Sensors and Materials, Vol. 35, No. 11 (2023) 3947–3968
MYU Tokyo

S & M 3461

Deep-learning-based Multi-behavior Classification of Animals for Efficient Health and Welfare Monitoring

Ruqin Wang,[1*] Wataru Noguchi,[2] Koki Osada,[1] and Masahito Yamamoto[3]

[1]Graduate School of Information Science and Technology, Hokkaido University,
Kita 14, Nishi 9, Kita-ku, Sapporo, Hokkaido 060-0814, Japan
[2]Education and Research Center for Mathematical and Data Science, Hokkaido University,
Kita 12, Nishi 7, Kita-ku, Sapporo, Hokkaido 060-0812, Japan
[3]Faculty of Information Science and Technology, Hokkaido University,
Kita 14, Nishi 9, Kita-ku, Sapporo, Hokkaido 060-0814, Japan

(Received May 18, 2023; accepted September 12, 2023)

Keywords: ethology, stereotypical behavior, animal behavior recognition, object detection, YOLOv5s

With the development of sensor technologies, sensors have become increasingly embedded in various fields, becoming an indispensable part of our daily lives, research, and work. Notably, in ethology, surveillance cameras, a type of optical sensor, are extensively used alongside machine learning to analyze animal behaviors. However, simply feeding vast amounts of sensor data into servers for processing is neither efficient nor sustainable. In line with the prevailing trend towards edge computing, it is becoming increasingly important to process and integrate the captured sensor information directly within the sensor itself. While we have not fully achieved this, the application of deep learning methods to facilitate efficient and rapid processing with low computational demands is a necessary progression. In our study, we used a method for outdoor animal behavior analysis using multi-target classification, taking advantage of the potential efficiency gains provided by deep learning. We focus on a polar bear's behaviors captured by an IoT-enabled surveillance camera in a zoo. The image data are first analyzed by using an object detection model to provide location sequences, movement speed, and coordinates of video frames, representing the animal's state. Using these sensor data, we developed a classification model that accurately classifies multiple behaviors. The detection of these behaviors, including stereotypical behavior, illustrates the potential of our system to comprehensively monitor the animal health status. Our method achieved accurate detection [98.3% average precision (AP) 50] and multi-behavior recognition (accuracy of 89.5%), while maintaining robustness against outdoor noise.

1. Introduction

The increasing integration of sensor technologies in various fields has become a defining trend in recent years, playing an integral role in research, industry, and daily life. Particularly in the domain of ethology, sensor technologies, notably surveillance cameras, are gaining

*Corresponding author: e-mail: ruqin.wang.q3@elms.hokudai.c.jp
https://doi.org/10.18494/SAM4521

ISSN 0914-4935
https://myukk.org/

momentum in the analysis of animal behaviors.[1] A continuous observation facilitated by IoT sensor technologies such as surveillance cameras provides extensive data on animal behavior, contributing significantly to fields such as ecology, wildlife conservation, animal welfare, and animal cognition research.

Recent advances have made it possible to integrate sensors with sophisticated data processing methods such as machine learning, paving the way for the improved analysis of complex patterns and behaviors.[2] This trend aligns with the ongoing shift towards edge computing, where data captured by sensors is processed and integrated directly within the sensor itself, promoting efficiency and real-time processing.[3] As sensor technologies and deep learning techniques advance, more robust, accurate, and rapid processing becomes possible, even with lower computational demands.[4,5]

Deep learning has already made a significant impact in the field of animal behavior analysis, [6] yielding promising results across a multitude of studies. Its ability to handle complex and high-dimensional data makes it particularly well-suited for recognizing patterns in animal behavior, aiding in tasks ranging from species identification to behavior classification.[7] Moreover, the incorporation of IoT devices, such as surveillance cameras, adds an additional layer of sophistication, allowing for continuous data collection and real-time analysis [8]. More researchers are leveraging the effectiveness of deep learning in the field of animal behavior analysis. For instance, the scratching and grooming behaviors of mice were detected using methods based on a convolutional recurrent neural network (CRNN)[7] and a three-dimensional convolutional neural network (3D-CNN),[9] respectively.

In this study, we aim to explore the potential of deep learning techniques, specifically YOLOv5[10] and ResNet-18,[11] in the classification of multiple animal behaviors for efficient health and welfare monitoring. By developing an effective and accurate behavior classification system, we hope to contribute to the understanding of animal behavior, improve animal welfare in captivity, and support ongoing conservation efforts. As deep-learning-based computer vision has demonstrated remarkable performance in various applications, including animal behavior analysis, we believe that our approach can significantly enhance the efficiency and accuracy of animal behavior classification.[7–9]

The contributions of this paper are as follows: first, we propose a simple and efficient background subtraction method that effectively eliminates the interference of the dynamic background and light pollution in subsequent YOLOv5s-based animal detection within real outdoor environments. This method ensures long-term stable and robust animal detection. Second, on the basis of long-term stable and highly accurate detection results, we introduce a method for detecting multiple animal behaviors, including stereotypical behavior. The detected trajectory information along with the cropped images of the polar bear is used as input to the behavior recognition network. The stereotypical behavior of the polar bear is identified by analysis focusing on the periodicity in trajectories. We verify the effectiveness of our methods in a real environment by applying them to videos of a polar bear kept at Sapporo Maruyama Zoo. The proposed method achieves an accurate and robust detection of multiple animal behaviors, providing accurate detection (98.3% AP50) and maintaining high robustness to various noises,

including Gaussian, uniform, and so forth. Additionally, our method enables comprehensive multi-behavior recognition (accuracy 89.5%), demonstrating its effectiveness in analyzing a wide range of animal behaviors.

2. Related Work

2.1 IoT applications in animal behavior analysis

Over recent years, IoT applications in animal monitoring have burgeoned, diversifying research objectives and expanding our understanding of animal behaviors in various contexts. Monitoring migratory patterns of wild animals,[12] analyzing grazing behaviors,[13] studying grazing site profiles, deciphering animal posture behaviors,[14] and detecting animal estrus cycles are among the areas that have benefited from IoT implementations. Most of these studies involve recording animal behavior for subsequent analysis, with a smaller subset focusing on real-time streaming analysis. Researchers have demonstrated the potential of IoT applications for comprehensive and effective animal behavior analysis. For example, one study employed machine learning techniques on GPS traces, collected over four months from 40 cows, to classify grazing behaviors.[13] In another study, we used supervised behavioral classification methods to distinguish between active and inactive behaviors in sheep, resulting in a classification accuracy of above 92% under different conditions.[14] These previous studies have served as the foundational underpinning for the methods we have employed in our current study.

2.2 Animal behavior with deep learning

Animal behavior analysis has become increasingly important across various research fields, as it provides valuable insights into the mental, physical, and cognitive status of experimental animals. Given the diverse range of behaviors exhibited by animals, researchers often focus on specific aspects according to their interests. However, the continuous monitoring and detection of multiple behaviors throughout the day are also essential for a comprehensive understanding of the animal's overall well-being and response to environmental factors.

Deep neural network technologies have recently made a significant impact on animal research to address these challenges. Convolutional neural networks (CNNs), which effectively extract visual features from images, have demonstrated outstanding performance in image classification tasks, including animal behavior analysis. Several studies have shown that CNN-based algorithms can accurately predict animal poses from images,[15] providing a foundation for detecting multiple behaviors simultaneously.

However, many existing methods, such as LEAP[16] and DeepLabCut,[17] primarily focus on pose estimation, which may not be sufficient for detecting a wide range of behaviors in a continuous manner throughout the day. These methods offer promising potential for detecting individual behaviors but may require further development or adaptation to accommodate the simultaneous detection of multiple behaviors in various settings.

2.3 Object detection

Object detection is a crucial task in computer vision and has been widely used in various applications, such as autonomous driving, surveillance, and object recognition. The mainstream object detection methods can be divided into two families: the R-CNN family,[18] from R-CNN, Faster R-CNN, to Mask R-CNN; and the YOLO family,[10] from YOLO v1 to the current v8 version.

The YOLO series, based on deep-learning regression methods, has achieved the advantages of a fast, simple pipeline, low background false detection ratio, and high generality. YOLOv5 has made significant improvements in terms of model size and accuracy. One of the most significant improvements of YOLOv5 is its model size, which is only 27 MB, compared with 244 MB for the previous version YOLOv4 that uses the darknet architecture, making YOLOv5 nearly 90% smaller. YOLOv5's very fast inference and small model size make it ideal for applications with limited computing resources. Moreover, YOLOv5s has been shown to be comparable to YOLOv4 in terms of accuracy, making it an excellent option for real-time object detection tasks.

2.4 Pretrained image encoder for efficient image classification in animal behavior analysis

Pretrained image encoders have recently gained popularity in deep learning, particularly for addressing the challenges of limited data availability in specific tasks such as animal behavior analysis. By utilizing transfer learning, pretrained models on large datasets can improve performance and robustness, enabling the efficient and effective processing of high-dimensional data, such as images, while maintaining the input data quality[19] This strategy aims to directly apply the knowledge acquired by a network model to solve similar problems, such as image-based recognition from small datasets.

The effectiveness of transfer learning is particularly important when dealing with the limited availability of annotated animal behavior data, which is often the case in ethology research. In this context, pretrained encoders can provide a strong foundation for extracting relevant features from animal behavior images and facilitate more accurate classification.

In our study, we aim to explore the potential of deep learning techniques, specifically YOLOv5 and ResNet-18, in classifying multiple animal behaviors in a continuous and efficient manner. By leveraging pretrained image encoders and transfer learning, we aim to address the challenges of limited data availability and develop an effective and accurate behavior classification system. This will contribute to our understanding of animal behavior and support ongoing research in animal welfare and conservation. The use of pretrained image encoders and transfer learning has the potential to enhance the effectiveness of animal behavior classification, particularly in situations where the available annotated data is limited.

3. Methodology

For the fixed viewing of videos, compared with the object as the target of detection, the background of the scene is relevantly invariant through frames. Therefore, before sending the input data to the single-object detection model, we preprocess the image by subtracting the background to detect the moving object easily.

3.1 YOLO-based single model detector

Consider a dataset of candidate images, denoted as D, consisting of input images $x \in R^d$, where $d = Width \times Height \times Channel$, and the corresponding bounding box labels $y \in R^K$, where d represents the size of the image and K denotes the number of box features (a,b,w,h,C). To simplify object detection, we can divide the input image into an $S = s \times s$ grid. Each grid cell is responsible for detecting an object if the center of the object falls within it. Therefore, the object detection model $f : R^d \rightarrow R^{|S|\times|B|}$ maps input images to output predictions for bounding boxes, where $|S|$ is the total number of grid cells and $|B|$ is the number of bounding boxes. For each sample $(x,y) \in D$, the predicted bounding box output $\tilde{y}$ is given by $\tilde{y} = f(x; \theta)$. Finally, the multi-parts loss $L(x, y)$ can be defined.

$$\begin{aligned} L(x,y) = {} & \lambda_{coord} \sum_{i}^{|S|} \sum_{j}^{|B|} \mathbb{I}_{ij}^{obj} \left[(a_i - \tilde{a}_i) + \left(b_i - \tilde{b}_i\right) \right] \\ & + \lambda_{coord} \sum_{i}^{|S|} \sum_{j}^{|B|} \mathbb{I}_{ij}^{obj} \left[\left(\sqrt{\omega_i} - \sqrt{\hat{\omega}_i}\right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i}\right)^2 \right] \\ & + \sum_{i}^{|S|} \sum_{j}^{|B|} \mathbb{I}_{ij}^{obj} \left(C_i - \hat{C}_i\right)^2 \\ & + \lambda_{noobj} \sum_{i}^{|S|} \sum_{j}^{|B|} \mathbb{I}_{ij}^{noobj} \left(C_i - \hat{C}_i\right)^2 \end{aligned} \tag{1}$$

In this study, the bounding box B is defined by its center position (a, b), width (w), and height (h). The confidence score is denoted by C. Thus, the ground truth labels can be expressed as $y_i = [a_i, b_i, w_i, h_i, C_i]$. To balance the penalty of confidence score and the overlap of bounding boxes, λ_{coord} and λ_{noobj} are employed. The terms $\mathbb{I}_{ij}^{obj}$ and $\mathbb{I}_{ij}^{noobj}$ indicate whether an object is truly detected within the predicted bounding box or not, respectively.

The YOLO network is optimized to minimize the loss function as the training objective to achieve the detection of the objects. The YOLO network consists of convolutional layers specifically designed for object detection tasks and comprises three critical modules, as illustrated in Fig. 1:

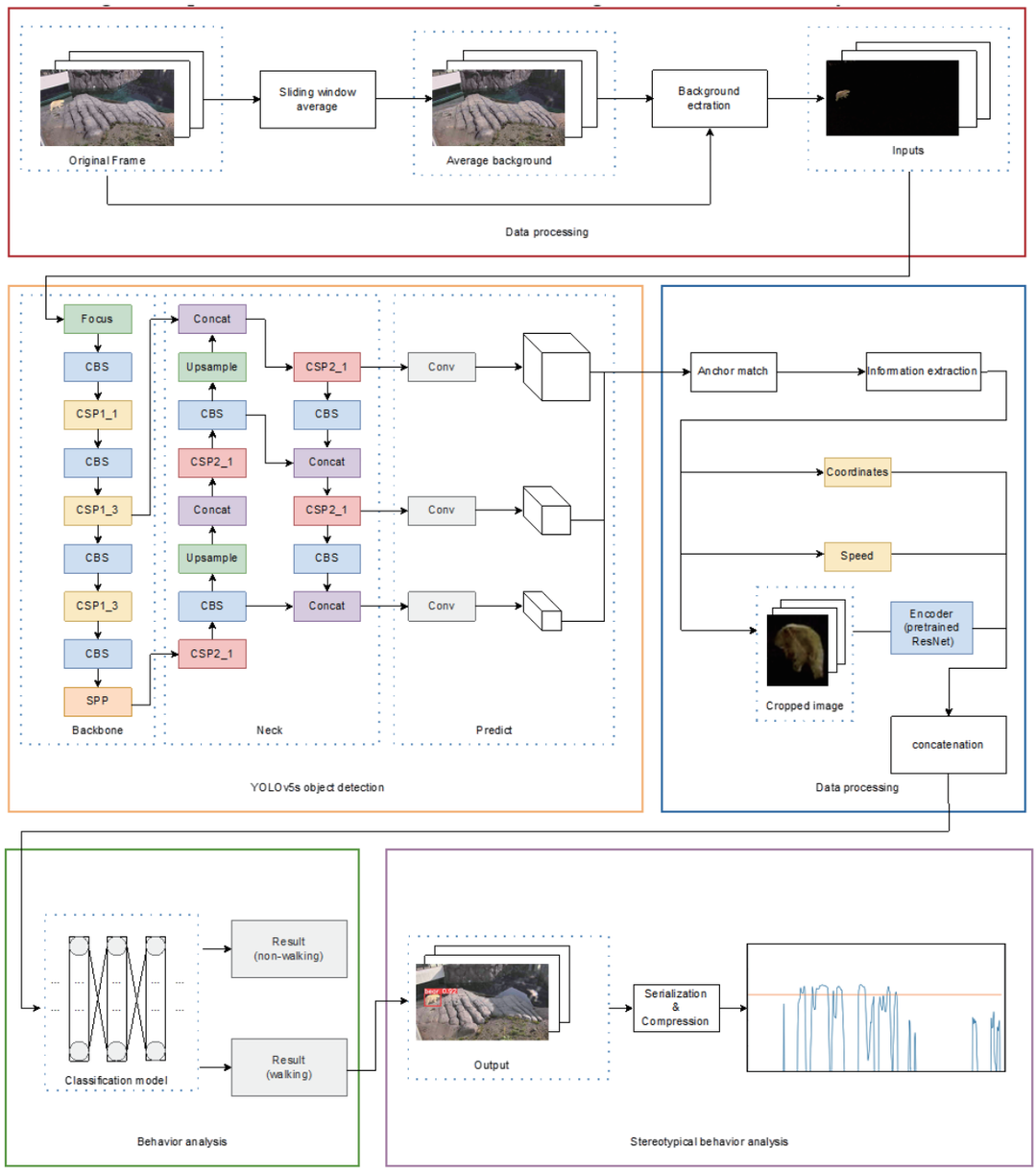


Fig. 1. (Color online) Overview of YOLO-based single model for animal behavior recognition.

1. **Backbone:** This module serves to extract generic feature representations and functions as a high-performance classifier network.
2. **Neck Network:** Situated between the backbone network and the prediction network, the neck network enhances the diversity and robustness of the extracted features.
3. **Prediction:** Three convolutional networks work together to produce the final output of the object detection results.

3.2 Background subtraction as preprocessing

In videos with a fixed viewpoint, the background remains relatively consistent across frames, while the target object of detection undergoes changes. Consequently, it is advantageous to preprocess the input data by subtracting the background, which facilitates the detection of the moving object by the single-object detection model.

A straightforward method for estimating the background image involves averaging all frames in the dataset. However, our polar bear dataset features a diverse range of background conditions, such as daytime, nighttime, and snowy scenes. Thus, a more effective and proactive approach is to estimate the background on the basis of temporally proximate frames rather than using the entire dataset. We use a sliding window technique to extract the current background context. Assuming a sliding window of length T, the background can be obtained using the following equation:

$$X_{avg} = \frac{1}{T}\sum_{t}^{T} x_t \,. \tag{2}$$

During our experiments, we discovered that a sliding window with $T = 30$ yields favorable results. In Sect. 5 of our experiment, we will evaluate the efficacy of this preprocessing approach.

3.3 Animal's stereotypical detection

In this study, one aspect we focus on is the detection of stereotypical behavior in polar bears, which is often characterized by repetitive and cyclical actions. However, it is essential to note that this is only part of our overall research objectives. Within this specific component, we emphasize the periodicity of the stereotypical behavior and disregard the detailed position information, which could introduce irrelevant noise when detecting periodicity.

To achieve this, the image space is divided into a 3 × 5 grid, and the detected position of the polar bear is assigned to one of the grid cells. Each grid cell is assigned a unique identifier, represented by a letter (A to O). Subsequently, the trajectory of the detected position is transformed into a sequence of letters corresponding to the grid cells. It is anticipated that stereotypical behavior will generate a periodic series of letters, as illustrated in Fig. 2.

The periodicity is quantitatively evaluated by using a compression algorithm. We first apply the compression algorithm to the converted letter sequence. Then, the compression ratio r is calculated as

$$\boldsymbol{r} = \frac{\boldsymbol{L} - \boldsymbol{L}'}{\boldsymbol{L}}. \tag{3}$$

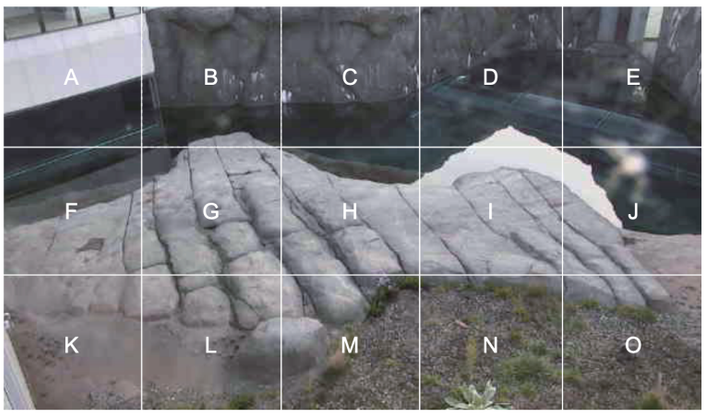


Fig. 2. (Color online) Grids above the image for locating the bear.

The compression ratio is determined by the number of bytes in the string before compression (***L***) and the number of bytes after compression (***L′***). Compression algorithms exhibit a characteristic where the compression ratio increases for consecutive or cyclic strings.

Consequently, a high compression ratio can serve as an indicator of stereotypical behavior in polar bears. However, when polar bears are at rest for short or long durations, a continuous string of identical characters forms, leading to a similar increase in compression ratio as observed in stereotypical behavior. In such cases, the compression ratio tends to be exceptionally high.

To mitigate the issue of excessive text compression rates when polar bears are stationary, successive identical characters in a string sequence are removed. Additionally, the special letter Z is employed to denote the absence of polar bears. Because of the proposed method, translating into a continuous sequence of the letter Z and ultimately condensing into a single letter Z ensure that the final detection results remain unaffected.

Under low light conditions, such as early morning and night, as well as when polar bears are in the pool, discontinuous detection is prone to occur. The sporadic appearance and disappearance also result in a high text compression ratio. To address this challenge, we first employ an animal detection model to identify instances of walking behavior. Then, we apply a moving average method to stabilize fluctuations in compression ratio for the detected walking instances. Finally, an upper threshold is imposed to restrict the elevated compression ratio, allowing for the identification of stereotypical behavior within the walking segments.

3.4 Animal behavior recognition model

In this study, we introduce an animal behavior recognition model based on the YOLOv5s framework for detecting animal targets. Owing to the remarkably high accuracy of YOLOv5s detection, we can confidently employ a classification approach that builds upon its outputs. After

obtaining the detection results from YOLOv5s, we extract the relevant information, including the target's center coordinates, current motion speed, and cropped images using the bounding box as a reference.

Our goal is to employ a simpler and more efficient model to achieve the rapid and accurate classification of animal behaviors. Therefore, before feeding the data into the classification model, we utilize a pretrained ResNet as an encoder to extract features and reduce the computational load. The reason for using ResNet as an encoder lies in its ability to minimize the computational burden while accurately extracting features. Subsequently, the output from ResNet is combined with two scalar values (center coordinates and motion speed), which then serve as the input for the classification model during training. This methodology ensures the development of a robust animal behavior recognition model that leverages the strengths of both YOLOv5s and ResNet, resulting in efficient and accurate classification performance.

We implement a fully connected neural network (FCN) classification model, which takes the position coordinates, movement velocity, and the feature vector extracted from an encoder as inputs, and outputs the classification of animal models. To measure the discrepancy between the actual and predicted categories, we employ the cross-entropy loss function, which is widely used in classification tasks. The loss function is expressed as

$$\boldsymbol{L} = -\frac{\mathbf{1}}{\boldsymbol{N}} \sum_{i=1}^{N} \boldsymbol{y}_i \log\left(\hat{\boldsymbol{y}}_i\right) \tag{4}$$

Here, $\boldsymbol{y_i}$ denotes the one-hot encoding of the actual category and $\hat{\boldsymbol{y}}_i$ represents the predicted probability distribution. In the FCN model, the input comprises position coordinates $\boldsymbol{p_x}$, $\boldsymbol{p_y}$, movement velocities $\boldsymbol{v_x}$, $\boldsymbol{v_y}$, and the feature vector $\boldsymbol{F}$ extracted by the encoder. These inputs can be concatenated to form a larger input vector:

$$\boldsymbol{I} = \left[\boldsymbol{p}_x, \boldsymbol{p}_y, \boldsymbol{v}_x, \boldsymbol{v}_y, \boldsymbol{F}_1, \boldsymbol{F}_2, \ldots, \boldsymbol{F}_n\right]. \tag{5}$$

$\boldsymbol{n}$ signifies the dimension of the feature vector generated by the encoder. In our model, we employ a pretrained ResNet18 model from PyTorch as the encoder and $n = 7 \times 7$. This value is dictated by the design of the ResNet architecture, which generates a 7×7 output feature map in its final layer. The extended input vector is subsequently passed through the FCN, resulting in the output probability distribution $\hat{\boldsymbol{y}}$. With this probability distribution and the one-hot encoding of the actual category at hand, the cross-entropy loss function L can be computed.

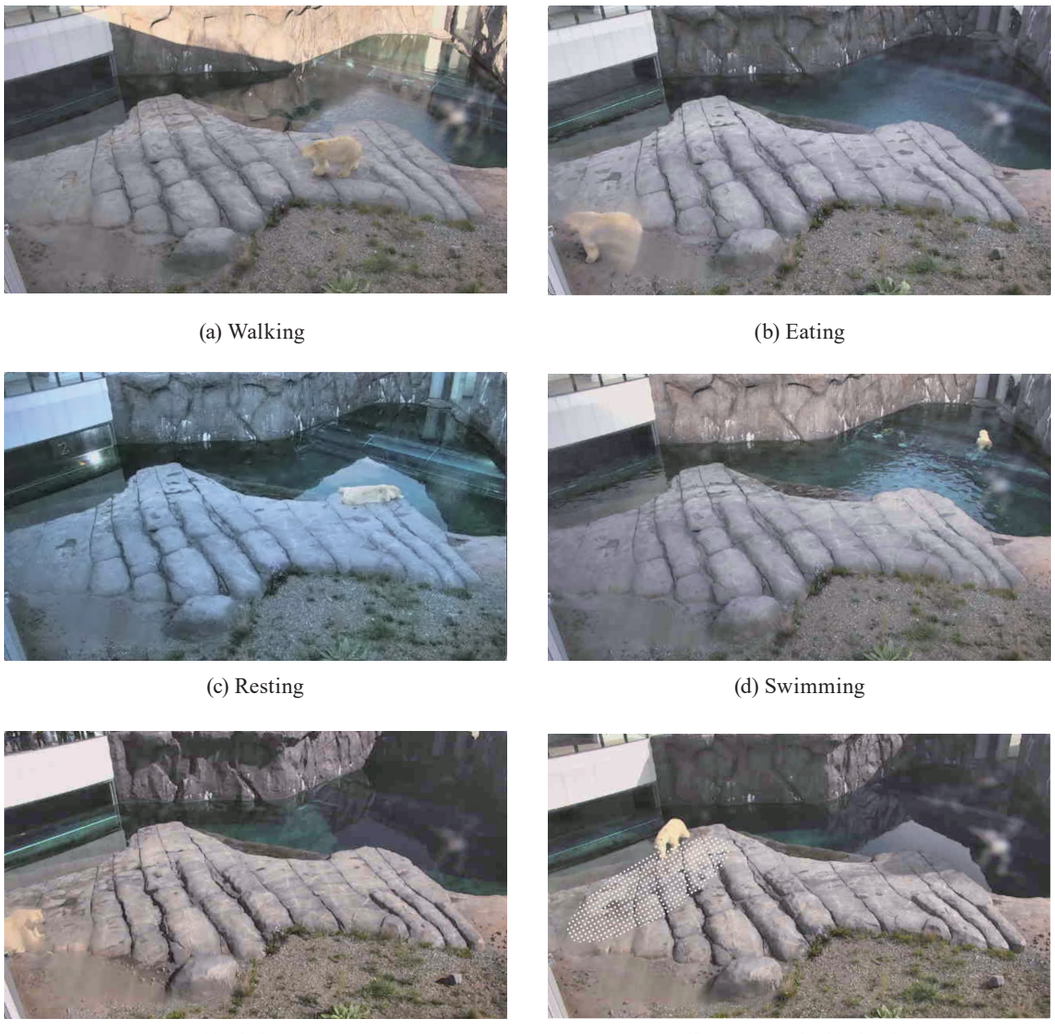
4. Experiment

In this section, we focus on the use of the Sapporo Maruyama Zoo polar bear dataset and the experimental setup for the YOLOv5s model. We also discuss the detection settings used for stereotypical behavior, highlighting the importance of accurate and efficient detection methods.

4.1 Dataset

The polar bear dataset used in this study is provided by Sapporo Maruyama Zoo. Figure 3 illustrates examples of frames from the videos. The footage is captured by one of the four surveillance cameras installed in the polar bear enclosure. As depicted in Fig. 3, the polar bear exhibits various behaviors in different situations, such as walking, eating, resting at night, and swimming in a pool. In addition to these behaviors, the polar bear also engages in sitting and stereotypical behaviors, which constitute the six primary actions we aim to detect in this study.

The dataset employed in this experiment comprises a continuous video and two types of label. One label pertains to the polar bear's location, providing information on the center point, top-left and bottom-right coordinates, as well as the length and width of the bounding box. The



(a) Walking (b) Eating

(c) Resting (d) Swimming

(e) Sitting (f) Stereotypical behavior

Fig. 3. (Color online) Frame of the video used as the dataset.

other label indicates the behavior of the polar bear, denoting whether it is performing a periodic movement (stereotypical behavior) or any of the other five behaviors. We annotated the periods when the polar bear exhibited constant motion as stereotypical behavior. These labels are assigned to each frame in the videos, with annotation performed by a human annotator under the guidance of zookeepers. In our experiment, we consider the repetitive pacing of the polar bear in the white-shaded area shown in Fig. 3(f) as an indication of the animal's stereotypical behavior. The long-duration pacing in this specific location serves as a basis for detecting and analyzing the stereotypical behavior of the polar bear in our dataset. In Fig. 3(a), where there is no specific location and apparent repetitive behavior in the polar bear's walking, we categorize these instances as normal walking behavior. In this context, we consider the polar bear's stereotypical behavior to be a special manifestation within its walking behaviors.

Recognizing the complexity of animal behavior, we aimed to capture the nuances of polar bear activities through a robust labeling process. We focused on visible physical activities such as walking, swimming, and eating, understanding that these categories may not encompass all possible behaviors or variations. To ensure a comprehensive and accurate representation, we involved seasoned zookeepers and multiple annotators, allowing for diverse perspectives and cross-verification. Despite these measures, we acknowledge that some level of uncertainty may persist in distinguishing complex or subtle behaviors, a challenge we aim to address as we refine our methodology.

We utilized pretrained COCO weights as the backbone for YOLOv5s. We adjusted the final prediction layer to accommodate our detection targets and further fine-tuned the model using the polar bear dataset. For fine-tuning YOLOv5s for polar bear detection, we utilized the video footage captured by one camera between 12:00 a.m. on August 26, 2020 and 24:00 a.m. on August 31, 2020. To train the behavior recognition model, we collected video data between September 22 and 28, 2020, from 05:00 to 20:00 each day. The complete dataset contains approximately 3200 frames per hour. The video features approximately have five frames per second, with a frame size of 368 × 640.

The polar bear is housed individually, which means that only one individual appears in the camera view. Moreover, the likelihood of mistakenly detecting people is relatively low, as zoo visitors observe the polar bear through tunnels set into the pool. Although zookeepers can enter the camera's visible area, their presence is infrequent.

4.2 Settings for fine-tuning model

In our experiments, we scale all training set images to a size of 640 × 640. Throughout the training process, we employ general data preprocessing techniques, as described in Ref. 20, such as random flipping, geometric distortion, light distortion, image masking, random erasing, cropping, and blending. Our implementation is based on the PyTorch framework, and we conduct both training and testing on a single NVIDIA RTX A5000 GPU. To optimize our network, we utilize the stochastic gradient descent (SGD) optimizer with a learning rate of 0.01 and a momentum of 0.9. Additional experimental hyperparameters are detailed in Table 1.

The behavior recognition model consists of a fully connected neural network, with ReLU activation functions for the hidden layers and a SoftMax activation function for the output layer.

Table 1
Detailed settings for fine-tuning the YOLOv5s model.

Parameter	Value
Learning	0.01
Learning rate decay	0.999
Learning rate decay step	1
Weight rate decay	5e−4
Momentum	0.937
Batch size	64
Number of epochs	50

To mitigate overfitting, dropout layers are incorporated between the hidden layers. The ResNet-18 encoder is employed to extract high-level features from the input images.

In our study, we employed the PyTorch deep learning framework[14] to implement our model. Specifically, we used the pretrained ResNet-18 model as the feature extractor, which was originally trained on the ImageNet dataset. To adapt the pretrained model to our animal behavior recognition task, we fine-tuned the last few layers of the ResNet-18 model. By leveraging the pretrained ResNet-18 model and the PyTorch framework, we were able to efficiently and effectively train our model for the classification of various animal behaviors.

For the training process, we use the cross-entropy loss function to optimize the classification model and also the SGD optimizer with specified learning rate and momentum values. A detailed overview of the behavior classification model configurations and training hyperparameters is shown in Table 2.

4.3 Setting for stereotypical behavior's serialization and compression

For monitoring the health condition of polar bears, it is desirable to be able to discriminate the stereotypical behavior at a time interval. Thus, to evaluate the stereotypical behavior, the detected trajectories of the polar bear were divided every 160 frames, which is three minutes of real time, and converted into a string. This enables us to determine whether the polar bear's behavior is stereotypical every 3 min.

5. Results and Analysis

In this section, we present the detection and behavior recognition results obtained by our method using the Sapporo Maruyama Zoo polar bear dataset. First, we present the training results of the YOLOv5s model, which showed high accuracy and low background false detection rate. Then, the results of behavior recognition including both stereotypical and other behaviors are presented.

Table 2
Overview of classification model aspects.

Classification model	
Architecture	FCN
Activation function (hidden layers)	ReLU
Dropout rate	0.5
Output layer activation	SoftMax
Encoder	
Architecture	ResNet-18
Pretrained	yes
Training	
Loss function	Cross-entropy loss
Optimizer	SGD
Learning Rate	0.01
Momentum	0.9

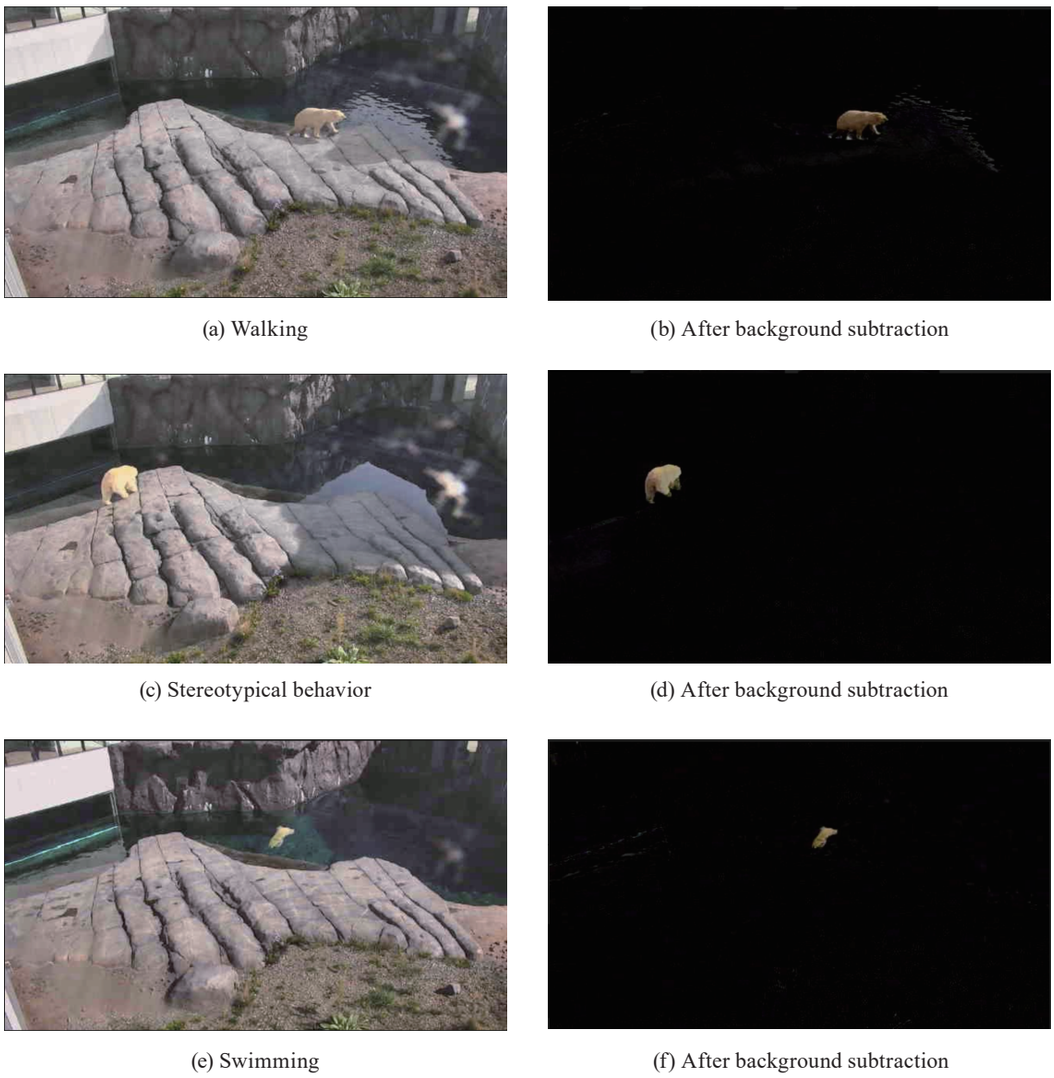
5.1 Preprocessing and detection of animal's position

Before starting the formal training, we preprocessed the video frames. Figure 4 shows the effectiveness of background subtraction across various scenarios involving polar bears, particularly when they are in motion. Figures 4(a), 4(c), and 4(e) show a polar bear performing different actions in various scenarios, whereas Figs. 4(b), 4(d), and 4(f) present the same scenes after applying background subtraction. Our data preprocessing method successfully filters out background elements unrelated to the detection object and proves especially useful for detecting moving polar bears.

Figure 5 depicts the training curve, illustrating the loss values for both the training and test sets. The test loss is presented in Fig. 5 to indicate the fine-tuned model's effectiveness in locating the object's center and the precision with which the predicted bounding box covers the object. The model exhibits rapid improvement in terms of precision, recall, and average precision, with performance converging after approximately 50 epochs. The test loss also displays a rapid decrease until about 50 epochs. We employ early stopping to select the optimal weights.

To better show our experimental results, we visualized examples of the detection results in Fig. 6. The examples demonstrate that the polar bear was detected with a high degree of confidence. Notably, even when the polar bear was swimming in the pool and appeared small within the image frame, detection was successful. This successful detection can be attributed to the use of preprocessing. We will further validate the effectiveness of preprocessing in the subsequent section.

The primary cause of erroneous detection is intermittent illumination reflection that affects video data, as shown in Figs. 7(a) and 7(b) for daytime and nighttime scenarios, respectively. Our preprocessing step, which includes a background subtraction algorithm, has significantly mitigated this issue, but a minimal number of misclassifications still arise. In Fig. 7(a), during



(a) Walking (b) After background subtraction

(c) Stereotypical behavior (d) After background subtraction

(e) Swimming (f) After background subtraction

Fig. 4. (Color online) Comparison of video frames before and after.

the day, sunlight reflects off the glass, leading to misclassifications. Similarly, in Fig. 7(b), artificial lighting at night can also cause reflections that disrupt accurate detection. These detection errors exhibit noncontinuous, random occurrences, making them particularly challenging to address. The instances of erroneous detection, while low, can impact the subsequent action recognition accuracy, demonstrating the critical need for ongoing optimization and improvements in the detection algorithm.

5.2 Robustness of detection

To evaluate the performance of our proposed method in terms of robustness, we trained the detection models with four different noise conditions. Figure 8 shows the four different noises added to the original video frames. Figure 8(a) shows the Gaussian noise condition with mean

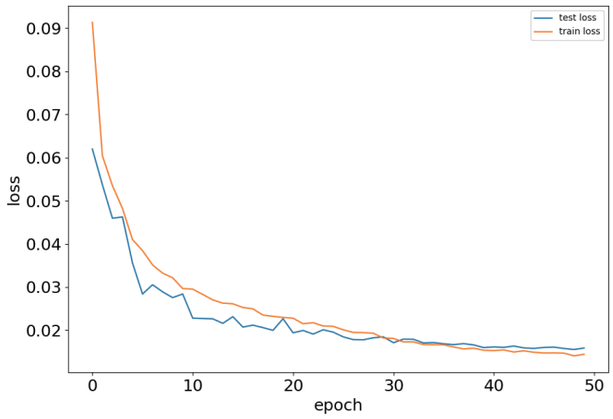


Fig. 5. (Color online) Learning curve during fine-tuning of the YOLO-based model on our dataset.

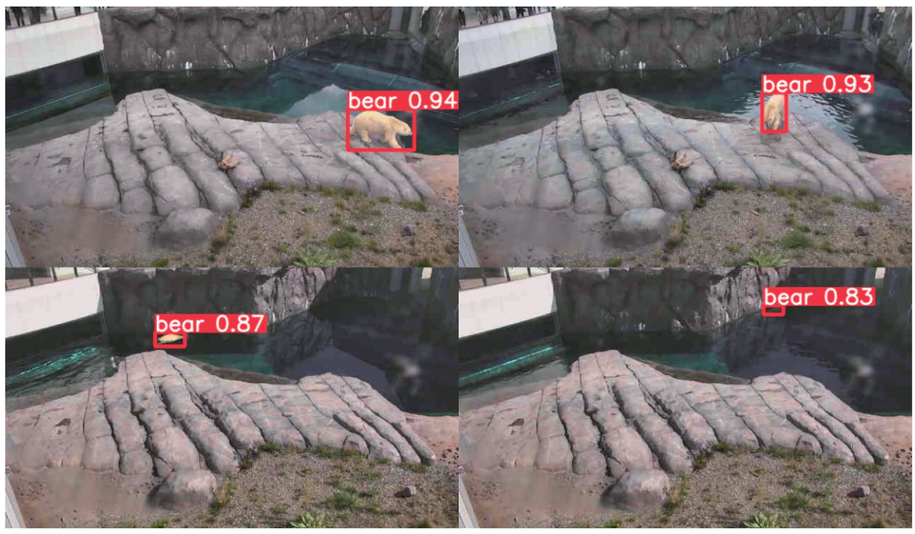
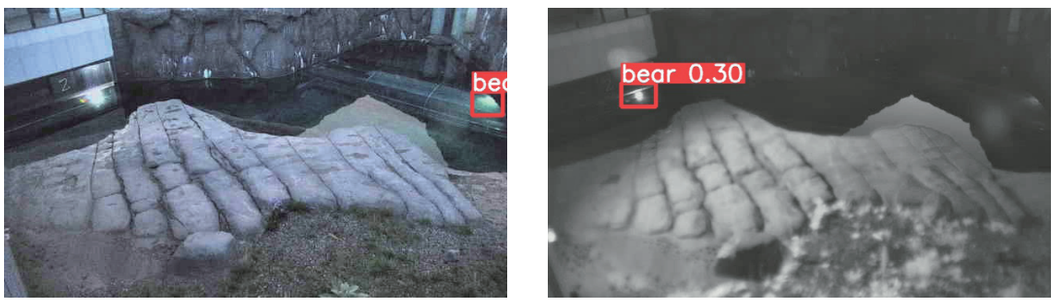
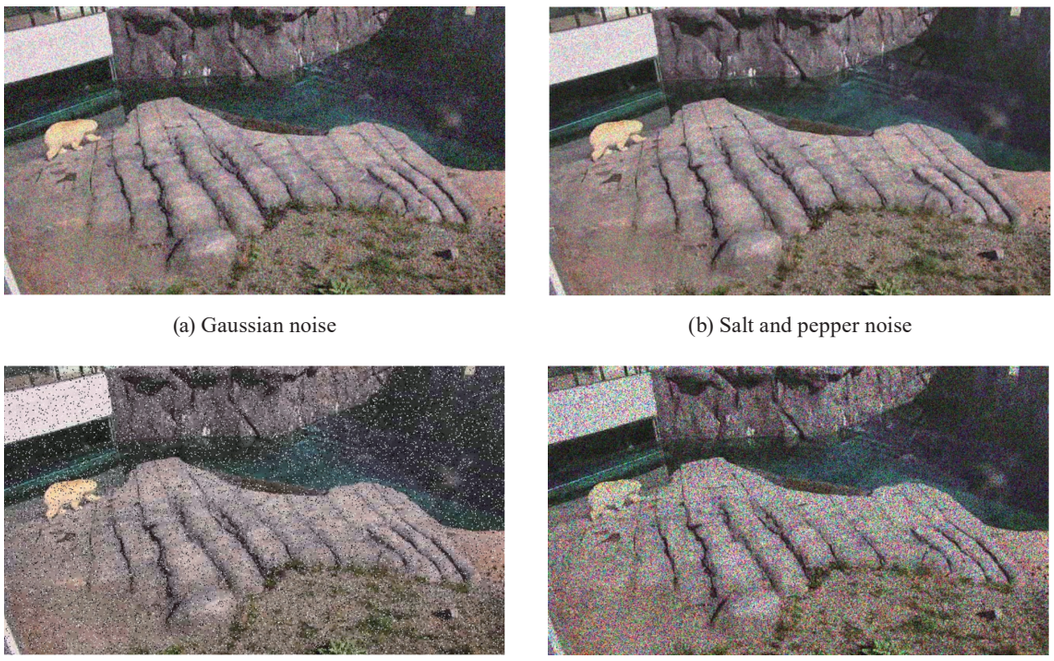


Fig. 6. (Color online) Our detection experimental results, given a video frame, output position information, and confidence.



(a) (b)

Fig. 7. (Color online) Our erroneous (a) daytime and (b) nighttime detection experimental results.



(a) Gaussian noise (b) Salt and pepper noise

(c) Poisson noise (d) Speckle noise

Fig. 8. (Color online) Adding Gaussian (a), salt and pepper (b), Poisson (c), and speckle (d) noise to the original frame for testing the robustness of detection.

$\mu = 0$ and variance $\sigma^2 = 10$, which increases the values on three different channels. Figure 8(b) shows the salt and pepper noises where the probabilities of setting each pixel to white and black are both 0.05. Figure 8(c) is the Poisson noise, and we set both the mean and the variance to 0.8. Figure 8(d) is the speckle noise, which differs from the other noises; this type of noise is determined by the image data: each pixel value in the image is scaled by random noises. For scaling random noises, we use mean $\mu = 0$ and $\sigma^2 = 1$. Half of the pixels in an image are scaled by the noises.

Table 3 shows the comparison of the average precision (AP) values of the detection results obtained by the models trained with or without data preprocessing and noise added to the video frame. Here, "Baseline" means the dataset without noise and without preprocessing for training, "Preprocessing" means the dataset without noise and with preprocessing for training, and "Noise Train" means the dataset with noise and without preprocessing for training.

In the first row, we use 20,000 video frames with continuous time as a train data set. In the second row, 1000 randomly selected discontinuous video frames are used as the training and test sets to obtain the results. These second results are provided to show the performance in the case of small data volume. In the last four rows, we compare the detection results with and without preprocessing for the noise-added frames. The detection models were trained using two datasets: one with 2000 continuous-time original video frames and another with 2000 preprocessed video frames. All models were initialized with weights from a pretrained YOLOv5s model.

Table 3
Results of AP values for polar bear detection under different conditions.

Type	Preprocessing	Baseline	Noise train
20000 samples	0.983	0.991	—
1000 samples	0.974	0.907	—
Gaussian noise	0.759	0.850	0.792
Salt and pepper noise	0.921	0.774	0.823
Poisson noise	0.922	0.938	0.921
Speckle noise	0.870	0.497	0.675

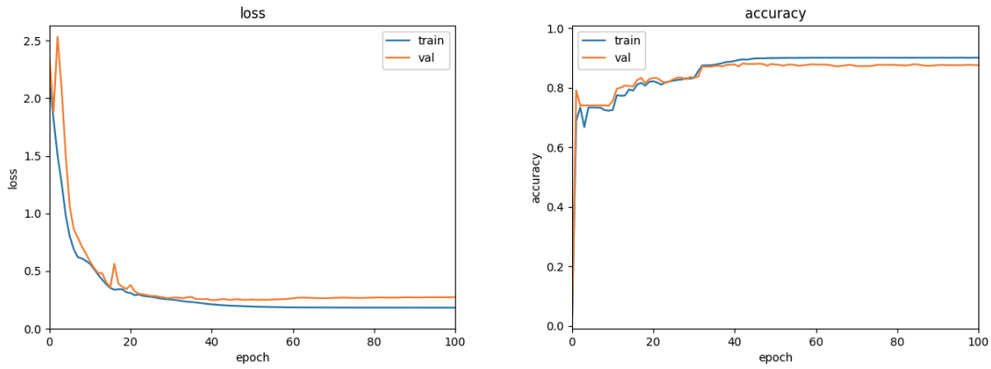
The results in Table 3 show that the recognition results on the completed training model after using data preprocessing are generally better than those without data preprocessing. Even if the model is trained using noisy data in the Noise Train setting, it cannot achieve the same recognition accuracy as that after preprocessing. Here, we believe that the preprocessing approach has good robust performance corresponding to different noises.

5.3 Recognition of animal behaviors

In this study, we perform both the recognition of various behaviors and the detection of stereotypical behaviors exhibited by polar bears. We employed a behavior recognition model to identify five different types of behavior, namely, swimming, walking, eating, sitting, and resting. The behavior recognition model was trained using a dataset containing instances of these five behaviors, and its performance was evaluated by monitoring the decrease in loss during training.

Figure 9 shows the training curve of the behavior classification model, illustrating the loss values during training. The loss value started at 2.5 and showed a significant decrease within the first 100 epochs, indicating that the model achieved satisfactory performance in a relatively short training period. This rapid decrease in loss demonstrates the model's effectiveness in learning the underlying patterns and distinguishing among the five types of behavior.

We performed a comparative experiment to show the impact of integrating image features, extracted by the encoder, into our model. The experiment involved training the FCN model on differing input sets: exclusively the coordinates, solely the speed information, and a combination of both the coordinates and the speed information. As the experiment was designed, these control groups did not utilize any image information and hence did not use a ResNet encoder. As delineated in Table 4, the experimental outcomes underscored our proposed method's superior performance. The highest accuracy achieved by the FCN model was 71.9%, significantly below 89.5% achieved by our method. Our model's distinct advantage stems from the amalgamation of three different types of information: coordinates, speed, and image features extracted via the ResNet encoder. This comparison clearly demonstrates that the integration of image features, derived from the ResNet encoder, has significantly contributed to the increased accuracy of our model in animal behavior classification. This increase highlights the effectiveness of our proposed method.



(a) (b)

Fig. 9. (Color online) Learning curve of the classification model on our dataset.

Table 4
Comparison of classification accuracy across different input types and models.

Epoch	Coordinates	Speed	Coordinates & speed	Coordinates & speed & image (Our Model)
50	65.4	62.3	67.4	87.2
100	67.2	64.6	71.9	89.5

After detecting the walking behavior, we then detected the stereotypical behavior (Fig. 10). Firstly, the polar bear trajectory was serialized into strings according to the detected positions. Then, compression was applied to the strings, and the result was evaluated for detecting the stereotypical behavior. Figure 11 presents the results of detecting the walking and non-walking behaviors before applying the detection algorithm. Figure 12 shows the results after stereotypical behavior detection.

To perform the stereotypical behavior detection, the threshold was changed from 0.0 to 0.8 in increments of 0.1 and compared with the ground truth labels on September 22. We set the compression ratio of 0.6 as the threshold, and the time above this threshold was considered the time of stereotypical behavior. Then, we evaluated the accuracy of the stereotypical behavior detection. The results showed that when using a moving average with a length of 500 frames, the accuracy reached 0.906.

Figure 13 indeed provides a visual representation of the detection results for the six different behaviors of the polar bear throughout the day, with a comparison between the predicted and actual behaviors. This juxtaposition allows for an effective evaluation of our model's performance.

One crucial factor we need to take into account is the effects of environmental elements, specifically the lighting conditions, on the accuracy of behavior detection. A prominent instance of this can be observed in Fig. 13, around 3 PM. Here, an incorrect detection occurred owing to the reflection of sunlight on a rock, leading to a prolonged period of error in our analysis. We have identified that abrupt changes in lighting conditions, such as sudden shifts in illumination

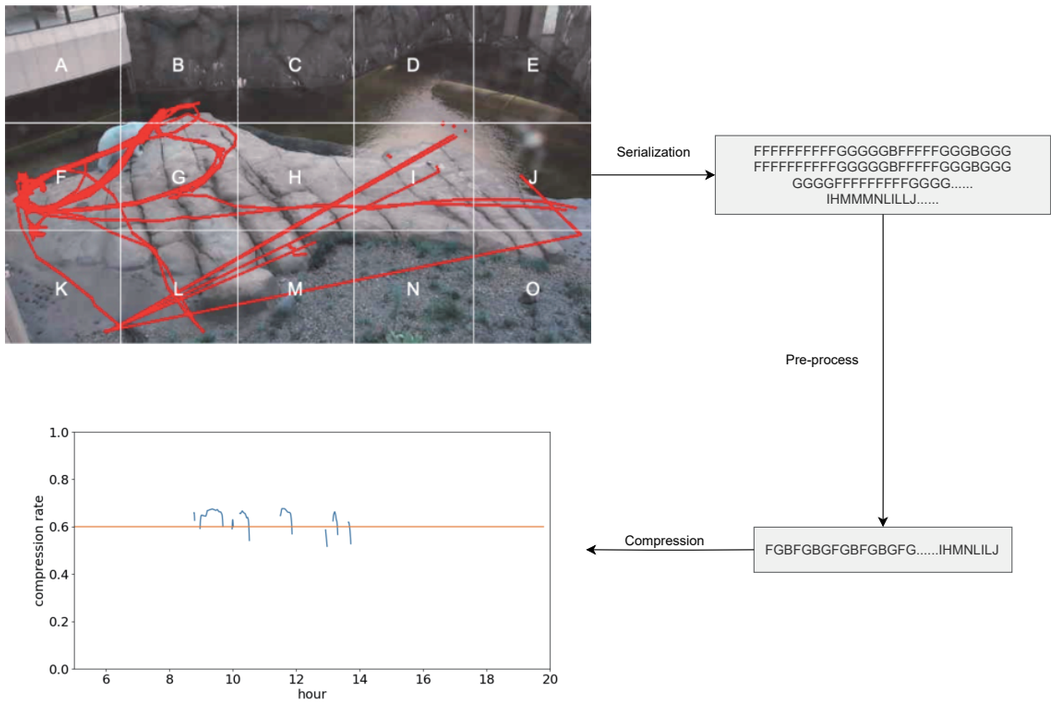


Fig. 10. (Color online) Stereotypical behavior's serialization and compression.

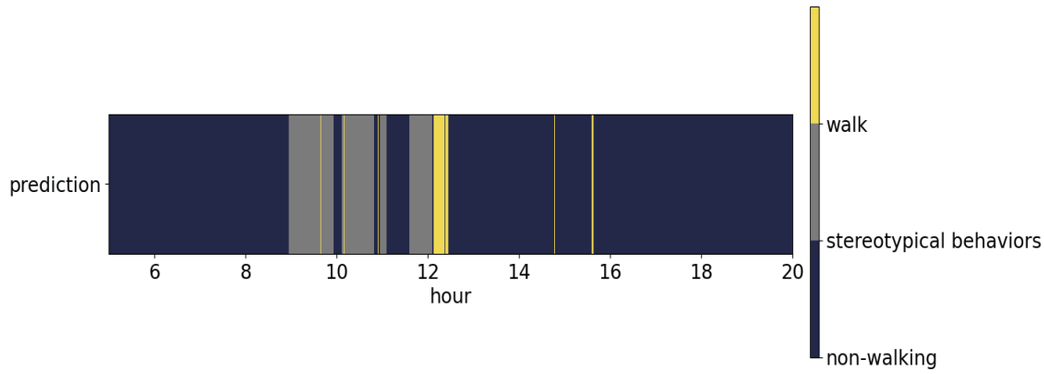


Fig. 11. (Color online) Before stereotypical behavior detection. The yellow segments represent the walking periods, whereas the dark blue segments indicate the time spent on activities other than walking.

due to weather variations, can introduce noise into our detection system. In particular, when light reflections on objects in the animal's environment, like a rock, are erroneously detected as part of the animal, it can momentarily affect the animal's perceived location. This misinterpretation consequently affects the speed estimation, resulting in incorrect behavior detection. Therefore, while our model maintains high overall accuracy, there are instances where environmental factors such as lighting can pose challenges to the detection process.

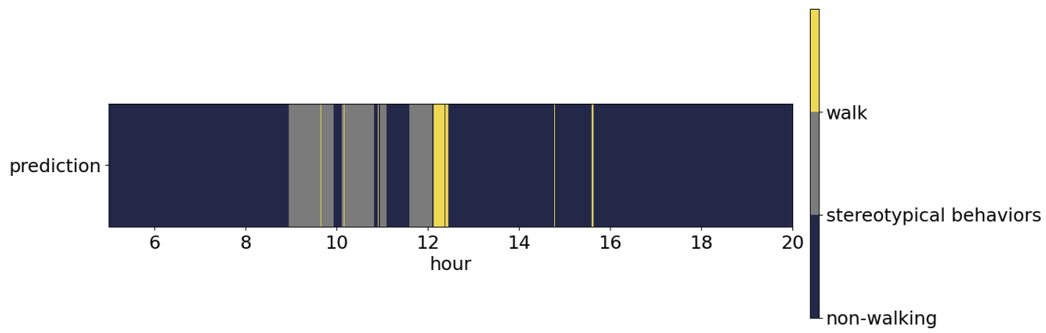


Fig. 12. (Color online) After stereotypical behavior detection. The yellow segments represent the walking periods, the grey segments indicate the time spent on stereotypical behaviors, and the dark blue segments represent the time dedicated to activities other than walking.

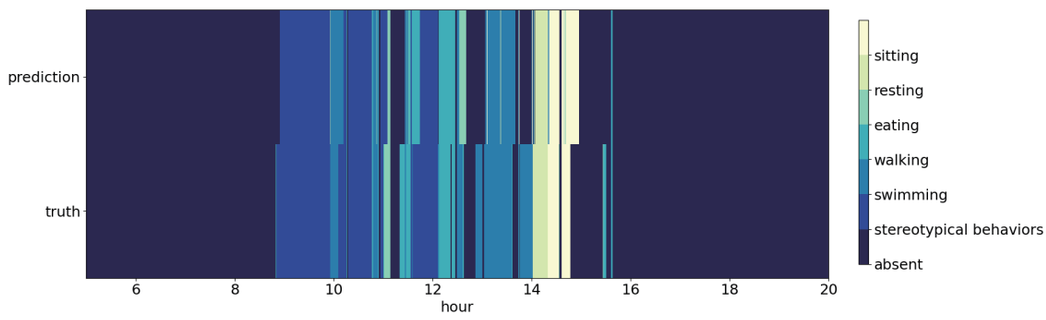


Fig. 13. (Color online) Comparison of predicted and ground truth full day behavior detection results.

Such variations can lead to inaccuracies in speed calculation, which subsequently affect the behavior classification output of our model. Despite our preprocessing steps to mitigate this issue, sporadic discrepancies can still occur. It is important to understand that these limitations represent common challenges in outdoor animal behavior analysis, and continuous improvements to our method will aim to further minimize their impact on the classification accuracy.

6. Conclusion and Future Work

In this study, we presented a comprehensive method for leveraging IoT sensor technologies, specifically surveillance cameras, to detect and analyze the daily behavior of animals, particularly polar bears. Recognizing and understanding these behavioral patterns are crucial for efficient health monitoring and timely intervention in animal welfare. Our method integrates a deep-learning-based object detection model, YOLOv5s, with a behavior classification model, turning raw video data into valuable insights about the animal's activities. Our data preprocessing technique involving background subtraction has effectively minimized the effect of dynamic background noise, enhancing the detection performance of our model.

Nevertheless, there remain limitations to address, such as the limited number of identifiable behaviors and challenges in analyzing brief, nontypical, or subtle behaviors. Furthermore, the unique potential of IoT technology to provide real-time analysis and remote monitoring is yet to be fully utilized. In our future work, we plan to expand the range of detectable behaviors and enhance the descriptiveness of the models. Additionally, we aim to incorporate more IoT capabilities to improve our system's real-time functionality, enabling prompt responses to critical changes in animal behavior. We also intend to apply our method to other animal species, broadening its applicability and impact.

By demonstrating the effectiveness of deep learning and IoT sensor technologies in detecting and analyzing animal behavior, we provide valuable tools for researchers, animal caregivers, and wildlife conservationists. As we address the existing limitations and continue refining our models, we hope to contribute significantly to the development of more effective, efficient, and responsive animal health monitoring systems in the future.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 22H03637. The authors wish to thank Sapporo Maruyama Zoo for providing the animal data.

References

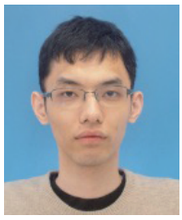
1 E. J. Bethell: J. Appl. Anim. Welfare Sci. **18** (2015) S18. https://doi.org/10.1080/10888705.2015.1075833
2 L. Nóbrega, A. Tavares, A. Cardoso, and P. Gonçalves: 2018 IoT Vertical and Topical Summit on Agriculture-Tuscany (IOT Tuscany, IEEE) 1–5.
3 D. J. Anderson and P. Perona: Neuron **84** (2014) 18.
4 B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba: 2016 IEEE Conf. Computer Vision and Pattern Recognition (CVPR, IEEE) 2921–2929.
5 S. Zhang, C. Zhang, and Q. Yang: Appl. Artif. Intell. **17** (2003) 375. https://doi.org/10.1080/713827180
6 M. S. Dawkins: Behav. Brain Sci. **13** (1990) 1. https://doi.org/10.1017/S0140525X00077104
7 K. Kobayashi, S. Matsushita, N. Shimizu, S. Masuko, M. Yamamoto, and T. Murata: Sci. Rep. **11** (2021) 658. https://doi.org/10.1038/s41598-020-79965-w
8 H. Chen, Z. He, B. Shi, and T. Zhong: IEEE Access **7** (2019) 157818.
9 N. Sakamoto, K. Kobayashi, T. Yamamoto, S. Masuko, M. Yamamoto, and T. Murata: Front. Behav. Neurosci. **16** (2022) 797860. https://doi.org/10.3389/fnbeh.2022.797860
10 J. Redmon, S. Divvala, R. Girshick, and A. Farhadi: 2016 Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR, IEEE) 779–788.
11 K. He, X. Zhang, S. Ren, and J. Sun: 2016 Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR, IEEE) 770–778.
12 J. Hunter, C. Brooking, W. Brimblecombe, R. G. Dwyer, H. A. Campbell, M. E. Watts, and C. E. Franklin: 2013 IEEE 9th Int. Conf. e-Science (IEEE 9th Int. Conf. e-Sci) 140–147.
13 M. L. Williams, N. Mac Parthaláin, P. Brewer, W. P. J. James, and M. T. Rose: J. Dairy Sci. **99** (2016) 2063. https://doi.org/10.3168/jds.2015-10254
14 C. Umstätter, A. Waterhouse, and J. P. Holland: Comput. Electron. Agric. **64** (2008) 19. https://doi.org/10.1016/j.compag.2008.05.004
15 R. Clubb and G. J. Mason: Appl. Anim. Behav. Sci. **102** (2007) 303. https://doi.org/10.1016/j.applanim.2006.05.033
16 A. Krause, S. Neitz, H. J. Mägert, A. Schulz, W. G. Forssmann, P. Schulz-Knappe, and K. Adermann: FEBS Lett. **480** (2000) 147. https://doi.org/10.1016/S0014-5793(00)01920-7
17 A. Mathis, P. Mamidanna, K. M. Cury, T. Abe, V. N. Murthy, M. W. Mathis, and M. Bethge: Nat. Neurosci. **21** (2018) 1281. https://doi.org/10.1038/s41593-018-0209-y

18 R. Girshick: 2016 Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR, IEEE) 580–587.
19 F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, and Q. He: 2020 Proc. IEEE (Proc. IEEE) 43–76.
20 F. Zhou, H. Zhao, and Z. Nie: 2021 IEEE Int. Conf. Power Electronics, Computer Applications (ICPECA, IEEE) 6–11.

About the Authors

Ruqin Wang received her B.S. degree in information science and technology from Ritsumeikan University, Japan, in 2022. She is currently pursuing her M.S. degree in information science and technology at Hokkaido University, Japan, and is in her second year of study. Her primary research interest is in the application of artificial intelligence (AI) to ethology, focusing on understanding animal behavior and improving animal welfare through advanced technology. (ruqin.wang.q3@elms.hokudai.ac.jp)

Wataru Noguchi received his Ph.D. degree in information science and technology from Hokkaido University, Japan, in 2019. From 2019 to 2023, he was a postdoctoral researcher at Hokkaido University. Currently, he is a specially appointed assistant professor at the Education and Research Center for Mathematical and Data Science, Hokkaido University. His research interests include artificial intelligence, deep learning, and cognitive modeling. (w.noguchi@mdsc.hokudai.ac.jp)



Koki Osada received his B.S. degree in information science and technology from Hokkaido University, Japan, in 2021. He is currently pursuing his M.S. degree in information science and engineering at Hokkaido University, Japan, and is in his second year of study. His primary research interest is in the field of artificial intelligence (AI), focusing on deep learning techniques and their applications in various domains. (kosd@ist.hokudai.ac.jp)

Masahito Yamamoto received his Ph.D. degree from the Graduate School of Engineering, Hokkaido University, Japan, in 1996. From 1996 to 1997, he was a research fellow of the Japan Society for the Promotion of Science. He was an assistant professor from 1997 to 2000 and an associate professor from 2000 to 2012 of Hokkaido University. Currently, he is a professor at the autonomous systems engineering laboratory, Hokkaido University, Japan (2012–). He is also a concurrent faculty member of the Center for Human Nature, Artificial Intelligence, and Neuroscience, Hokkaido University (2020–). His research interests include artificial life and intelligence, swarm intelligence, combinatorial optimization, and board game artificial intelligence (AI) programming. (masahito@ist.hokudai.ac.jp)