuously perform object detection, this method reduces unnecessary YOLO inference caused by false triggers from motion sensors.

Wild animal detection using YOLO. After motion is detected, the proposed device detects wild animals using YOLO. When wild animals are detected, it tracks them by referring to the detection results. When wild animals are not detected, it performs zoom operations and reattempts detection to improve accuracy by adjusting the size of objects.

Tracking target selection using DeepSORT. When multiple wild animals are detected, the proposed device selects a single target to simplify PTZ camera control. Since animals tend to move in groups, it uses DeepSORT to assign temporary identifiers (IDs) to multiple detected animals and selects an appropriate tracking target. Once selected, it continuously tracks the target using the target's ID, enabling stable PTZ camera control.

PTZ camera control. To track wild animals, the proposed device controls the PTZ camera based on the target's position and movement between frames. This control keeps the target within the field of view and enables stable tracking.

3.2 Structure

Fig. 2 shows the structure of the proposed device. Black arrows represent power supply signals, and blue arrows represent data or control signals.

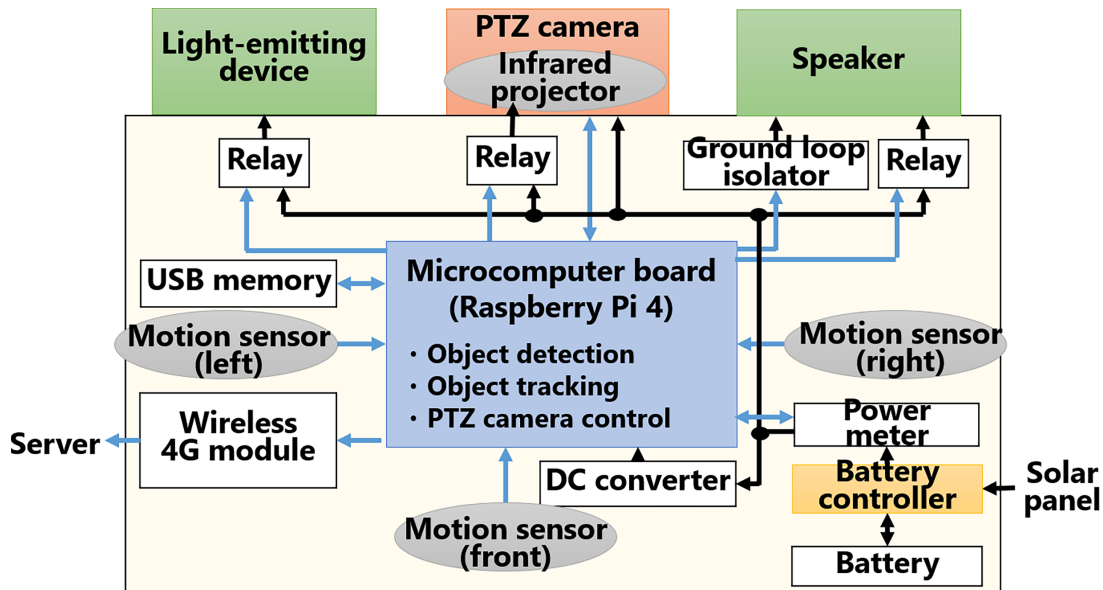


Figure 2: Structure of the proposed tracking device.

The proposed device consists of a microcomputer board and peripheral components, including motion sensors, a PTZ camera with an infrared projector, a light-emitting device, a speaker, relay modules, a Universal Serial Bus (USB) memory, and a wireless 4G module. An additional ground loop isolator is introduced to minimize electrical noise, enabling the device to generate clear alert sounds that improve audibility for nearby people and enhance the effectiveness of scaring wild animals. In addition, an additional power meter is introduced to measure current, voltage, and power consumption, enabling estimation of the remaining battery power.

Each component of the proposed device has a role in efficient wild animal detection and tracking. The microcomputer board performs object detection, tracking, and PTZ control. The motion sensors detect movement around the device (e.g., front, right, and left). The PTZ camera captures images. The light-emitting device and speaker generate alerts when animals are detected. The relay modules control power to peripheral devices to reduce unnecessary power consumption. The power meter measures power usage. The USB memory stores detection results. The wireless 4G module transmits detection results and voltage data to a server.

The proposed device is powered by a battery and a solar panel, making power efficiency essential for outdoor operation without stable power supplies. To reduce power consumption, the infrared projector, speaker, and light-emitting device are activated only when needed using relay modules. In contrast, the PTZ camera remains continuously powered due to its long initialization and connection time with the microcomputer board.

3.3 Operation Flow

The proposed device starts operating when a motion sensor is triggered, as shown in Fig. 3. It checks the shooting mode of the PTZ camera and turns on the infrared projector in night mode. Then, it rotates the PTZ camera towards the triggered sensor. Next, it reads multiple frames and applies frame differencing to detect motion. Three consecutive frames are used because the proposed device uses the three-frame differencing algorithm described in [6]. The differences between the first and second frames and between the second and third frames are combined using an AND operation. If a wild animal freezes after hearing the camera rotation sound, frame differencing may fail to detect its motion. However, the animal can be detected once it moves again. Frame differencing is used to reduce power consumption by avoiding YOLO inference for every captured frame, while maintaining a balance between detection performance and power consumption.

If motion is detected, the proposed device performs wild animal detection using YOLO. For wild animal detection, the three frames are processed sequentially to improve the probability of detecting a wild animal, since detection may fail in individual frames due to posture changes, motion blur, or partial occlusion. Once a wild animal is detected in a frame, the detection process terminates and tracking starts while generating sound and light. Therefore, the number of detection attempts ranges from one to three. If target animals are not detected, it adjusts the zoom level of the PTZ camera and reattempts detection across all zoom levels to improve detection accuracy by changing the object's size.

If multiple target animals are detected, the proposed device uses DeepSORT to assign temporary tracking IDs to each detected animal to associate detections across consecutive frames and maintain stable tracking of the selected target over a short period. YOLO only generates bounding boxes for each frame and does not provide associations between the same animals across consecutive frames. In contrast, DeepSORT associates detections across consecutive frames using motion prediction, spatial consistency, and appearance features, thereby maintaining a temporary tracking ID for each detected animal.

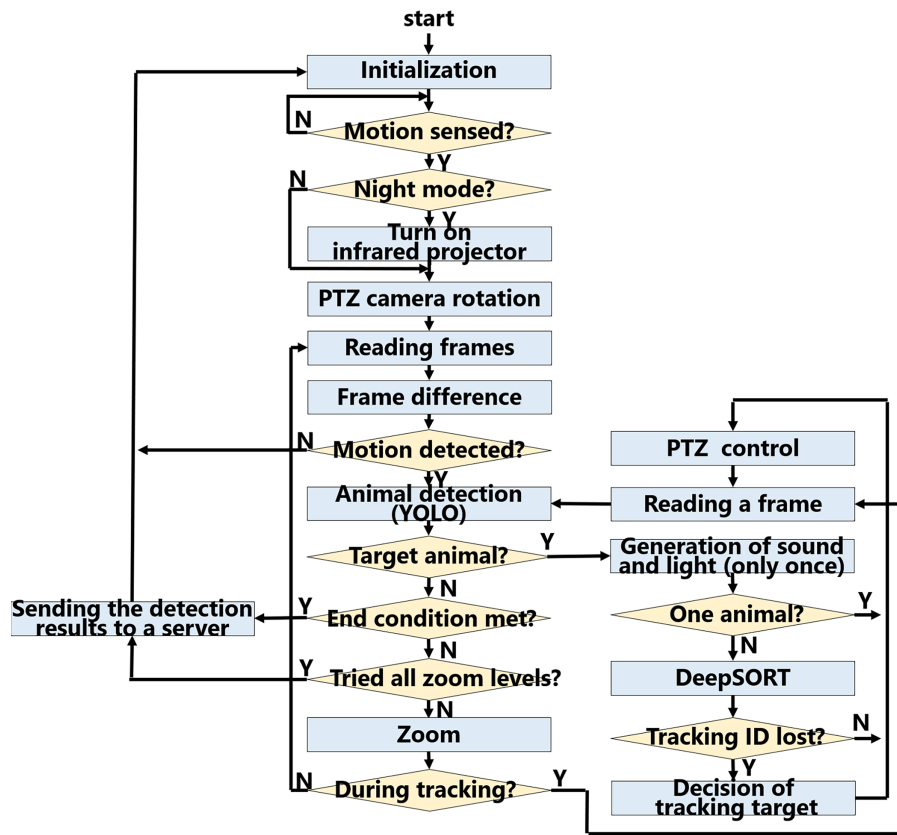


Figure 3: Operation flow of the proposed tracking device.

After performing DeepSORT, the proposed device selects a single tracking target to simplify PTZ control. The size of each animal is then estimated using bounding boxes, and the largest animal is selected as the parent if the size difference exceeds a threshold. If the parent cannot be tracked, the leading animal is selected based on the movement direction estimated from their center coordinates. This target selection process is used to determine which animal should be tracked when multiple animals are detected, and is performed only during tracking initialization or when the tracking ID is lost. Once the target is selected, it is continuously tracked using the tracking ID maintained by DeepSORT in subsequent frames.

The proposed device controls the PTZ camera under three conditions: (i) when changes in the target's center coordinates exceed predefined thresholds, (ii) when changes in target size exceed predefined thresholds, and (iii) when the target is near the image edge. The PTZ control values correspond to control duration, as the proposed device assumes a commercial PTZ camera with a time-based control scheme.

Fig. 4a shows an example of PTZ camera control when changes in the target's center coordinates exceed predefined thresholds. The pan and tilt thresholds are both set to 50. Since the changes in both x - and y -coordinates exceed the thresholds, pan and tilt control are performed simultaneously based on PTZ control values. After the PTZ movement is completed, the target position is observed at frame $(i + j + k)$.

Fig. 4b shows an example of PTZ camera control when changes in the target's size exceed predefined thresholds. The object-area-based zoom threshold is 21,120 pixels. After the zoom control is completed, the target area is updated at frame $(i + j + k)$. Pan and tilt control are performed before zoom control to keep the target centered.

Fig. 4c shows an example of PTZ camera control when the target is near the edge of the image. This control prevents the target from moving out of the field of view. The gray area represents the predefined edge region. Since the target is near the edge, pan and tilt control are performed based on PTZ control values.

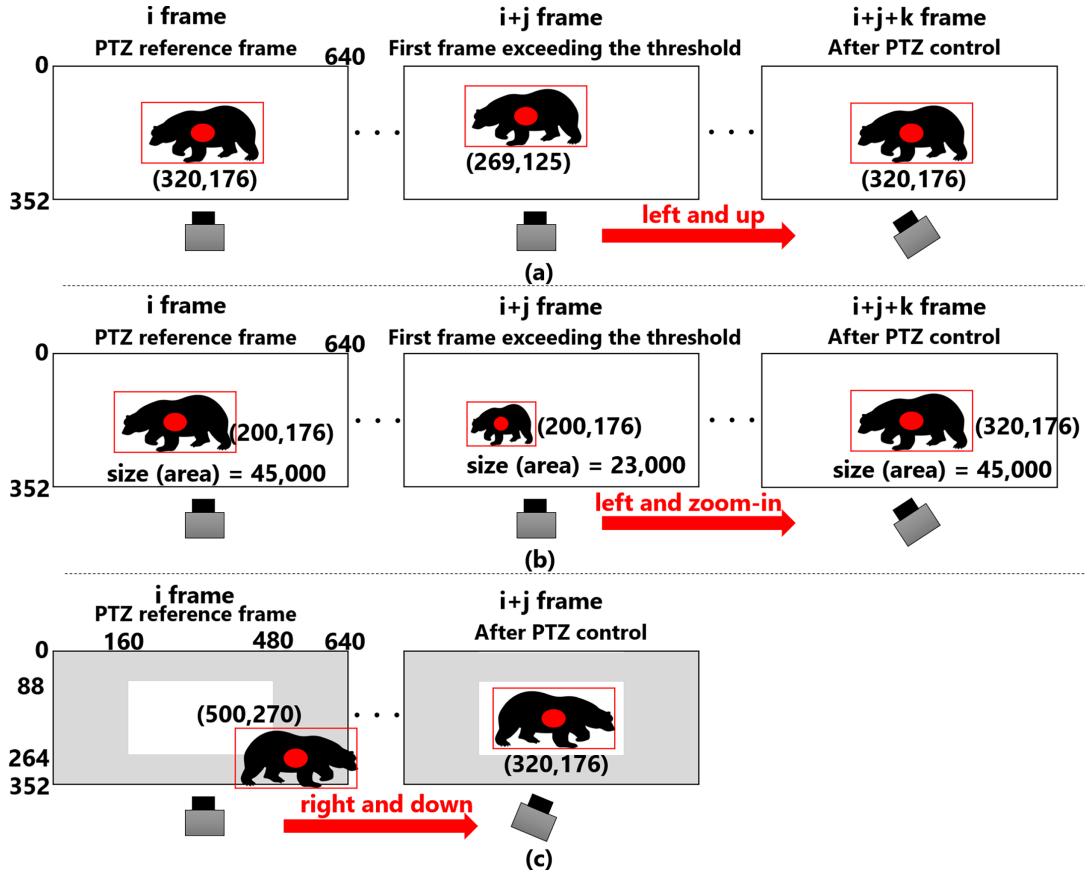


Figure 4: Examples of PTZ camera control. The pan, tilt, and object-area-based zoom thresholds are set to 50, 50, and 21,120, respectively. (a) when changes in the target's center coordinates exceed predefined thresholds, where the target center coordinates change from (320, 176) at frame i to (269, 125) at frame $(i + j)$; (b) when changes in the target's size exceed predefined thresholds, where the target area decreases from 45,000 at frame i to 23,000 at frame $(i + j)$; and (c) when the target is near the image edge, where the target center coordinates are (500, 270) at frame i .

Even if wild animals cannot be detected, the proposed device continues tracking until a predefined tracking time elapses or all zoom levels have been checked. After tracking, it sends detection results to a server.

3.4 Optimization Methods

To achieve efficient wild animal tracking, this section describes three optimization methods in the proposed device:

- Use of different object detection models for before and during tracking to improve accuracy and energy efficiency
- Parallel object detection to improve frame rate
- Dynamic PT control for adapting to animal movements

3.4.1 Use of Different Object Detection Models for before and during Tracking

The proposed device begins tracking only after detecting wild animals, meaning that object detection is required both before and during tracking. It uses detection models suitable for each stage for the following reasons.

Object detection has different requirements before and during tracking. Before tracking, a highly accurate model is required to reduce false detections and avoid unnecessary power consumption. Reducing missed detections is also important to ensure device reliability to notify nearby people of the appearance of wild animals. During tracking, a high-speed model is required in addition to sufficient accuracy, since PTZ control must be performed frequently based on continuous detection results to achieve stable tracking.

To address the different requirements, the proposed device uses two object detection models for before and during tracking: a high-accuracy model for before tracking and a high-speed model for during tracking. This enables smooth tracking while minimizing unnecessary power consumption.

3.4.2 Parallel Object Detection

If object detection, tracking, and PTZ camera control are performed sequentially, the frame rate becomes low. In particular, object detection requires a long processing time and becomes the main bottleneck, leading to long waiting times before the next PTZ control and causing tracking failures. In addition, detection may fail due to changes in object size or pose or partial occlusion, and tracking stops until the target is detected again.

To address these issues, the proposed device performs object detection in parallel to improve the frame rate. YOLO-based detection typically dominates processing time. In contrast, DeepSORT-based tracking depends on the number of detected animals; however, since it is rare for many animals to appear simultaneously, its processing time is relatively short and unlikely to become a bottleneck. Moreover, the tracking process involves data dependencies and database access, making parallelization more complex. Therefore, only object detection is parallelized to improve processing efficiency.

Here, “parallel” means that multiple object detection processes, each consisting of image reading and YOLO inference, are executed simultaneously on different Central Processing Unit (CPU) cores. The main process only dispatches execution commands, while each worker process independently reads images and performs inference. For an n -core CPU, $n-1$ cores are assigned to parallel object detection, and the remaining core is used for the other processes. Therefore, the number of parallel processes is restricted by the number of CPU cores.

Fig. 5 shows an example in which three object detection processes run in parallel, where arrows represent data dependencies. By parallelizing object detection, PTZ control can be performed more frequently, resulting in smoother tracking. In addition, increasing the number of detection attempts across frames reduces the risk of missed detections.

Fig. 6 shows the operation flow for the parallel object detection corresponding to Fig. 5. The proposed device does not assume that all CPU cores require the same processing time. Each frame is assigned a frame number before parallel processing, and each detection result is stored together with its corresponding frame number in a queue. The proposed device retrieves the detection results according to the frame numbers rather than the completion order of the parallel processes. Therefore, even if one core requires more processing time than the others, the frame order is preserved. After receiving the detection results, it tracks wild animals based on each detection result.

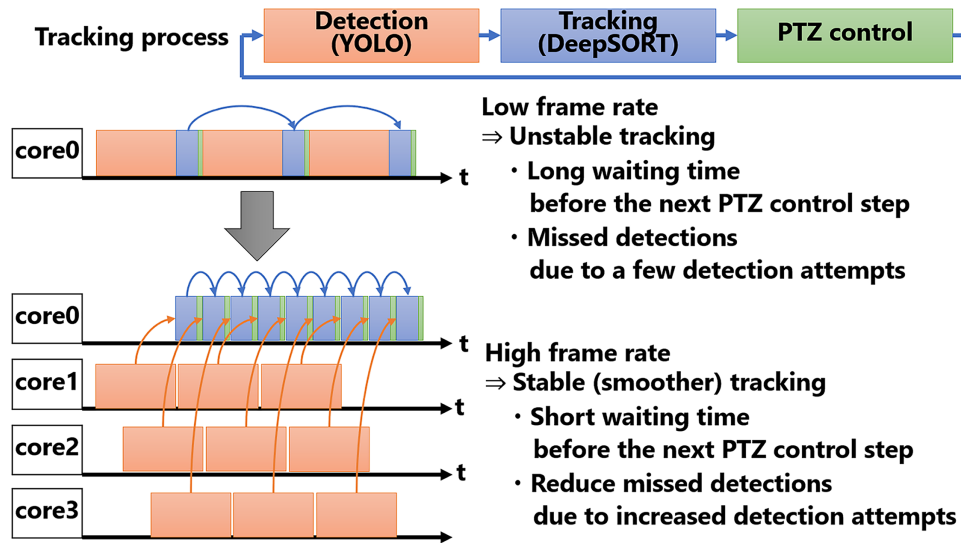


Figure 5: Parallel object detection. In this example, a 4-core CPU is assumed. Three cores are assigned to parallel object detection, while the remaining core is used for the other processes.

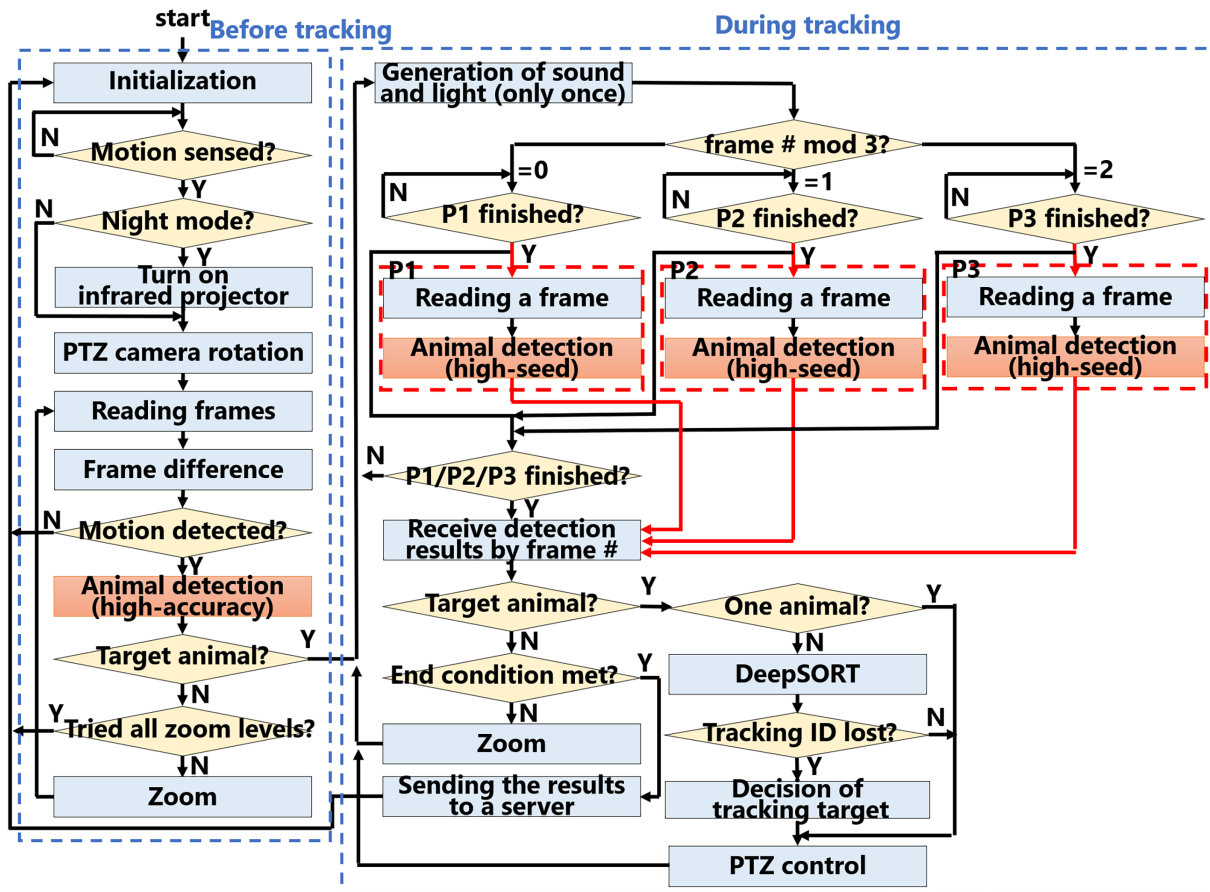


Figure 6: Operation flow for parallel object detection.

3.4.3 Dynamic PT Control

The simplest approach for PT control is to use fixed control values that are independent of the target's position or its changes. However, this approach cannot adapt PT control to changes in the target's position. Therefore, it may fail to keep the target within the field of view, resulting in tracking loss.

To track animals in response to their movements, the proposed device dynamically calculates PT control values based on changes in the target's center coordinates. This enables smooth tracking and reduces the risk of losing the target compared with fixed-value PT control. However, excessive computation for camera control degrades real-time performance; therefore, it is necessary to simplify and dynamically compute PT control values. Note that zoom control is not dynamically calculated, since frequent zoom operations would increase waiting time due to focus adjustments.

To simplify PT control computation and enable dynamic calculation, we introduce the following functions:

$$t_{pan} = \begin{cases} 0, & |\Delta x| \leq th_{pan} \\ \frac{\Delta x}{v_x} \times tm_{pan}, & |\Delta x| > th_{pan} \end{cases} \quad (1)$$

$$t_{tilt} = \begin{cases} 0, & |\Delta y| \leq th_{tilt} \\ \frac{\Delta y}{v_y} \times tm_{tilt}, & |\Delta y| > th_{tilt} \end{cases} \quad (2)$$

where t_{pan} and t_{tilt} represent the pan and tilt control values (seconds), Δx and Δy are the displacements of the target center coordinates in the x - and y -directions (pixels), v_x and v_y are the measured pixel displacement rates during pan and tilt operations (pixels/second), th_{pan} and th_{tilt} are the pan and tilt movement thresholds (pixels), and tm_{pan} and tm_{tilt} are compensation factors for pan and tilt control to account for target movement during PT operation. Δx and Δy are obtained from changes in the target's center coordinates. v_x and v_y are obtained in advance by observing the displacement of an object at the image center for a given pan or tilt control time. In the equations, PT control is not performed when the target displacement is within the thresholds. When the displacement exceeds the thresholds, the pan and tilt control values are calculated based on the target displacement and the compensation factors for target movement during PT operation.

Fig. 7 shows an example of the dynamic PT control method using Eqs. (1) and (2), corresponding to Fig. 4a. The thresholds for both pan and tilt are 50. v_x and v_y are both 220, and tm_{pan} and tm_{tilt} are both 1.5 as parameters for calculating the PT control values. Since the changes in both x and y exceed the thresholds, the proposed device performs leftward pan for 0.348 s and upward tilt for 0.348 s. After the PT control is completed, the target position is observed at frame $(i + j + k)$.

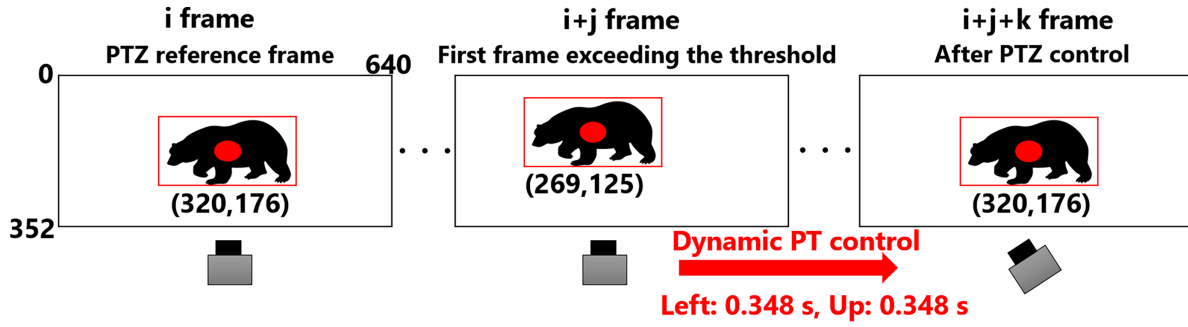


Figure 7: Example of the dynamic PT control corresponding to Fig. 4a, where the target center coordinates change from (320, 176) at frame i to (269, 125) at frame $(i + j)$. The values of the pan threshold, tilt threshold, v_x , v_y , tm_{pan} , and tm_{tilt} are set to 50, 50, 220, 220, 1.5, and 1.5, respectively. The coordinate differences are $\Delta x = |269 - 320| = 51$ and $\Delta y = |125 - 176| = 51$, resulting in $t_{pan} = \Delta x / v_x \times tm_{pan} = 0.348$ s and $t_{tilt} = \Delta y / v_y \times tm_{tilt} = 0.348$ s, respectively.

3.5 Implementation

3.5.1 Hardware Architecture

We implemented the proposed device using commercially available components, as listed in Table 1. The components were connected using jumper wires and Direct Current (DC) cables according to the structure shown in Fig. 2.

Table 1: Used components of the proposed tracking device.

Name	Component
Microcomputer board	Raspberry Pi 4 Model B 4 GB RAM
GPIO expansion board	GeekPi Easy Multiplexing Board
Motion sensor	PIR Infrared Motion Sensor HC-SR501
PTZ camera	Ctronics CTIPC-285C-Z4
Light-emitting device	Patlight SKH-M1J
Speaker	12V 100W 110dB Loud Electronic Horn
Ground loop isolator	Smof Ground Loop Noise Isolator
Audio conversion adapter	MillSO USB Audio Conversion Adapter
Relay module	TASOJIMA 4ch 5V Relay Module
Power meter	Strawberry Linux INA260 Module
Wireless 4G module	Candy Pi Lite+ D
USB memory	Elecom MF-PKU3032GBK/E USB Memory
DC converter	YMax Backup Converter Board BE0917
Battery controller	Renogy 10A PWM Charge Controller
Battery	Long DC12V 50Ah
Solar panel	Renogy DC 12V 100W

We used a Raspberry Pi 4 running Raspberry Pi Operating System Bullseye (64-bit) as the single microcomputer board. The proposed device is powered by a battery for outdoor environments where a power supply is not available. Therefore, we selected the Raspberry Pi 4 by considering its low power consumption and low cost. In addition, the Raspberry Pi 4 provides the computational capability required

for the object detection and tracking tasks while maintaining a balance between processing capability and power consumption, making it suitable for the current implementation.

We used a Ctronics CTIPC-285C-Z4, which captures images at 640×352 resolution and supports 4x optical zoom. We selected the camera by considering the requirements of efficient image processing. Although higher-resolution cameras can provide more detailed images and potentially improve detection performance, they also increase the amount of image data, resulting in higher storage requirements and longer image acquisition and processing times. In practice, the camera supports both 640×352 and 1920×1080 resolutions. However, when capturing images at 640×352 resolution, the image acquisition and saving times were 0.002 and 0.010 s, respectively, whereas those at 1920×1080 resolution increased to 0.023 and 0.113 s, respectively. Therefore, we selected the 640×352 resolution as a practical solution by considering the trade-off between image quality, computational load, and storage requirements.

3.5.2 Software Architecture

We used YOLOv5 [18] for wild animal detection and Omni-Scale Network (OSNet) [19] for feature extraction in DeepSORT. YOLOv5s and YOLOv5n were used for detection before and during tracking, respectively. The same OSNet model as in [17] was used.

The YOLOv5 training dataset consisted of training, validation, and test sets with 60%:20%:20% (7950, 2650, and 2633 images, respectively). Table 2 represents the class distribution. The dataset included Japanese bears, wild boars, raccoon dogs, humans, cars, palm civets, dogs, cats, deer, foxes, antelopes, rabbits, crows, monkeys, and background images. The dataset used in this work was constructed by extending the dataset used in our previous work [6]. Additional images were captured using the same camera from FY2023 to FY2024 and added to the existing dataset. No publicly available datasets or Internet images were used. All images were manually annotated with bounding boxes using LabelImg [20]. Furthermore, to improve the generalization performance, data augmentation techniques, including random cropping, flipping, brightness adjustment, saturation adjustment, hue adjustment, and blurring, were applied to the training images, increasing the number of training images to 96,760.

Table 2: Dataset distribution of annotated animals.

Class	Train		Validation		Test	
	Images	Objects	Images	Objects	Images	Objects
Japanese bear	503	622	169	197	168	210
Wild boar	2231	3141	748	1074	745	1050
Raccoon dog	923	1021	308	345	307	340
Human	795	1166	263	401	252	402
Car	621	668	207	221	210	227
Palm civet	180	190	61	65	59	61
Dog	111	153	43	62	38	55
Cat	283	283	97	97	95	95
Deer	279	282	93	93	93	96
Fox	211	211	70	70	71	72
Antelope	193	193	65	65	64	64
Rabbit	215	215	72	72	73	73
Crow	42	55	14	32	14	27
Monkey	136	218	46	62	44	69
Background	1476	0	479	0	476	0

YOLOv5 was trained from scratch. The training was conducted for 300 epochs with a batch size of 16 using the Stochastic Gradient Descent (SGD) optimizer and a learning rate of 0.01. The input resolutions were set to 480×480 for YOLOv5s and 256×256 for YOLOv5n to reduce inference time. Other training parameters were kept at the default settings. The models were trained using PyTorch 2.0.0 on an NVIDIA GeForce RTX 3070 with CUDA 11.7. The trained models were then converted to TensorFlow Lite int8 format using TensorFlow 2.9.1 and TFLite 2.4.0.

Table 3 represents the comparison of YOLOv5 and YOLO11 in terms of detection accuracy, inference time, and power consumption. Precision and recall were evaluated using the test dataset. The inference time and power consumption were measured on the Raspberry Pi 4 without peripheral devices such as the PTZ camera, using 500 randomly selected images from the test dataset. The reported values represent the mean $\pm$ standard deviation. As shown in Table 3, YOLOv5 and YOLO11 showed different detection performance depending on the model size, although the differences in precision and recall were relatively small. The power consumption of YOLOv5 and YOLO11 was almost identical, while YOLOv5 achieved shorter inference times than YOLO11. Considering the trade-off between detection accuracy, inference time, and power consumption, YOLOv5 was used in this work because the proposed device requires frequent object detection during tracking and aims to minimize energy consumption for long-term battery-powered operation. Note that the higher power consumption of the nano models may be attributed to their higher CPU utilization (over 40%), whereas the small models maintained CPU utilization at around 30%.

Table 3: Comparison of YOLOv5 and YOLO11 in terms of detection accuracy, inference time, and power consumption.

Model (Image Size)	Precision	Recall	Inference Time [s]	Power [W]
YOLOv5s (480×480)	0.936	0.863	0.753 ± 0.073	6.128 ± 0.600
YOLOv5n (256×256)	0.920	0.808	0.104 ± 0.025	6.805 ± 0.340
YOLO11s (480×480)	0.948	0.868	1.169 ± 0.051	5.991 ± 0.589
YOLO11n (256×256)	0.910	0.799	0.187 ± 0.060	6.591 ± 0.637

Based on [21], we implemented the tracking program on the Raspberry Pi 4 using Python 3.9.2 and tflite-runtime 2.11.0. During tracking, parallel wild animal detection and PTZ control were implemented using multithreading. For PTZ control, thresholds for changes in the x - and y -coordinates and the bounding-box area were set to 50, 50, and 21,120, respectively. v_x (v_y) was set to 220 and increased by 60 per zoom level, and tm_{pan} (tm_{tilt}) was set to 1.5 times the control duration to account for target movement. In addition, each edge region was set to 25% of the image width and height. Since the image height is relatively small, both pan and tilt operations are applied when the target is detected near the edge region along the width to prevent losing the target. The PTZ control thresholds were manually determined based on the observed tracking behavior of the proposed device during development to achieve stable PTZ control.

In the frame differencing process, only the lower 70% of the image is used. Three consecutive frames are captured at 0.3 s intervals to increase the frame differences caused by animal movement. The thresholds for the number of changed pixels were empirically set to 300 during daytime and 30 during nighttime. YOLO is performed only when the frame difference exceeds the corresponding threshold. The confidence and Intersection over Union (IoU) thresholds for YOLO were set to 0.7 and 0.45, respectively. When Japanese bears, wild boars, or deer are detected, the proposed device generates sound and light after 60 s and sends the detection results to a server. The reason for the 60-s delay is to separately evaluate the power consumption of the sound and light generation.

4 Experiments

In the experiment, we verify the effectiveness of three optimization methods. Furthermore, through field tests, we demonstrate the proposed device's ability to track wild animals and evaluate it in terms of operation time, power consumption, and energy consumption.

4.1 Evaluation of Use of Different Object Detection Models

To verify the effectiveness of the use of different object detection models for before and during tracking, we evaluated detection performance and energy efficiency. Detection performance was evaluated under animal-present and animal-absent conditions. Energy efficiency was evaluated based on energy consumption during tracking.

Table 4 represents the detection performance of YOLOv5s and YOLOv5n in terms of detection success rate and no-detection rate. The rows represent the models, and the columns represent the detection success rate under animal-present conditions (171 cases) and the no-detection rate under animal-absent conditions (1870 cases). YOLOv5s detected animals in 122/171 cases (0.713) and detected no animals in 1846/1870 cases (0.987). YOLOv5n detected animals in 94/171 cases (0.549) and detected no animals in 1734/1870 cases (0.927). These results indicate that YOLOv5s achieves better detection performance than YOLOv5n, improving the detection success rate by 29% and the no-detection rate by 6%.

Table 4: Detection performance of YOLOv5s and YOLOv5n.

Model	Detection Success Rate	No-Detection Rate
YOLOv5s	0.713	0.987
YOLOv5n	0.549	0.927

Table 5 represents the average operation time (T), the average power consumption (P), the average energy consumption (E), and the number of events (N) when using YOLOv5s and YOLOv5n. The values of T and P are reported as mean $\pm$ standard deviation. Each process corresponds to the process shown in Fig. 3, where *Tracking* represents the entire processing during tracking, and *Idle* represents the idle state. The results were obtained from a daytime video sequence consisting of 234 frames, and each experiment was repeated three times. For each process, the operation time and power consumption were measured every time the process was executed, and the reported values are the averages of these measurements. The operation time was measured using Python's `perf_counter` function. The power consumption was measured by the power meter. The idle power consumption was measured for 5 s immediately after tracking was completed. The average energy consumption was calculated by multiplying the average operation time by the average power consumption. As shown in Table 5, YOLOv5n achieved a shorter operation time for wild animal detection than YOLOv5s, resulting in a 64% reduction in energy consumption of *Tracking*.

From these results, YOLOv5s applied before tracking enables the detection of more animals while suppressing unnecessary tracking operations due to its higher detection success and no-detection performance. During tracking, YOLOv5n reduces energy consumption owing to its shorter operation time per frame. Therefore, by separating the detection processes before and during tracking and applying stage-appropriate object detection models, both detection performance and energy efficiency can be improved.

Table 5: Operation time (T), power consumption (P), energy consumption (E), and the number of events (N) using YOLOv5s and YOLOv5n.

		Yolov5s				Yolov5n			
	Process	T [s]	P [W]	E [J]	N	T [s]	P [W]	E [J]	N
Before tracking	Reading frames	0.96 ± 0.01	7.06 ± 0.72	6.78	3	0.96 ± 0.01	7.04 ± 0.73	6.76	3
	Frame difference	0.04 ± 0.01	7.80 ± 0.29	0.32	3	0.04 ± 0.01	7.65 ± 0.87	0.31	3
	Animal detection	0.92 ± 0.03	8.32 ± 0.53	7.66	3	0.28 ± 0.02	8.04 ± 0.51	2.26	3
During tracking	Reading a frame	0.02 ± 0.01	8.45 ± 1.07	0.17	300	0.02 ± 0.01	9.21 ± 1.26	0.185	634
	Animal detection	0.81 ± 0.03	8.75 ± 1.15	7.09	300	0.09 ± 0.01	9.70 ± 1.26	0.88	634
	PTZ control	0.90 ± 0.03	8.41 ± 1.05	7.57	60	0.79 ± 0.04	8.35 ± 1.10	6.60	21
	Zoom	6.96 ± 3.65	10.19 ± 0.85	70.93	9	6.36 ± 3.27	10.20 ± 0.87	64.88	9
	Tracking	88.31 ± 0.37	8.71 ± 1.15	769.19	3	29.02 ± 0.49	9.53 ± 1.29	276.56	3
	Idle	–	7.07 ± 0.71	–	–	–	7.02 ± 0.62	–	–

4.2 Evaluation of Parallel Object Detection

To verify the effectiveness of the parallel object detection, we evaluated the performance with and without parallel execution of YOLOv5n during tracking during the daytime. The evaluation metrics include the average frame rate (FPS), the average power consumption (P), and FPS per watt (FPS/W) calculated from the average FPS and average power consumption. The values of FPS and P are reported as mean $\pm$ standard deviation. The results were obtained from a daytime video sequence consisting of 234 frames, and each experiment was repeated three times. For each process, the operation time and power consumption were measured every time the process was executed, and the reported values are the averages of these measurements. The daytime experiments were conducted without the infrared projector.

Table 6 represents the results of FPS , power consumption, and FPS per watt during tracking with and without parallel execution of YOLOv5n. The method with parallel execution achieved 11.91 FPS with 10.31 W, resulting in 1.15 FPS/W , whereas the method without parallel execution achieved 7.31 FPS with 9.53 W, resulting in 0.77 FPS/W . This resulted in improvements of 62% in FPS and 49% in FPS/W , while the power consumption increased by 9%. Table 7 represents the average operation time, the average power consumption, the average energy consumption, and the number of events during tracking with and without parallel execution of YOLOv5n. Although the parallel execution increases the per-frame energy consumption due to higher computational load, it reduces the total energy consumption during tracking by improving processing efficiency. Therefore, overall energy efficiency can be improved by parallel object detection during tracking. Note that since the parallel execution performs a more computationally intensive workload before entering the idle state, the measured idle power consumption may still include the residual effect of the preceding processing.

Table 6: FPS , power consumption, and FPS per watt during tracking with and without parallel object detection.

Method	FPS	P [W]	FPS/W
Non-parallel	7.31 ± 0.04	9.53 ± 1.29	0.77
Parallel	11.91 ± 0.09	10.31 ± 1.30	1.15

Table 7: Operation time (T), power consumption (P), energy consumption (E), and the number of events (N) during tracking with and without parallel execution.

Process		Non-Parallel				Parallel			
		T [s]	P [W]	E [J]	N	T [s]	P [W]	E [J]	N
Before tracking	Reading frames	0.96 ± 0.01	7.04 ± 0.73	6.76	3	0.99 ± 0.05	7.34 ± 0.85	7.27	3
	Frame difference	0.04 ± 0.01	7.65 ± 0.87	0.31	3	0.04 ± 0.01	7.97 ± 0.01	0.32	3
	Animal detection	0.28 ± 0.02	8.04 ± 0.51	2.26	3	0.26 ± 0.02	7.96 ± 0.44	2.07	3
During tracking	Reading a frame	0.02 ± 0.01	9.21 ± 1.26	0.185	634	0.05 ± 0.02	10.32 ± 1.22	0.52	696
	Animal detection	0.09 ± 0.01	9.70 ± 1.26	0.88	634	0.15 ± 0.04	10.45 ± 1.31	1.57	696
	PTZ control	0.79 ± 0.04	8.35 ± 1.10	6.60	21	0.74 ± 0.19	9.54 ± 1.47	7.06	17
	Zoom	6.36 ± 3.27	10.20 ± 0.87	64.88	9	6.35 ± 3.25	10.03 ± 1.28	63.69	3
	Tracking	29.02 ± 0.49	9.53 ± 1.29	276.56	3	19.22 ± 0.14	10.31 ± 1.30	198.16	3
	Idle	–	7.02 ± 0.62	–	–	–	7.25 ± 0.69	–	–

4.3 Evaluation of Dynamic PT Control

To verify the effectiveness of the dynamic PT control, we evaluated the dynamic PT control and the fixed PT control. The fixed PT control used constant pan and tilt durations of 0.1, 0.5, and 1.0 s for pan control, and 0.1, 0.2, and 0.3 s for tilt control, whereas the dynamic PT control adaptively determines the control duration using Eqs. (1) and (2). Note that tm_{pan} and tm_{tilt} were both set to 1.0 because only the relationship between the target coordinates and the calculated PT control values was evaluated. The evaluation was conducted using simulated target movements, where the target was initially located at the image center (320, 176) and moved to (419, 227), (517, 259), or (591, 303). After applying PT control, performance was evaluated using RMSE and maximum error between the final target position and the image center.

Table 8 shows the RMSE and maximum error in the x and y directions for the fixed and dynamic PT control methods under different control value settings. The rows correspond to different control value settings for the pan and tilt axes, while the columns present RMSE and maximum error in both directions. The fixed control method can achieve lower errors under specific value settings and target positions. However, its performance is highly sensitive to target position changes, which may lead to increased errors and potential tracking failure. In contrast, the dynamic control method maintains relatively low errors regardless of target position, indicating more robust and stable tracking performance. These results indicate that the dynamic PT control method is more suitable for stable object tracking under varying target positions, as it consistently reduces sensitivity to positional changes.

In addition, the computational time, including frame-to-frame coordinate difference calculation, was 0.007 s for the fixed control and 0.004 s for the dynamic control. This indicates that both methods have comparable computational costs and are suitable for real-time implementation.

Table 8: RMSE and maximum error for fixed and dynamic PT control methods.

Method	t_{pan} [s]	t_{tilt} [s]	$RMSE_x$	$RMSE_y$	$e_{max,x}$	$e_{max,y}$
Fixed	0.1	0.1	161.9	56.5	227.0	89.5
		0.2	161.3	38.0	226.5	54.0
		0.3	162.5	64.6	230.0	86.5
	0.5	0.1	80.5	58.3	121.5	93.0
		0.2	78.2	33.4	117.5	44.5
		0.3	78.4	52.0	117.0	76.5
	1.0	0.1	142.2	51.8	212.0	81.5
		0.2	142.8	30.2	212.5	43.0
		0.3	140.1	52.2	204.5	74.0
Dynamic	–	–	21.9	47.5	30.5	63.5

4.4 Field Test Results

We placed the proposed device, with the three optimization methods applied, in Aizu-Wakamatsu City, Fukushima Prefecture, Japan. The proposed device was operated from 26 May to 30 November 2025, between 5:00 pm and 9:00 am of the next day.

4.4.1 Detection and Tracking Performance

During the field test, the motion sensor triggered the proposed device in a total of 5270 cases (trigger events). For each case, the captured image was manually inspected to determine whether an animal was present. Animals were visually confirmed in 206 cases, while no animals were present in the captured images for the remaining 5064 cases, as shown in [Table 9](#).

Table 9: Effect of frame differencing.

Condition	Cases	Motion Detected	No Motion Detected
No animal cases	5064	1870	3194
Animal present cases	206	171	35

Among the 5064 cases with no animals, frame differencing detected no motion in 3194 cases, thereby preventing unnecessary detection processes and reducing energy consumption. Among the 206 cases with animals, frame differencing detected motion in 171 cases, allowing the proposed device to proceed to animal detection. In the remaining 35 cases, no motion was detected by frame differencing; 16 cases were caused by small movements and 19 cases were caused by incomplete body capture. These results indicate that frame differencing effectively reduced unnecessary detection and power consumption, making it suitable for battery-powered devices. However, frame differencing may fail to trigger object detection when the target movement does not exceed the predefined threshold or when only a small portion of the target is captured.

The proposed device detected wild animals in 122 of the 171 cases that passed frame differencing, corresponding to a detection rate of over 71%, as shown in [Table 10](#). Detection was missed in 49 cases. [Fig. 8](#) shows example images of missed detections. In these cases, the animals occupied only a small region of the image, were partially occluded, had low contrast with the background, or appeared blurred due to adverse

weather conditions, making detection difficult for object detection. Among the 49 missed detections, 5 cases were caused by small target size, 25 cases were caused by partial visibility, 7 cases were caused by low contrast with the background, and 4 cases were caused by image blur. The remaining 8 cases appeared to be detectable by visual inspection. When the YOLO confidence threshold was reduced from 0.7 to 0.6, 4 of these 8 cases were correctly detected. However, lowering the confidence threshold also increases the number of false positives. Therefore, selecting an appropriate confidence threshold involves a trade-off between detection rate and false positives.

Table 10: Detection performance after frame differencing.

Cases	Correct Detections	Missed Detections
171	122	49



Figure 8: Example images of missed detections caused by (a) small target size, (b) partial visibility, (c) low contrast with the background, and (d) image blur.

Table 11 represents the number of PTZ control operations. The columns represent the total number of PTZ control operations and their breakdown into *Meaningless control* (operations performed after the target had already disappeared from the image, mainly zoom operations), *Correct control* (operations in which the target remained within the field of view after control), and *Incorrect control* (operations in which the target was lost from the field of view after control).

Table 11: Number of PTZ control operations.

Total	Meaningless Control	Correct Control	Incorrect Control
1035	163	780	92

The PTZ camera was controlled in 1035 cases in total. Among these operations, excluding 163 meaningless controls, 872 effective control operations were analyzed. Among the effective controls, 780 cases were correctly controlled, corresponding to approximately 89%, indicating that the target animal remained within the field of view after PTZ control in most cases. In contrast, 92 cases were incorrect controls, where the target could not be maintained within the field of view after the control. These incorrect controls mainly occurred due to delays in camera response and situations where the target moved rapidly or unpredictably, resulting in a failure to maintain visual tracking.

Fig. 9 shows an example of the tracking result when a wild boar appeared at 7:12 p.m. on July 27. At the 57th frame, the center coordinates of the target were (150.5, 106.0). Since the target was located near the edge of the image, the proposed device performed pan and tilt control to move the PTZ camera leftward and upward. At the 82nd frame, the center coordinates changed from (268.5, 251.0) to (217.5, 245.5). Since the change in the x -coordinate exceeded the threshold, the proposed device performed an additional leftward

pan control. At the 168th frame, the target was temporarily lost, and a zoom-in operation was performed to detect the wild animal. As a result, the target was detected again, allowing the tracking to continue.

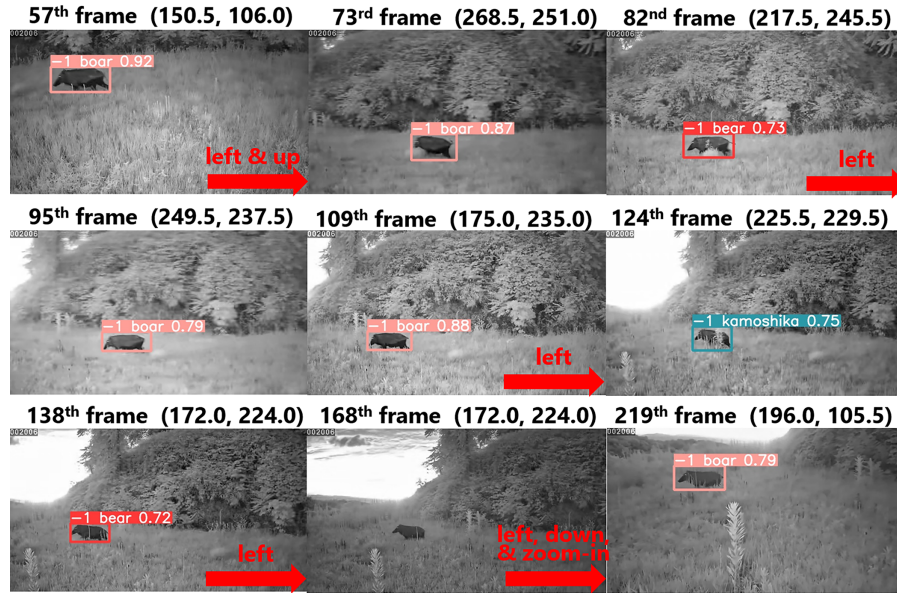


Figure 9: Example of the tracking result at 7:12 p.m. on July 27.

4.4.2 Energy Consumption

We evaluated the average operation time, power consumption, and energy consumption of the proposed device for the 122 cases in which wild animals were detected after passing frame differencing. Tables 12 and 13 represent the average operation time (T), the average power consumption (P), the average energy consumption (E), and the number of events (N) of the proposed device without sound and light alerts and with sound and light alerts. For each process, the operation time and power consumption were measured every time the process was executed, and the reported values are the averages of these measurements.

Before tracking was initiated, the maximum operation times of *Frame difference* and *Animal detection* were 0.15 and 3.97 s, respectively. Consequently, the maximum operation time from the start of *Frame difference* to the completion of *Animal detection* was 4.12 s, while the average operation time was 1.52 s.

The operation time for *Tracking* varies depending on the detection duration. Similarly, the operation time for *PTZ control* also varies depending on the target's coordinates. The operation time for *Animal detection* is consistent across tracking, taking 0.17 s per image for the inference. The proposed device achieved 10.67 FPS on average, and the FPS was limited by object detection and DeepSORT.

The power consumption for *Tracking* was 10.14 W during the daytime and 13.47 W during the nighttime. The higher power consumption during the nighttime is due to the use of the infrared projector. Compared with *Idle*, the power consumption of *Tracking* increased significantly because of the high power consumption of the PTZ camera, such as *PTZ control* and *Zoom*. *Tracking* with sound and light generated during tracking consumed more power than when sound and light were generated after tracking because the proposed device performed tracking while generating them.

Table 12: Operation time (T), power consumption (P), energy consumption (E), and the number of events (N) of the proposed device without sound and light alerts.

Process		Day				Night			
		T [s]	P [W]	E [J]	N	T [s]	P [W]	E [J]	N
Before tracking	Camera rotation	2.61 ± 0.01	8.45 ± 0.99	22.06	2	2.59 ± 0.01	11.05 ± 1.84	28.62	18
	Reading frames	1.03 ± 0.11	7.21 ± 0.88	7.43	4	0.99 ± 0.08	10.50 ± 1.38	10.40	48
	Frame difference	0.04 ± 0.03	7.42 ± 1.01	0.30	4	0.04 ± 0.02	11.17 ± 1.14	0.45	48
	Animal detection	1.83 ± 0.84	7.71 ± 0.59	14.11	4	1.75 ± 0.83	11.00 ± 1.44	19.25	48
	Zoom	–	–	–	–	10.60 ± 3.02	11.27 ± 2.08	119.47	12
During tracking	Reading a frame	0.07 ± 0.05	10.08 ± 1.22	0.71	1010	0.06 ± 0.04	13.53 ± 1.41	0.82	13,116
	Animal detection	0.17 ± 0.05	10.22 ± 1.11	1.74	1010	0.16 ± 0.13	13.54 ± 1.44	2.17	13,116
	DeepSORT	–	–	–	–	0.08 ± 0.03	13.64 ± 1.77	1.10	11
	PTZ control	0.51 ± 0.60	9.76 ± 1.76	4.98	21	0.24 ± 0.36	13.05 ± 1.48	3.14	797
	Zoom	6.32 ± 3.28	10.42 ± 0.90	65.86	12	5.96 ± 2.96	13.87 ± 1.31	82.67	134
	Tracking	25.14 ± 3.67	10.14 ± 1.18	254.92	4	33.99 ± 18.34	13.47 ± 1.40	457.85	36
Idle		–	6.95 ± 0.68	–	–	–	7.06 ± 0.99	–	–

Table 13: Operation time (T), power consumption (P), energy consumption (E), and the number of events (N) of the proposed device with sound and light alerts.

Process		Night							
		Sound & Light After Tracking				Sound & Light During Tracking			
		T [s]	P [W]	E [J]	N	T [s]	P [W]	E [J]	N
Before tracking	Camera rotation	2.59 ± 0.01	11.27 ± 1.73	29.19	31	2.59 ± 0.01	11.17 ± 1.58	28.93	4
	Reading frames	0.99 ± 0.06	10.91 ± 0.98	10.80	68	1.02 ± 0.13	11.10 ± 0.74	11.33	21
	Frame difference	0.04 ± 0.02	11.27 ± 1.30	0.45	68	0.04 ± 0.02	10.97 ± 0.75	0.44	21
	Animal detection	1.29 ± 0.60	11.56 ± 0.90	14.92	68	1.41 ± 0.63	11.71 ± 0.66	16.52	21
	Zoom	8.91 ± 0.01	12.19 ± 0.92	108.62	5	8.92 ± 0.01	12.24 ± 1.04	109.19	2
During tracking	Reading a frame	0.06 ± 0.04	13.65 ± 1.63	0.82	21,178	0.08 ± 0.04	16.49 ± 6.76	1.32	16,603
	Animal detection	0.16 ± 0.07	13.74 ± 1.49	2.20	21,178	0.15 ± 0.05	16.61 ± 6.80	2.50	16,603
	DeepSORT	0.08 ± 0.10	12.72 ± 1.49	1.02	210	0.07 ± 0.04	14.29 ± 5.02	1.00	1374
	PTZ control	0.23 ± 0.38	12.87 ± 1.39	2.96	1349	0.15 ± 0.20	14.48 ± 4.96	2.18	4439
	Zoom	6.18 ± 3.06	14.14 ± 1.22	87.39	202	6.03 ± 2.99	21.21 ± 7.92	127.90	76
	Sound & light	50.29 ± 6.20	18.50 ± 7.42	930.37	63	59.12 ± 20.39	20.41 ± 7.38	1206.64	19
	Sending results	8.94 ± 32.23	7.22 ± 0.69	64.55	63	4.73 ± 0.73	7.62 ± 0.73	36.05	19
	Tracking	30.42 ± 9.84	13.59 ± 1.42	413.41	63	94.11 ± 29.69	16.13 ± 6.55	1518.00	19
Idle		–	7.00 ± 0.90	–	–	–	7.04 ± 0.84	–	–

Direct implementation of existing PTZ-based tracking approaches in [9–14] on the proposed device is difficult due to differences in hardware configuration and operating conditions. Existing approaches generally assume continuous video streaming without motion-triggered operation, which results in continuous image acquisition and object detection. Therefore, we compared the power consumption of the proposed device during the idle operation with the power consumption when the proposed device continuously performs video streaming and animal detection. Compared with continuous video streaming approaches, the proposed device achieved lower power consumption in *Idle*. In continuous video streaming approaches, wild animal detection is performed continuously, resulting in constant power consumption of 7.71 W during

the daytime and 11.00 W during the nighttime, as shown in *Animal detection* of Table 12. In contrast, the proposed device reduced the power consumption of *Idle* by 9% during the daytime and 35% during the nighttime, as it operates only when motion is detected by the motion sensor.

The energy consumption of the proposed device depends on the detection duration, the number of PTZ camera control operations, and whether sound and light were generated. Similarly to the power consumption, the energy consumption during the nighttime was higher than that during the daytime.

Fig. 10 shows the relationship between the voltage of the proposed device and the daily sunshine duration. The blue line represents the voltage measured by the power meter at 00:00 (midnight), while the red line represents the sunshine duration obtained from the Japan Meteorological Agency [22]. When several days of sunshine are lacking, the voltage drops below 12.0 V. Nevertheless, the proposed device can operate sufficiently using the battery and solar panel if sufficient sunshine is available.

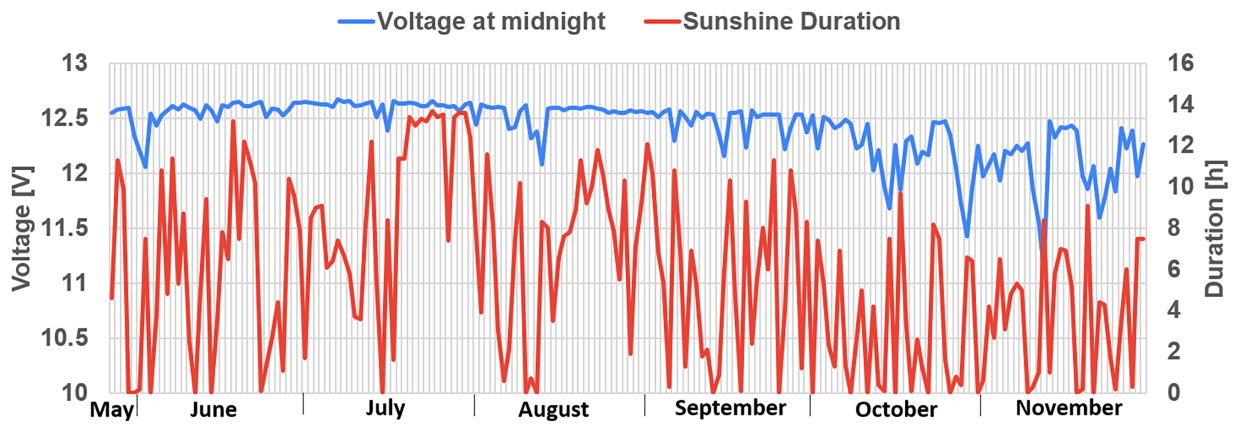


Figure 10: Relationship between the voltage of the proposed device and the daily sunshine duration.

The daily energy consumption depends on the idle duration and the frequency and duration of tracking operations. The idle power consumption of the proposed device was 6.95–7.06 W. During tracking operations, the power consumption was 10.14 W during the daytime and 13.47–16.13 W during the nighttime. Therefore, the daily energy consumption can be estimated based on the operating time of each mode. Since animal appearance events are expected to be infrequent in typical wild animal monitoring scenarios, the idle operation can become a dominant component of the total daily energy consumption. For example, when no motion events occur during a day, the energy consumption is 169.44 Wh/day under the idle operation. In contrast, assuming that 20 tracking events occur during the nighttime and each event requires 2 min, the energy consumption is estimated to be 175.49 Wh/day. These estimations provide a reference for selecting a battery capacity depending on the expected animal activity.

5 Discussion

5.1 PT Control

In the proposed device, the simple dynamic PT control method is introduced to reduce the computational cost of determining PT control values, thereby enabling frequent camera control on a battery-powered device. Although it reduces computational cost, more advanced control methods, such as Proportional-Integral-Derivative (PID) control, Kalman filter-based prediction, or learning-based approaches, may improve tracking accuracy. Investigating these approaches while considering the computational constraints of battery-powered devices will be considered in future work.

The PT control time calculation assumes that the target maintains its current movement direction during the short PT control interval. Although wild animals may move unpredictably, this short-term assumption is used to calculate the PT control values based on the observed target movement. Sudden changes in movement direction may affect the tracking performance of the proposed device. The robustness against various animal movement patterns is a consideration for future improvements.

5.2 Tracking

In the complete pipeline, 122 of 206 animal-present cases were detected, corresponding to an overall detection rate of 59.2%. The missed detections occurred at two stages: frame differencing and object detection. In 35 cases, the frame difference did not exceed the predefined threshold, and animal detection was not performed. In the remaining 49 cases, frame differencing triggered object detection, but YOLO failed to detect the animals due to factors such as partial visibility, small target size, or background similarity. Achieving over 71% detection under challenging outdoor conditions, which are characterized by unpredictable movement, variable lighting, complex backgrounds, etc., demonstrates the effectiveness of the proposed device. On the other hand, these results indicate that improvements in both the motion-based trigger mechanism and the object detection model are necessary to further improve the detection performance of the proposed device.

In the proposed device, the Raspberry Pi 4 and Ctronics CTIPC-285C-Z4 camera are used to realize a battery-powered device. However, the performance and efficiency of the proposed device may be further improved by evaluating newer embedded platforms, such as the Raspberry Pi 5 and Graphics Processing Unit (GPU)-equipped devices. In addition, tracking performance may be improved by introducing a high-speed PTZ camera and incorporating motion direction prediction based on pose estimation. These improvements will be considered in future work.

6 Conclusion

In this paper, we proposed an efficient battery-powered autonomous wild animal tracking device that integrates object detection, tracking, and PTZ camera control. The proposed device enables autonomous wild animal tracking in outdoor environments without stable power supplies. To achieve efficient tracking, it uses different object detection models for before and during tracking, performs parallel object detection, and applies dynamic PT control.

Experimental results demonstrated the effectiveness of the proposed device. The use of different object detection models for before and during tracking improved the detection success rate by 29% and reduced the energy consumption during tracking by 64%. The parallel object detection increased FPS by 62% and improved FPS/W by 49%. The dynamic PT control reduced RMSE compared with the fixed control method. The field tests showed that the proposed device tracked wild animals with an 89% success rate in maintaining the target within the field of view after PTZ control. The power consumption during tracking was 10.14 W during the daytime and 13.47 W during the nighttime, while the idle power consumption was 6.95 and 7.06 W, demonstrating that the proposed device can operate using a solar panel and a battery.

In our future work, we are going to improve the detection accuracy by training with many images of partially visible animals, small-sized animals, and animals blending into the background. In addition, we are going to improve the tracking performance by predicting the target's movement near image edges, considering the PTZ camera's response delay, and investigating lightweight tracking algorithms. Furthermore, we are going to evaluate the energy consumption of the proposed device using alternative hardware platforms and investigate more advanced PTZ control methods while considering the trade-off between tracking accuracy and computational cost.

Acknowledgement: We would like to thank the people in the Aizu Area of Fukushima Prefecture for their cooperation in the field test. We also acknowledge the support of the Research Institute of Science and Technology for Society, Japan Science and Technology Agency.

Funding Statement: This work was funded by the Japan Science and Technology Agency, grant number JPMJRS24I3.

Author Contributions: Conceptualization, Shogo Semba, Hiroshi Saito, Yoichi Tomioka and Yukihide Kohira; methodology, Shogo Semba, Hiroshi Saito, Yoichi Tomioka and Yukihide Kohira; software, Shogo Semba; validation, Shogo Semba; formal analysis, Shogo Semba; investigation, Shogo Semba; resources, Hiroshi Saito; data curation, Shogo Semba; writing—original draft preparation, Shogo Semba; writing—review and editing, Hiroshi Saito, Yoichi Tomioka and Yukihide Kohira; visualization, Shogo Semba; supervision, Hiroshi Saito; project administration, Hiroshi Saito; funding acquisition, Hiroshi Saito. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: Data and materials used in this research cannot be distributed due to privacy concerns, as the images collected may contain people or vehicles. We agreed with the residents at the field test locations not to distribute the images to anyone else.

Ethics Approval: Not applicable. This research detects and tracks wild animals in the images taken by wild animal tracking devices and does not involve invasive animal experimentation.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ministry of Environment. Number of captures and damage situation [Internet]. [cited 2026 Jan 6]. Available from: <https://www.env.go.jp/nature/choju/docs/docs4/>. (In Japanese).
2. Ministry of Agriculture, Forestry and Fisheries. Crop damage situation [Internet]. [cited 2026 Jan 6]. (In Japanese). Available from: https://www.maff.go.jp/j/seisan/tyozyu/higai/hogai_zyoukyou/index.html.
3. Tomkiewicz SM, Fuller MR, Kie JG, Bates KK. Global positioning system and associated technologies in animal behaviour and ecological research. *Phil Trans R Soc B*. 2010;365(1550):2163–76. doi:10.1098/rstb.2010.0090.
4. Ross R, Anderson B, Bienvenu B, Scicluna EL, Robert KA. WildTrack: an IoT system for tracking passive-RFID microchipped wildlife for ecology research. *Automation*. 2022;3(3):426–38. doi:10.3390/automation3030022.
5. Farine DR, Penndorf J, Bolcato S, Nyaguthii B, Aplin LM. Low-cost animal tracking using Bluetooth low energy beacons on a crowd-sourced network. *Methods Ecol Evol*. 2024;15(12):2247–61. doi:10.1101/2024.08.15.608178.
6. Sato R, Saito H, Tomioka Y, Kohira Y. Energy reduction methods for wild animal detection devices. *IEEE Access*. 2022;10:24149–61.
7. Ray S, Stopfer MA. Argos: a toolkit for tracking multiple animals in complex visual environments. *Meth Ecol Evol*. 2022;13(3):585–95. doi:10.1111/2041-210x.13776.
8. Pereira TD, Tabris N, Matsliah A, Turner DM, Li J, Ravindranath S, et al. SLEAP: a deep learning system for multi-animal pose tracking. *Nat Meth*. 2022;19(4):486–95. doi:10.1038/s41592-022-01426-1.
9. Martono NP, Tsukamoto R, Ohwada H. An Internet of Things approach to vision-based livestock monitoring: PTZ cameras for dairy cow identification. *Telecom*. 2025;6(4):82. doi:10.3390/telecom6040082.
10. You JA, Hu Z, Xiao H, Xu C. Cell-based target localization and tracking with an active camera. *Appl Sci*. 2022;12(6):2771. doi:10.3390/app12062771.
11. Niu H, Liu J, Yu Z, Zheng D, He P, Wang F. Real-time object tracking system using PTZ camera. In: *Proceedings of the 2022 IEEE 4th International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*; 2022 Oct 12–14; Dali, China. New York, NY, USA: 2022. p. 471–8.
12. Sandha SS, Balaji B, Garcia L, Srivastava M. Eagle: end-to-end deep reinforcement learning based autonomous control of PTZ cameras. In: *Proceedings of the 8th ACM/IEEE Conference on Internet of Things Design and Implementation*; 2023 May 9–12; San Antonio, TX, USA. New York, NY, USA: ACM; 2023. p. 144–57.

13. Yang Z, Fang H, Liu H, Li J, Jiang Y, Zhu M. Active visual perception enhancement method based on deep reinforcement learning. *Electronics*. 2024;13(9):1654. doi:10.3390/electronics13091654.
14. Fang H, Liu H, Wen J, Yang Z, Li J, Han Q. Automatic visual enhancement of PTZ camera based on reinforcement learning. *Neurocomputing*. 2025;626(5):129531. doi:10.1016/j.neucom.2025.129531.
15. Ultralytics. Ultralytics YOLO documentation [Internet]. [cited 2026 Jan 6]. Available from: <https://docs.ultralytics.com/models>.
16. Wojke N, Bewley A, Paulus D. Simple online and realtime tracking with a deep association metric. In: *Proceedings of the 2017 IEEE International Conference on Image Processing*; 2017 Sep 17–20; Beijing, China. p. 3645–9.
17. Semba S, Saito H, Tomioka Y, Kohira Y. A battery-powered wild animal tracking device using a PTZ camera and deep learning. In: *Proceedings of the 2024 IEEE International Conference on Systems, Man, and Cybernetics*; 2024 Oct 6–10; Kuching, Malaysia. p. 3181–6.
18. Ultralytics. YOLOv5 [Internet]. [cited 2023 Apr 19]. Available from: <https://github.com/ultralytics/yolov5>.
19. Zhou K, Yang Y, Cavallaro A, Xiang T. Omni-scale feature learning for person re-identification. In: *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision*; 2019 Oct 27–Nov 2; Seoul, Republic of Korea. p. 3701–11.
20. Tzutalin. LabelImg [Internet]. [cited 2023 June 19]. Available from: https://github.com/HumanSignal/labelimg?utm_source=chatgpt.com.
21. Brostrom M. Real-time multi-object tracker using YOLOv5 and deep sort [Internet]. [cited 2022 Apr 19]. Available from: <https://github.com/mikel-brostrom/boxmot/tree/fc1da45>.
22. Japan Meteorological Agency. Historical weather data download [Internet]. [cited 2025 Dec 24]. Available from: <https://www.data.jma.go.jp/risk/obsdl/index.php>. (In Japanese).