Camera Vision Based Animal Beat Back System for Agriculture Using Machine Learning

Rathish J [1], Manikandan S [1], Priyadarshini S [2], Velmani Ramasamy S [3]

[1]Student, Department of Computer science and Engineering, Adithya institute of technology, Coimbatore, Anna University, Tamil Nadu, India.

[2]Assistant Professor, Department of Computer science and Engineering, Adithya institute of technology, Coimbatore, Anna University, Tamil Nadu, India.

[2]Professor, Department of Computer science and Engineering, Adithya institute of technology, Coimbatore, Anna University, Tamil Nadu, India.

Corresponding Author: rathish55543@gmail.com

Abstract: - Recently, automatic wild animal detection methods using deep learning for taken images by camera traps have been reported. Energy consumption is important for edge devices that include deep learning because such devices are required to use outside where commercial power is not supplied. In this paper, we propose energy reduction methods for a wild animal detection device. The proposed methods are sensitivity adjustment for the motion sensor, attachment of a hat, motion detection by a frame difference method, and separation of functions on the device. The sensitivity adjustment for the motion sensor reduces the number of taking images by the camera. The attachment of a hat reduces the number of sensing's by the motion sensor. The frame difference method reduces the number of inferences by deep learning. The separation of functions on the device reduces the power consumption in both operation time and idle time. In the experiments, we evaluate the effect of the proposed four methods by applying them to a wild animal detection device which we proposed previously. We compare the energy reduction ratio when each method is applied and all methods are combined. Compared to the device without the proposed methods, we can reduce the energy consumption by more than half when we combined all methods.

Key Words: - ***Animal detection, Energy reduction, Deep learning.***

I. INTRODUCTION

Recently, personal injuries and crop damages by wild animals such as bears and boars have become a significant problem in Japan. According to a report by the Ministry of the Environment, the number of personal injuries by Japanese black bears (*Ursus thibetanus japonicus*) in 2020 was 158 [1].

Manuscript revised May 15, 2023; accepted May 16, 2023. Date of publication May 18, 2023.

This paper available online at www.ijprse.com

ISSN (Online): 2582-7898; SJIF: 5.59

Similarly, the crop damages by Japanese wild boars (*Sus scrofa leucomystax*) in 2019 was about 4.6 billion JPY [2].

In Japan, the traditional detection of wild animals was based on sightings. However, it took a long time for the notification to people. In fact, we have registered an email service provided by Aizu-Wakamatsu city when wild animals are detected. The time for notification is dozens of minutes to hours. If someone is near wild animals and he/she is unaware of the notification, he/she may encounter an accident by the wild animals. To reduce such an accident, it is important to detect wild animals safely and notify the detection information immediately.

We have developed a wild animal detection system using deep learning (DL) in [3] to detect wild animals automatically and notify people of the detection immediately. The detection device takes a static image by a camera when a motion sensor senses some movement. Then, the detection device infers whether a target wild animal is in the image using DL (i.e., edge computing). When a target wild animal is detected, the detection device alerts the detection to people near the device by generating a sound and a light. Also, the detection device notifies the detection information to registered people using an email via a server. The detection device is powered by a solar panel and a battery so that it operates outside where commercial power is not supplied. Similar detection devices were proposed by other organizations [4]–[8].

Energy consumption is one of the important subjects in detection devices. This is because it affects the size of the battery and the solar panel. If a large solar panel and a large battery are required, the location to place the devices is restricted. Also, the cost of the devices is increased. The problems of location and cost may reduce the chances to use the devices. As a result, the accidents and the crop damages by wild animals will not be reduced. Therefore, reducing the energy consumption of the detection devices is one of the important subjects.

On the other hand, detection devices that use a motion sensor and a camera have useless operations which just waste energy consumption. The detection devices take images regardless of the appearance of wild animals whenever the motion sensor senses some movements. If we use DL in the detection devices, DL is performed for such images although no wild animals appear. To reduce the energy consumption of the detection devices, it is required to reduce useless sensing's, taking images, and inferences.

In this paper, we propose energy reduction methods for wild animal detection devices. The proposed methods consist of sensitivity adjustment for the motion sensor, attachment of a hat, motion detection by a frame difference method, and separation of functions on the devices. The first method reduces the energy consumption of the detection devices by reducing the number of taking images by the camera. The second method reduces the energy consumption of the detection devices by reducing the number of sensings by the motion sensor. The third method reduces energy consumption by reducing the number of inferences by DL. The last method reduces the energy consumption during operation time and idle time separating the functions to a parent node and multiple child nodes. In the experiments, we confirm the effect of each method and the combinations of all methods.

The main contribution of this paper is:

To reduce the energy consumption of wild animal detection devices which use a motion sensor and DL like [3]–[8].

The reduction is achieved by reducing useless operations for sensing, taking images, and inferences by the proposed methods. The reduction results in the reduction of the battery size and the solar panel size required for the devices. It increases the location to place the devices and reduces the device cost. As we can place more devices, we may reduce accidents and crop damages by wild animals.

The organization of this paper is as follows. In Section II, we describe related work. In Section III, we describe the overview of the wild animal detection device proposed in [3] which is the target of the proposed methods. In Section IV, we describe the proposed methods. In Section V, we describe the experimental results. Finally, in Section VI, we describe the conclusion and future work.

II. Related Work

Camera traps which consist of a motion sensor and a cameraare well used to detect wild animals. When some movementsare sensed by the motion sensor, it triggers the camera to take images. Analyzing taken images, we can specify speciesand behaviors of wild animals [9], [10]. Trail cameras are representative devices of camera traps. A wide variety of commercial trail cameras with different functions is avail- able [11]. In the trail cameras, we can change the batterylife by changing the sensitivity of motion sensors, the sizeof taken images, and the frequency of communications.

Recently, various studies have addressed the detection of wild animals using DL. Most of such studies aim to automatethe detection of wild animals because the manual analysis ofa huge number of images taken by camera traps is very time-consuming work. They can be classified into the development of DL models for detection or classification of wild animals [12]–[18] and the development of a device to detect wild animals [3]–[8], [19]. In this section, we address them and describe the difference from the proposed methods.

Nguyen *et al.* used DL to filter and identify wild animals in [12]. In filtering, DL was used to detect whether wild animals exist or not in taken images. In identification, DL was used to classify wild animals into three or six different species. A light AlexNet, VGG-16, and ResNnet-50 were used as Convolutional Neural Network (CNN) models. In the experiments, the accuracy of filtering and identification was evaluated for the Wildlife Spotter dataset.

Gomez *et al.* proposed a wild animal identification method for images taken by camera traps in [13]. In the experiments, four data sets were prepared from the Snapshot Serengeti dataset. Those data sets were an unbalanced num- ber of images, a balanced number of images, objects in the foreground of images, and segmented animal images with twenty-six classes. Used CNN models were AlexNet, VGGNet, GoogLeNet, and ResNet. In all CNN models, the use of the segmented animal images reached the best accuracy.

Noroussadeh *et al.* proposed not only animal identification methods but also counting and behavior describing methods using DL in [14]. CNN models for image classification were used for all of the methods. In the experiments, the accu- racy of identification (48 classes), counting (12 classes), and behavior describing (6 classes) was evaluated using AlexNet, NiN, VGG, GoogLeNet, and ResNet with the Snapshot Serengeti dataset. The authors showed that the accuracy was almost equivalent to the classification by experts.

Thangarasu *et al.* compared animal identification methods by machine learning (ML) methods in [15]. The target ML methods were DL, Support Vector Machine, and Random Forest. In the experiments, the authors showed that DL with Inception-v3 was the highest accuracy for twelve animal classification for the KTH dataset.

Takagi and Hirano proposed a classification method for agricultural work and wild animals using DL in [16]. For images taken by camera traps, DL was performed at a server located in the cloud. Images taken at daytime were classified as agricultural work while images taken at nighttime were classified as appearance or not of wild animals.

Yousif *et al.* proposed a computer vision tool that can classify a large number of pictures taken from camera traps into humans, animals, and backgrounds in [17]. To accelerate the classification with high accuracy, the authors coupled foreground object segmentation through background subtraction with deep learning classification. In the experiment, the authors evaluated the CPU time with the accuracy of the method.

Jamil *et al.* proposed a detection method for Himalayan Bear, Marco Polo Sheep, and Snow Leopard in [18]. They used Inception-v3 and k-Nearest Neighbour for classification. They also evaluated the proposed method comparing with other models such as ResNet-50.

Compared to the methods in [12]–[15], [17], [18], this paper focuses on a device to detect wild animals using DL. Note that the method in [16] used a frame difference method to classify agricultural work. Although the proposed methods also use a frame difference method, the aim of the frame difference method in the proposed methods is to reduce the number of inferences by DL to reduce the energy consumption of detection devices.

Elias *et al.* proposed an Internet-of-Things (IoT) system to monitor wild animals in the UCSB Sedgwick Reserve [4]. The IoT system consists of cloud, edge servers, and sensings. The edge servers perform DL for images taken by camera traps which are a kind of sensings. The cloud is used to generate trained DL models. The authors used a synthetic training set for training which consists of composite images of wild animals and empty (background) images taken at the UCSB Sedgwick Reserve, reducing transfers of images to prepare the training set. The authors showed that the high accuracy was reached even such a synthetic training set was used.

Monburinon *et al.* proposed a hierarchical edge computing-based image recognition system in [5]. The system consists of a cloud computing layer, edge computing layer, and physical interaction layer. Images taken by a camera in the physical interaction layer are sent to the edge computing layer. Then, DL is performed for those images. Training to develop a DL model is performed at the cloud computing layer. Raspberry Pi was used for the edge computing layer to reduce the system cost. As an application example, the authors evaluated the recognition accuracy for wild animals.

Curtin and Matthews proposed a wild animal detection device based on Raspberry Pi 3 Model B+ in [6]. Similar to [5], the authors used Raspberry Pi to realize the device at a low cost. In the experiments, the authors evaluated the accuracy to identify snow leopards. The accuracy was 97 % for images obtained by the Internet and 74 % for images taken by a camera.

Dihingia *et al.* proposed a wild animal detection device based on Raspberry Pi 3 in [7]. MobileNet SSD was used as the CNN model. However, compared to other literature, they did not describe the accuracy of DL.

Zualkernan *et al.* proposed an IoT system that uses Rasp- berry Pi to classify wild animals from images taken by camera traps in [8]. Images taken by camera traps are sent to an edge device which is based on Raspberry Pi. The edge device per- forms DL for the images. When wild animals are identified, the information is sent to the cloud so that users can check the information from mobile devices. In the experiments, the authors evaluated the accuracy when various CNN models are used.

Kamesaka and Hoshino proposed an IoT system to check whether a group of Japanese monkeys (*Macaca fuscata*) is entered into a cage in [19]. The system consists of sensors, cameras, and Raspberry Pi. When something is detected by a motion sensor, Raspberry Pi is launched to take images in the cage. When a group of Japanese monkeys is detected, the information is sent to users. The users check the Japanese monkeys through a browser and push a button to close the door of the cage.

In [5], [6], the authors evaluated the power consumption of the developed devices. In [19], the authors addressed reducing the power consumption of Raspberry Pi by launching only when sensors operate. Other literature did not address power consumption.

Compared to [4]–[8], [19], the main contribution of this paper is to reduce the energy consumption of wild animal detection devices. As an example, we use our developed detection device described in [3]. Note that our previous work [3] did not include the proposed methods described in this paper. Also, this paper includes small modifications of the detection device to reduce the execution time (see Section IV.A).

III. Overview Of Wild Animal Detection Device

In this section, we briefly explain the wild animal detection device used in this paper. We call the device as *detection device* in the rest of the paper. Please refer to [3] in detail.

3.1 *Structure And Processing Flow*

Figure 1 shows the structure of the detection device. The center of the detection device is the Raspberry Pi 3 Model A Passive Infrared Ray (PIR) motion sensor is used to trigger the device. A light sensor is used to judge daytime or nighttime. An infrared projector is used to take images at nighttime. Therefore, according to the value of the light sensor, a relay module cuts off the power supply to the infrared projector. A wireless module is used to communicate with a server. A light emitting device and a speaker are used for alerting people near wild animals while driving away the wild animals. The power supply of the detection device consists of a solar panel, a lead-acid battery, and a battery controller to use the detection device outside where a commercial power supply is unavailable.

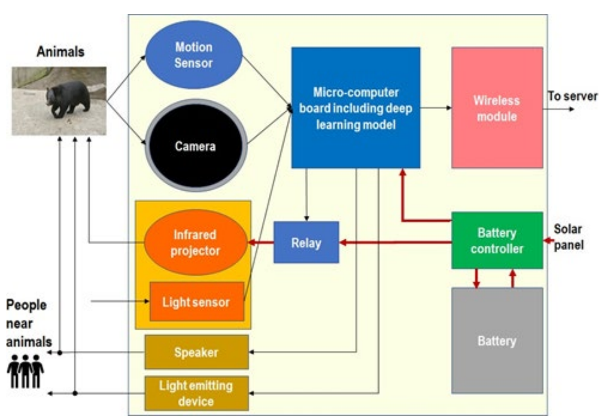

Fig.1. The structure of the detection device proposed in [3] (this figure is modified one of Fig. 3 in [3]).

Figure 2 shows the processing flow of the detection device. The detection device starts the processing when the motion sensor senses something. The light sensor measures the brightness in front of the detection device. The detection device turns on the infrared projector through the relay mod- ule when the value of the light sensor is less than the assigned *threshold* value. We use the light sensor to distinguish day- time and nighttime. Then, the detection device takes an image using the camera. For the taken image, the detection device infers the existence of target animals such as Japanese black bears in the taken image. A pre-trained Inception-v3 [20] using ImageNet [21] is used as the trained model. When a target animal is detected, the detection device alerts people near the detection device generating a sound and a light. Also, the detection device notifies the detection information to authorized people using email through a server.

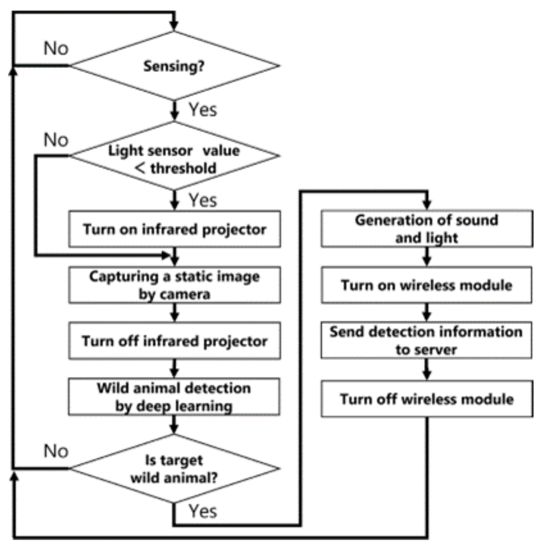

Fig.2. The processing flow of the detection dvice in [3] (this figure is also modified one of Fig. 2 in [3]).

3.2 About Power and Energy Consumption

Power reduction methods were applied to the detection device in [3]. First, the detection device does not start the process- ing until something is detected by the motion sensor like trail cameras. Second, the power is supplied to the infrared projector at nighttime only, by the use of the light sensor and the relay module. As a result, the average power consumption during the processing time and the idle time without the speaker and the light emitting device was 3.5 W and 2.5 W.

The operating power consumption of the detection device except for the speaker and the light emitting device was the largest in the order of the infrared projector, communication module, DL, and camera and sensor. As the infrared projector requires DC 12 V, the power consumption was the largest. On the other hand, when the speaker and the light emitting device are used, their power consumption will be the largest. This is because the alert time will be longer than the taking time by the camera while requiring DC 12 V for the speaker and the light emitting device as same as the infrared projector. Although the power consumption of each component is important, the actual energy consumption depends on the operation frequency. Compared to the sensor, camera, and DL, the operating frequency of the speaker and the light emit- ting device is very low. This is because the speaker and the light emitting device are operated only when target animals such as Japanese black bears are detected by DL after the motion sensor senses. In fact, there were many days in field tests when the speaker and the light emitting device never worked. On the other hand, the operating frequency of the sensor and the camera is very high. In particular, as the motion sensor is operated by light reflection and wind-induced fluctuations in trees, there are many useless operations. Also, DL is performed when the motion sensor senses something. Inference for images without animals is also meaningless.

It just wastes energy consumption.

From the above observations, it is important to reduce the number of sensings and inferences to reduce the energy consumption of the detection device. As the motion sensor triggers the detection device, the reduction of the number of sensings results in the reduction of the energy consumption directly. Reducing the power consumption of the communication module and the power consumption of the detection device in idle time are also important.

IV. PROPOSED METHOD

In this paper, we propose four energy optimization methods for the wild animal detection device reported in [3]. *Sensitivity adjustment for the motion sensor* reduces the number of taking images by the camera. *Attachment of a hat* reduces the number of sensings by the motion sensor. *A frame difference method* is used to reduce the number of inferences by DL. *Separation of functions* reduces the energy consumption of the detection device in both processing time and idle time.

The main contribution of this paper is to clarify the effect of the energy reduction for each method. The sensitivity adjustment for the motion sensor is general for trail cameras. Frame difference methods are well used in image processing to detect some motion from images. Although they are general, to the best of our knowledge, none of the literature does describe the effect on the energy consumption of wild animal detection devices. Also, we clarify the combination effect of the proposed methods. Note that the proposed methods are applicable to other detection devices if they use motion sensors, DL, or Wi-Fi.

4.1 Preparation Of a Baseline Device

To confirm the effect of the proposed methods, we prepare a baseline device. The baseline device is a modified version of the detection device in [3]. The modifications are as follows. Figures 3 and 4 represent the structure and the processing flow of the baseline device.

- Taking multiple images
- Use of a lightweight CNN model
-

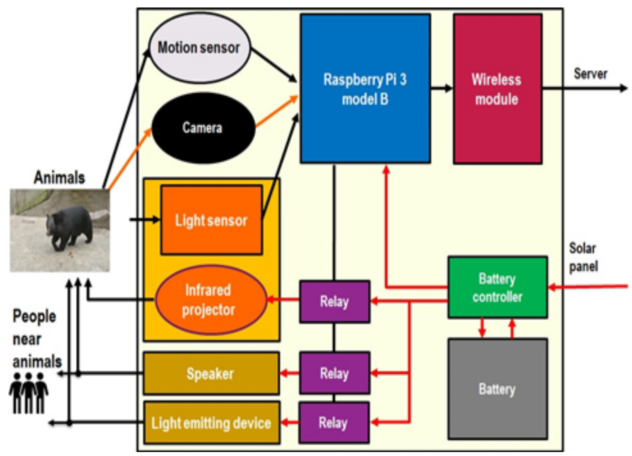

Fig.3. The structure of the baseline device.

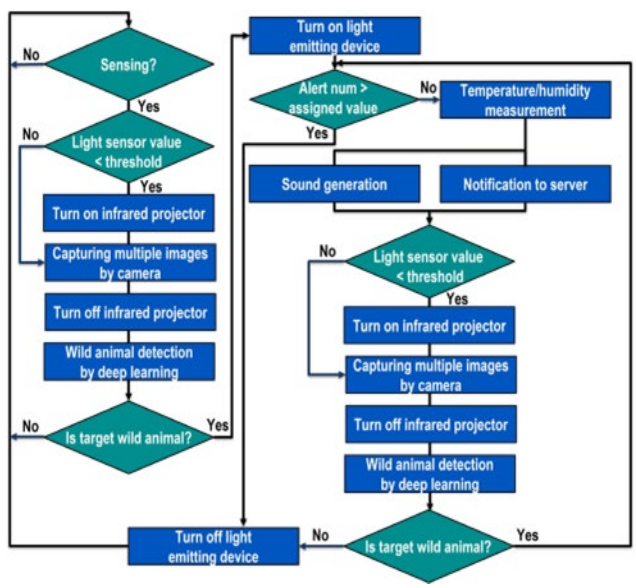

Fig.4. The processing flow of the baseline device.

- Loading of a trained model on Raspberry Pi in advance
- Use of relay modules to control power supply to the speaker and the light emitting device
- Use of a temperature and humidity sensor
- Execution of communication and sound and light generation by multi-threading
- Repeated processing from DL to alerting

The baseline device takes multiple images by the cam- era when sensed. The original device in [3] just takes an image when sensed. However, there may be no animals in the image depending on their movement (e.g., running). There- fore, by taking images multiple times assigned by users, we improve the possibility that wild animals are in taken images. Note that any comparison among images to detect a motion is not performed in the baseline device.

To reduce the inference time, a lightweight CNN model is used in the baseline device. The original device in [3] takes a few seconds for the inference using Inception-v3. As the power consumption of DL is higher than that of the motion sensor and the camera, the energy consumption becomes large when the inference time is long. To solve this problem, we use MobileNet-v2 [22] in the baseline device. For a pre- trained model of MobileNet-v2 with ImageNet, we generate a trained model using transfer learning with animal images collected by ourselves. Note that we discuss the dataset for training and the accuracy of the trained model in Section V.

We reduce the inference time by loading the trained model on Raspberry Pi in advance. In the original device, loading of the trained model is performed for each inference [3]. As the loading time dominates the inference time (about 20 s), we reduce the loading time so that the trained model is loaded when the power supply for Raspberry Pi is turned on.

The speaker and the light emitting device consume more power compared to other components. On the other hand, they are used only when target animals such as Japanese black bears are detected. We use relay modules to supply power for the speaker and the light emitting device when target animals are detected.
The measurement of temperature and humidity is to record which condition wild animals appear in. Reducing the wait- ing time until some of them are completed, communication, sound generation, and light generation are performed by multi-threading. Repeated processing from DL to alerting is to drive away wild animals. When a particular sound is used, wild animals may get used to the sound. To solve this problem, the baseline device generates a different sound ran- domly. Currently, fourteen sounds are installed in the baseline device.
On the other hand, the baseline device repeats from DL to alerting forever if wild animals are used to all sounds. To avoid such a case, repeated processing from DL to alerting is terminated by assigning the repeat number.
The baseline device takes 3 s from sensing to inference when three images are taken and inferred, because of the above modifications. The inference time per image is about 0.35 s. Compared to the original detection device which took about 26.5 s from sensing to inference, a large portion of the processing time is reduced in the baseline device. This contributes to the reduction of energy consumption because energy consumption depends on not only power consumption but also execution time.

4.2 *Sensitivity Adjustment for The Motion Sensor*

The detection device starts the processing when something is sensed by the motion sensor. As the used motion sensor is a PIR sensor, it senses a thermal reaction in front of the sensor. Therefore, it senses not only the target wild animals but also people, cars, light reflection, or grass or tree fluctuation by the wind. As sensing except for wild animals just wastes the energy consumption of the detection device, this method reduces the energy consumption by reducing the number of taking images by the camera while filtering useless sensings. The sensitivity adjustment for the motion sensor is based on the setting of *num* which represents the number of continuous sensings. In other words, the detection device takes images and starts inference when the motion sensor senses *num* times. Adjustment by the number of continuous sensings is reasonable because it can be used for various PIR sensors under various conditions. Some PIR sensors allow users to set the sensing range or the holding time while some of them do not. On the other hand, the motion sensor may sense just once when the detection device is placed on an animal trail, because wild animals may walk through the place. On the other hand, the motion sensor may sense multiple times when the detection device is placed in livestock sheds or vegetable fields because wild animals eat something there.

We need to decide the value of *num* from the PIR sensor's specification or the placed location. In general, the PIR sensor stores the sensing value during the holding time and has a waiting time for the next sensing. Therefore, we need to consider the holding time and the waiting time. Note that the energy consumption of the detection device will be decreased when a large value is assigned to *num*. This is because it restricts taking images by the camera. However, it may reduce the possibility that wild animals are in taken images.

4.3 Attachment of A Hat

The attachment of a hat reduces the number of sensings. It reduces the energy consumption of the detection device.

The number of sensings differs according to the shape of the hat. Also, as the targets of our detection device are Japanese black bears or Japanese wild boars, the detection device would like to sense near the ground. This is because those animals move while walking. Therefore, we prepare two shapes for the hat: arch type and knife type as shown in Fig. 5(a) and (b). The arch type can sense from the side or below. The knife type restricts sensing even the side or below. The knife type may reduce more energy consumption because it restricts the number of sensings compared to the arch type. The hat is made of a rigid polyvinyl chloride tube.

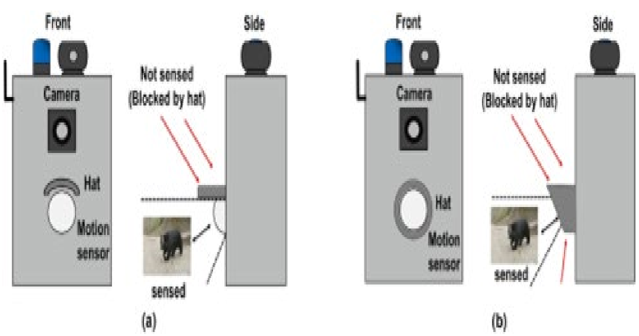

Fig.5. Attachement of a hat to the motion sensor: (a) the arch type and (b) the knife type.

4.4 Use Of a Frame Difference Method

In this paper, we use a frame difference method to detect the movement of wild animals. We restrict inference by the magnitude of movement. It reduces the energy consumption of the detection device by reducing the number of inferences. We adopt a general method for the frame difference which is implemented by OpenCV.

First, the detection device obtains the difference images from pairs of images as shown in the center of Fig. 6. Next, the detection device calculates the product of the difference images and performs the binarization to the product to obtain the region of the moving object (bottom of Fig.6).

The detection device performs inference by DL when the sum of the binarized values in the region is more than *threshold*. Otherwise, it skips inference and waits for the next sensing.

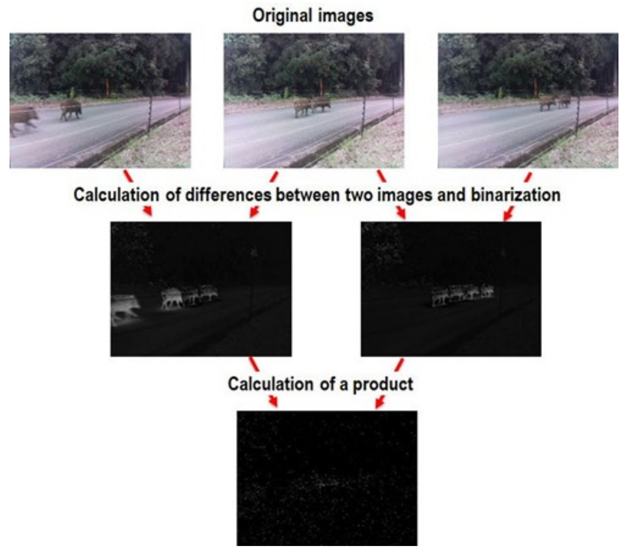

Fig.6. Calculation of frame differences to identify the movement of objects.

The value of *threshold* affects the detection of wild ani- mals. A smaller value allows the detection device to infer slight movement. As a result, the detection device may con- sume more power. On the other hand, a larger value restricts inference by the detection device. It may reduce the energy consumption of the detection device. However, it may lose the detection of wild animals when the size of them on images is small.

We separate the value of *threshold* in daytime and night- time. This is because the information such as color in night- time images is less compared to the information in daytime images. Therefore, a smaller value is assigned to *threshold* for nighttime.

Note that as the difference is almost none, DL is not performed when wild animals are standing in front of the detection device. In such a case, the detection device cannot detect the wild animals until they move.

4.5 Separation Of Functions

This method reduces the power consumption of the detection device by separating the functions. In the detection device, the processing is performed when something is sensed by the motion sensor. Therefore, the power consumption of idle time dominates the battery life when sensing is little. The power consumption of the idle time also depends on the communication module because the communication module is always on to notify and alert the detection information immediately.

This method realizes the separation of the functions introducing a sensor network. The sensor network consists of one parent node and multiple child nodes as shown in Fig. 7. As Wi-Fi is available for Raspberry Pi 3 Model B, the par- ent node and the child nodes are connected using Wi-Fi. Although the area of Wi-Fi is quite narrow compared to Low Power Wide Area (LPWA), there are two merits of using Wi-Fi. First, it does not require any additional cost. Second, we can transfer images in the sensor network. Therefore, we use Wi-Fi. To extend the Wi-Fi area slightly, we use external antennas.

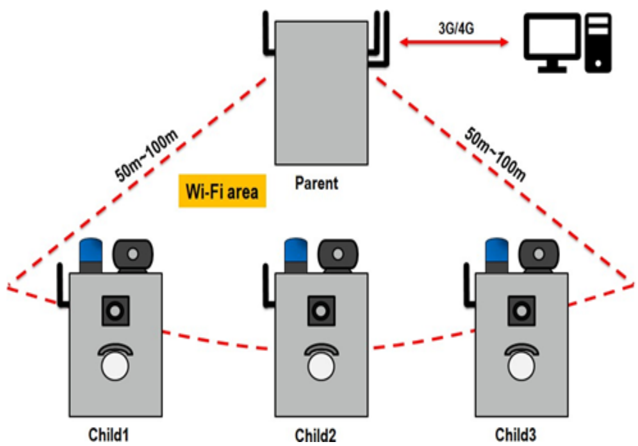

Fig.7. Separation of functions into a parent device and multiple child devices.

The parent node consists of a micro-computer board, a communication module, and a battery controller as shown in Fig. 8(a). Figure 8(b) represents the processing flow. The parent node just forwards the detection information that came from the child nodes to a server using a wireless module such as 3G or 4G. As the processing of the parent node is very limited, we use a low-end micro-computer board for the parent node to reduce the power consumption of idle time.

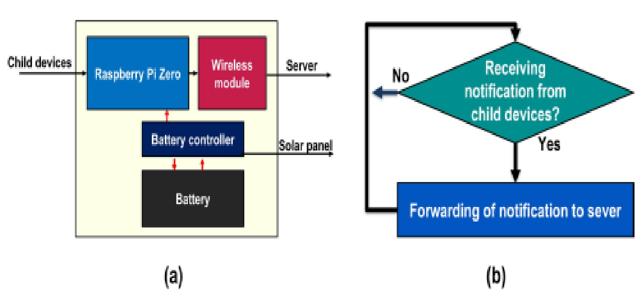

Fig.8. Parent device: (a) structure and (b) processing flow.

Figures 9 (a) and (b) represent the structure and the processing flow of the child nodes. The child nodes do not use a wireless module for a public line. The child nodes perform animal detection and alerting. The detection information is transferred to the parent node using Wi-Fi. Because of the absence of the wireless module, we can reduce the power consumption of the child nodes.

V. EXPERIMENTS

We evaluate the effect of the proposed four methods. In the sensitivity adjustment for the motion sensor, we evaluate the

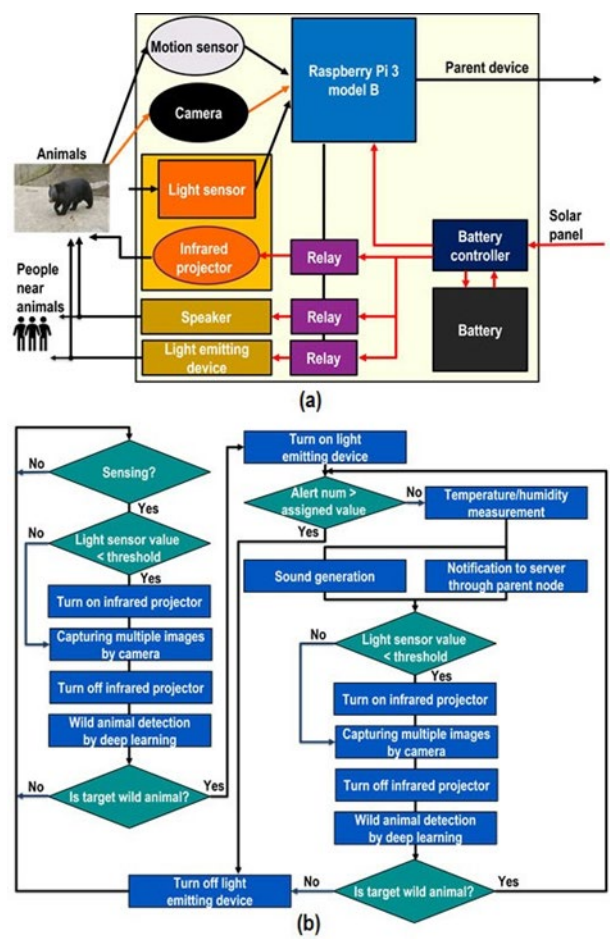

Fig.9. Child device: (a) structure and (b) processing flow.

number of taking images by the camera. In the attachment of a hat, we evaluate the number of sensings. In the frame difference method, we evaluate the number of inferences. In the separation of functions, we evaluate the reduction of energy consumption. Also, we calculate the energy consumption of the detection device when each method is applied and all methods are combined to confirm the usefulness of the proposed methods for the reduction of the energy consumption. For the combination, we assume four practical cases.

Before the evaluation, we prepared the baseline device as described in Section IV.A. Table 1 shows the used components. For the CNN model, we used a pre-trained model of MobileNet-v2 with ImageNet. We generated two trained models using transfer learning with images which were col- lected by ourselves from the Internet and our previous field tests. We also used images provided by citizens and local governments. The first model which consists of seven classes (*bear*, *boar*, *background*, *craw*, *monkey*, *raccoon*, and *other*) was for daytime. The second model which consists of six classes (*bearnight*, *boarnight*, *backgroundnight*, *abnormal*, *raccoonnight*, and *othernight*) was for nighttime. *bear* and *bearnight* include Japanese black bear images in daytime and nighttime, respectively. For training, we prepared 75 images and 25 images for each class as training data and validation data. In total, 525 images and 175 images were used for training and validation in the first model while 450 images and 150 images were used in the second model.

Table.1. Used components in the experiments.

Micro-computer board	Raspberry Pi 3 Model B
Motion sensor	Seeedstudio PIR Motion Sensor
Camera	Kuman Raspberry Pi Camera SC25
Infrared projector	Broadwatch SEC-IRLED-2B
Light sensor	TSL2591
Wireless module	Candy Pi Lite+ D
Relay	SODIAL 4ch 5V relay module
Light emitting device	Patlight SL08-M1JN
Speaker	DC12V 60W 110db speaker
Battery controller	ALLPOWERS 20A PWM Charge Controller
Battery	Long DC12V 36Ah
Solar Panel	Renogy DC12V 100W

For the model development, we used Google Colaboratory, TensorFlow, and Keras. We used MobileNet-v2 with Ima- geNet specified by Keras [23] for both models. As we used the transfer learning, the architecture of MobileNet-v2 except for the fully connected layer and the output layer was the same as the one provided in [23]. The image size was 224 x 224, the used optimizer was Adam, the learning rate was 0.0001, the batch size was 15, and the epoch size was 100. The width multiplier and the depth multiplier of MobileNet-v2 were both 1. The training was performed on a Linux machine with Intel Core-i7 and NVIDIA RTX 2080 Ti. During the training, training data were extended by the function of Keras. We used rescale, shear, zoom, and horizontal flip. We did not perform quantization during training.

After training, we tested the generated models to eval- uate the accuracy of the generated models. For the test, we prepared 25 images for each class. In total, 175 images and 150 images were used for testing the first model and the second model. Table 2 represents the performance of the generated models. In this research, as the target animals are Japanese black bears and Japanese wild boars, we evaluated'' accuracy'', ''precision'', ''recall'', and ''f-value'' for the target animals in the generated models. The performance for Japanese black bears was better in the first model while the performance for Japanese wild boars was better in the second model. Note that as the target of this paper is to reduce the energy consumption of the detection device, we are going to improve the accuracy of the generated models in our future work. Finally, we converted the trained models to TensorFlow Lite models because we implemented the TensorFlow Lite environment on Raspberry Pi.

To evaluate the energy consumption of the baseline device with the evaluation of the execution time and power consumption, we performed the following trial multiple times. Initially, we held an image of a Japanese black bear in front of the baseline device. When the baseline device sensed, it took three images by the camera. For each image, the baseline device performs DL. If the Japanese black bear was detected, the baseline device immediately alert the detection skipping DL for the rest of the images. After alerting, the baseline device again took three images and perform DL for each. However, we did not hold the image of the Japanese black bear in front of the baseline device. Therefore, the baseline device returned to the initial statement (i.e., waiting for the next sensing).

Table 3 shows the average execution time (T), average power consumption (P), and average energy consumption (E) of the baseline device in daytime and nighttime. We classify the process into''sensing'', "taking images'', ''inference'', and ''alerting and communication''. To obtain the execution time, we inserted *perf_counter* functions of *time* module to the program of the baseline device implemented in Python 3. Note that the time complexity of the program is $O(nm)$ where n represents the number of inferences and m represents the number of repeated processing's from DL to alerting to drive away wild animals in the worst case. To obtain the power consumption, we measured the average current and voltage of the baseline device by inserting IndoorCorgi ESP-PowerMonitor [24] to the power line. Note that the power consumption in the idle time was 2.5 W.

The energy consumption of "alerting and communication'' was the largest because the alerting time was the longest and the speaker and the light emitting device consume the largest power. In the nighttime, both execution time and power consumption were increased due to the use of the infrared projector.

As we could not prepare to check outside, we evaluated the sensitivity adjustment for the motion sensor using movies for Japanese black bears and Japanese wild boars taken by trail cameras. The number of movies was 77. Those movies were collected from trail cameras located at 9 places.

We made the relationship between *num* and movies as follows. The average holding time and the waiting time to the next sensing by the used PIR motion sensor were about 2.3 s and 1.2 s, respectively. As the sensing range and the holding time of the used PIR sensor could not be arranged from outside. We measured the average holding time and the waiting time. We assumed that the time from sensing to sensing was 3.5 s. According to the time that Japanese black bears and Japanese wild boars were taken in the movies, we related the number of continuous sensings (i.e., *num*). *num* is 1, 2, 3, or 4 when the taken time of those animals was 0.1 - 3.5 s, 3.6 - 7.0 s, 7.1 - 10.5 s, or more than 10.6 s, respectively. For example, if a Japanese black bear was taken 9 s in a movie, we regarded that the motion sensor senses 3 times continuously.

Table 4 shows the number of movies assuming the number of continuous sensings by the motion sensor. Also, it shows the ratio, R_{sense}, of the number of movies for ''num=1''. The number of movies implies the number of taking images by the camera.

According to the increase of *num*, the number of corresponding movies is reduced. It implies that the number of taking images by the camera is reduced when *num* is increased.

Table.2. Performance of the trained models.

model	animal	accuracy [%]	precision [%]	recall [%]	f-value [%]
First model (daytime)	Japanese black bear	97.7	86.2	100.0	92.6
	Japanese wild boar	94.9	80.8	84.0	82.4
Second model (nighttime)	Japanese black bear	94.7	90.5	76.0	82.6
	Japanese wild boar	96.7	91.7	88.0	89.8

Table.3. The Execution Time, Power Consumption, And Energy Consumption of The Baseline Device

daytime	T [s]	P [W]	E [J]
sensing (sense)	0.1	2.9	0.3
taking images (image)	0.7	3.6	2.5
inference (dl)	1.0	4.5	4.5
alerting and communication	61.4	9.5	583.3
nighttime	**T [s]**	**P [W]**	**E [J]**
sensing (sense)	0.1	3.4	0.3
taking images (image)	1.7	7.2	12.2
inference (dl)	1.1	3.9	4.3
alerting and communication	62.7	10.0	627.0

Table.4. The number of movies assuming the number of continuous sensings by the motion sensor.

num	movies	R_{sense} [%]
1	77	100.0
2	63	81.8
3	52	67.5
4	46	59.7

In cases when *num* is one, the motion sensor may sense not only wild animals but also other objects or light reflections. The latter just wastes the energy consumption of the detection device. On the other hand, if we set up *num* more or equal to two, we may filter useless sensings because it requires movement more than 3.6 s. In particular, wild animals may stay a long time at vegetable fields and livestock sheds for eating. Therefore, by increasing *num*, we focus on sensing wild animals. In other words, we reduce the energy consumption of the detection device by filtering useless sensings.

In the attachment of a hat, we evaluated the number of sensings during four days when the arch or knife type hat was attached to the motion sensor. As shown in Fig. 10, we prepared two detection devices. One was without a hat and the other was with a hat. The difference between motion sensors in the detection devices was 2 cm in height and 30 cm in width. Note that in the first four days we used the arch type hat while in the second four days we used the knife type hat. Table 5 represents the number of sensings in daytime and nighttime with and without a hat when the hat type was changed. It also shows the ratio of the hat, R_{hat} , for without hat.

In both daytime and nighttime, the number of sensings was reduced by the use of the hat. Although the comparison between arch type hat and knife type hat is unfair because the data were taken on different days, we guess that the knife type hat can reduce the number of sensings more than the arch type. This is because sensing from the side and below is restricted.

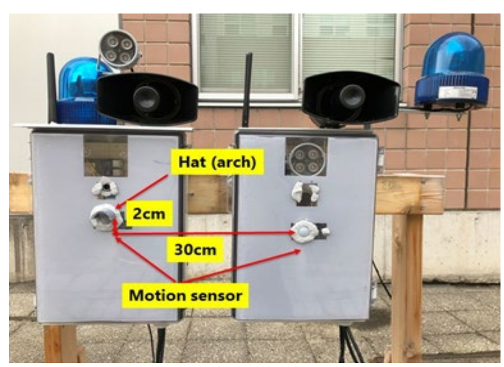

Fig.10. Evaluation on the effect of a hat for sensing's.

Table 5. The number of sensings with and without hats.

daytime	no hat	hat	R_{hat} [%]
arch	1,323	1,268	95.8
knife	1,239	810	65.4
nighttime	no hat	hat	R_{hat} [%]
arch	7	5	71.4
knife	3	0	0

For the evaluation of the frame difference method, we used two detection devices (device 1 and device 2) located at different places. We evaluated 206 images for device 1 and 362 images for device 2. All of them were taken in the daytime. Note that no Japanese black bear and Japanese wild boar were included in the images.

We evaluated the number of inferences when *threshold* was changed to 50, 100, 150, or 200. Table 6 shows the value of *threshold,* the number of images, the number of inferences, and the ratio of inference for images, R_{frame}.

Table.6. The number of inferences when the *threshold* value for frame differences is changed.

device 1	images	inferences	R_{frame} [%]
50	206	183	88.8
100	206	122	59.2
150	206	27	13.1
200	206	15	7.3
device 2	images	inferences	R_{frame} [%]
50	362	175	48.3
100	362	141	39.0
150	362	114	31.5
200	362	95	26.2

We could reduce the number of inferences when we increase *threshold*. It results in the reduction of the energy consumption in the detection device. On the other hand, the evaluation shows the importance to assign *threshold* properly considering the placed location. The reduction in device 1 was 40.8 % (100 - 59.2) when the value of *threshold* was 100 while the reduction in device 2 was 51.7 % although *threshold* was 50.

In the separation of functions, we used Raspberry Pi Zero for the parent node. This is because the parent node just for- wards the detection information to a server. Due to the use of Raspberry Pi Zero, we could reduce the power consumption of the parent node. Table 7 represents the execution time (T), power consumption (P), and energy consumption (E) per forwarding. The power consumption of the idle time was 1.4 W.

Table.7. The execution time, power consumption, and energy consumption of the parent device.

	T [s]	P [W]	E [J]
forwarding	11.9	3.5	41.7

The power consumption per forwarding was larger than that of ''sensing'' in the baseline device. The increase was caused by the processing of the wireless module. Also, the execution time became longer than that of the baseline device. The reason came that the computation power of Raspberry Pi Zero is smaller than that of Raspberry Pi 3. On the other hand, the power consumption of the idle time was decreased. There- fore, the use of Raspberry Pi Zero contributes to the reduction of the energy consumption if the idle time dominates the total time.

For the child nodes, we removed the wireless module. The other settings are the same as the ones of the baseline device. Table 8 represents the execution time (T), power consump- tion (P), and energy consumption (E) of "sensing'', ''taking images'', ''inference'', and ''alerting and communication'' in the child nodes for both daytime and nighttime. Also, Table 8 represents the reduction ratio of the energy consumption for the baseline device. The power consumption of the idle time was 1.4 W which was the same as the parent node. This comes from the absence of the wireless module.

By the separation of functions, the power and energy consumption of the child nodes were reduced compared to the baseline device. Although we just removed the wireless mod- ule from the child nodes, it contributed to a large reduction in the power consumption.

Next, we evaluate the energy consumption of the detection device for each proposed method. We calculate the energy consumption when the detection device operates at full oper- ation for one hour (= 3,600 s) of daytime using the formu- las (1) to (6). The formulas (1) to (6) represent the total energy consumption E_{total}, the energy consumption of the idle time E_{idle}, the idle time T_{idle}, the energy consumption when the detection device operates from sensing to DL, E_{s2dl}, the energy consumption when the detection device operates from sensing to frame difference, E_{s2i}, and the energy consumption when the detection device operates from sensing.

Table.8. The execution time, power consumption, and energy consumption of the child device.

daytime	T [s]	P [W]	E [J]	reduc. [%]
sensing (sense)	0.1	1.7	0.2	33.3
taking images (image)	1.0	2.0	2.0	20.0
inference (dl)	1.6	2.4	3.8	15.6
alerting and communication	61.4	8.3	509.6	12.6
nighttime	**T [s]**	**P [W]**	**E [J]**	**reduc. [%]**
sensing (sense)	0.1	1.7	0.2	33.3
taking images (image)	1.9	4.1	7.8	36.1
inference (dl)	1.5	2.4	3.6	16.3
alerting and communication	62.7	8.7	545.5	13.0

only, respectively. Note that T_{idle} is obtained by multiplying 1 - R_{hat} (the ratio that the motion sensor does not sense due to the hat) to 3,600. T_{sense}, P_{sense}, T_{image}, P_{image}, T_{dl}, and P_{dl} in the formulas (4) to (6) represents the execution time

(T) and the power consumption (P) of ''sensing'', ''taking images'', and ''inference'' in Tables 3 and 8. N_{s2dl}, N_{s2i}, and N_s represent the numbers of operations from sensing to DL, sensing to frame difference, and sensing only.

$$E_{total} = E_{s2dl} + E_{s2i} + E_s + E_{idle} \quad (1)$$

$$E_{idle} = T_{idle}P_{idle} \quad (2)$$

$$T_{idle} = 3,600(1 - R_{hat}) \quad (3)$$

$$E_{s2dl} = N_{s2dl}(T_{sense}P_{sense} + T_{image}P_{image} + T_{dl}P_{dl}) \quad (4)$$

$$E_{s2i} = N_{s2i}(T_{sense}P_{sense} + T_{image}P_{image}) \quad (5)$$

$$E_s = N_s(T_{sense}P_{sense}) \quad (6)$$

N_{s2dl}, N_{s2i}, and N_s are calculated by using the formulas

(7) to (13). They are calculated from the ratios, R_{sense} and R_{frame}, obtained from each proposed method and the weights from the execution time of the processing for the time that the detection device is operated within one hour (i.e., the time corresponds to 3,600 - T_{idle}).

$$N_{s2dl} = \frac{3,600R_{hat}R_{s2i}}{(1 - R_{sense})T_{sense} + (R_{sense} - R_{s2i})T_{s2i} + R_{s2i}T_{s2dl}} \quad (7)$$

$$N_{s2i} = N_{total}(R_{sense} - R_{s2i}) \quad (8)$$

$$N_s = N_{total}(1 - R_{sense}) \quad (9)$$

$$N_{total} = N_{s2dl}/R_{s2i} \quad (10)$$

$$R_{s2i} = R_{sense}R_{frame} \quad (11)$$

$$T_{s2i} = T_{sense} + T_{image} \quad (12)$$

$$T_{s2dl} = T_{sense} + T_{image} + T_{dl} \quad (13)$$

Table 9 represents the energy consumption of the detection device by each proposed method per hour. ''separation of function'' was the best to reduce energy consumption. 45.2 % of the energy consumption was reduced for the baseline device. Next was the ''sensitivity adjustment for the motion sensor''. It could reduce about 20 % to 26 % of the energy consumption for the baseline device. The knife type hat reduced 13.3 % of the energy consumption for the baseline

Table.9. Energy consumption of each method per hour when full operations are assumed.

method	N_{s2dl}	N_{s2t}	N_s	E_{s2dl} [J]	E_{s2i} [J]	E_s [J]	E_{idle} [J]	E_{total} [J]	reduc. [%]
baseline	2,000	0	0	14,620	0	0	0	14,620	-
adjust ($num = 1$)	2,000	0	0	14,620	0	0	0	14,620	0.0
adjust ($num = 2$)	590	0	131	10,301	0	1,368	0	11,669	20.2
adjust ($num = 3$)	295	0	142	8,145	0	2,924	0	11,069	24.3
adjust ($num = 4$)	185	0	125	6,986	0	3,843	0	10,828	25.9
frame (device 1, 50)	1,894	239	0	13,845	672	0	0	14,517	0.7
frame (device 1, 100)	1,531	1,055	0	11,192	2,965	0	0	14,157	3.2
frame (device 1, 150)	507	3,363	0	3,706	9,450	0	0	13,156	10.0
frame (device 1, 200)	301	3,822	0	2,200	10,740	0	0	12,940	11.5
frame (device 2, 50)	1,355	1,450	0	9,905	4,075	0	0	13,980	4.4
frame (device 2, 100)	1,180	1,846	0	8,626	5,187	0	0	13,813	5.5
frame (device 2, 150)	1,017	2,212	0	7,434	6,216	0	0	13,650	6.6
frame (device 2, 200)	888	2,501	0	6,491	7,028	0	0	13,519	7.5
hat (arch)	1,916	0	0	14,006	0	0	378	14,384	1.6
hat (knife)	1,308	0	0	9,561	0	0	3,114	12,675	13.3
separation (child)	1,333	0	0	8,011	0	0	0	8,011	45.2

Table.10. Energy consumption when all methods are combined.

method	N_{s2dl}	N_{s2i}	N_s	E_{s2dl} [J]	E_{s2i} [J]	E_s [J]	E_{idle} [J]	E_{total} [J]	reduc. [%]
baseline	2,000	0	0	14,620	0	0	0	14,620	-
separation (child)	1,333	0	0	8,011	0	0	0	8,011	45.2
case 1 (device 1)	829	105	0	4,982	228	0	1,744	6,954	52.4
case 2 (device 1)	220	152	83	2,631	1,234	508	1,744	6,117	58.2
case 3 (device 1)	26	172	96	466	2,420	1,159	1,744	5,789	60.4
case 4 (device 1)	9	115	83	215	2,302	1,496	1,744	5,757	60.6
case 1 (device 2)	607	650	0	3,648	1,411	0	1,744	6,803	53.5
case 2 (device 2)	152	238	87	1,818	1,933	532	1,744	6,027	58.8
case 3 (device 2)	62	135	95	1,110	1,899	1,147	1,744	5,900	59.6
case 4 (device 2)	32	90	83	764	1,802	1,496	1,744	5,806	60.3
separation (child, idle)	0	0	0	0	0	0	5,040	5,040	65.5

device. The last was "frame difference'' It could reduce 0.7 % to 11.5 % for the baseline device.

Finally, we calculate the energy reduction of the child nodes by the combination of the proposed methods. We assume device 1 and device 2 for the estimation. In the combination, we consider four practical cases (case 1, case 2, case 3, and case 4). In case 1 where *num* and *threshold* are 1 and 50, we assume that the movement of wild animals is fast or the size of the wild animals in taken images is small. In case 4 where *num* and *threshold* are 4 and 200, we assume that the movement of them is slow or the size of the wild animals in taken images is large. Cases 2 and 3 are assumed in between cases 1 and 4 where *num* and *threshold* are 2 and 100 in case 2 and 3 and 150 in case 3. The targets of the estimation are sensing and inference in the daytime. As the number of alerting and communication is very few compared to sensing, we exclude alerting and communication. Similarly, as the number of sensings in the nighttime is very few compared to the daytime, we also exclude the estimation in the nighttime. We also assume that the hat type is the knife type.

Table 10 represents the energy consumption for cases 1 to 4 per hour. Similar to Table 9, we obtained these values using the formulas (1) to (13). By combining all methods, we could reduce more energy consumption for the child node (separation (child)). It was 52.4 % to 60.6 % for the baseline

device. We also know that the energy consumption was reduced in proportion to the value of "num'' and "threshold''. As the energy consumption of the detection device when all methods are combined closes to 5,040 [J] that is the energy consumption when no operation during one hour (separation (child, idle)), we could confirm the effectiveness to combine all methods.

We can expect that the size of the solar panel and battery can be reduced by the proposed methods. The size of the solar panel and battery will be decided by the values of *num* and *threshold,* the alerting time by the speaker and the light emitting device, and the power generation efficiency of the solar panel. For *num* and *threshold,* we need to decide considering the placed location of the detection device and the situation of wild animals photographed. For example, the settings of case 1 or case 2 will be useful when the detection device is placed to animal trails while the settings of case 3 or case 4 will be useful when it is placed to livestock sheds or vegetable fields. We are going to clarify the relationship between the value of *num* and *threshold* and the size of the solar panel and battery in our future work.

Although we focus on the reduction of useless sensing's and inferences, the energy consumption of "alerting and communication'' may not be ignored according to the placed location. Wild animals may appear frequently if they obsess with crops or get used to sound and light to drive away. In those situations, the detection device just wastes energy consumption, because wild animals may not pay attention to the detection device. It may require the detection device to strengthen the function of driving away wild animals. Also, placing the detection device or the parent node at a location where the radio wave of the public line is weak may waste energy consumption. The wireless module consumes more power to connect to the public line. We need to pay attention for the location to place the detection device.

VI. Conclusion

In this paper, we proposed energy reduction methods for a wild animal detection device. The proposed method consists of the sensitivity adjustment for the motion sensor, the attachment of a hat, the frame difference method, and the separation of functions. In the experiments, we confirmed the reduction of the number of taking images by the sensitivity adjustment for the motion sensor, the amount of sensing's by the attachment of a hat, the reduction of the number of inferences by the frame difference method, and the reduction of the energy consumption by the separation of functions. Also, we evaluated the energy reduction when each proposed method is applied and all methods are combined. Combining the proposed methods, we could reduce more than half of the energy consumption from sensing to inference of the detection device.

In our future work, we are going to clarify the required battery and solar panel size through a field test. Also, we are going to compress the CNN model size to reduce the energy consumption during inference. Compression of the CNN model may lose the accuracy. Therefore, we are going to check the effect of compression using the ablation study. Evaluation of the accuracy for the generated models will be also performed during the field test.

Acknowledgment:

The authors would like to thank Mr. Hajime Takeda, who supported the settings of the detection devices, and also would like to thank the Aizu Development Bureau and people in the Yagisawa area of Aizu-Misato town, who supported the field tests.

References

[1]. Ministry of Environment, Japan. Number of Personal Injury Caused by Bears. (In Japanese). Accessed: Feb. 22, 2022.

[2]. Ministry of Agriculture, Forestry, and Fisheries, Japan. Damage to Crops Caused by Wild Animals. (In Japanese). Accessed: Feb. 22, 2022.

[3]. H. Saito, T. Otake, H. Kato, M. Tokutake, S. Semba, Y. Tomioka, and Y. Kohira, ''Battery-powered wild animal detection nodes with deep learning,'' IEICE Trans. Commun., vol. E103.B, no. 12, pp. 1394–1402, Dec. 2020.

[4]. A. R. Elias, N. Golubovic, C. Krintz, and R. Wolski, ''Where's the bear: Automating wildlife image processing using IoT and edge cloud systems,'' in Proc. 2nd Int. Conf. Internet-Things Design Implement., Apr. 2017, pp. 247–258.

[5]. N. Monburinon, S. M. S. Zabir, N. Vechprasit, S. Utsumi, and N. Shiratori, ''A novel hierarchical edge computing solution based on deep learning for distributed image recognition in IoT systems,'' in Proc. 4th Int. Conf. Inf. Technol. (InCIT), Oct. 2019, pp. 294–299.

[6]. B. H. Curtin and S. J. Matthews, ''Deep learning for inexpensive image classification of wildlife on the raspberry Pi,'' in Proc. IEEE 10th Annu. Ubiquitous Comput., Electron. Mobile Commun. Conf. (UEMCON), Oct. 2019, pp. 82–87.

[7]. U. Dihingia, P. Amar, M. M. Shyam, V. Thomas, and S. Chidambaram, ‘‘Animal identification using deep learning on raspberry Pi,’’ Int. J. Res. Eng., Sci. Manage., vol. 3, no. 3, pp. 454–456, 2020.

[8]. I. A. Zualkernan, S. Dhou, J. Judas, A. R. Sajun, B. R. Gomez, L. A. Hussain, and D. Sakhnini, ‘‘Towards an IoT-based deep learning architecture for camera trap image classification,’’ in Proc. IEEE Global Conf. Artif. Intell. Internet Things (GCAIoT), Dec. 2020, pp. 1–6.

[9]. X. Yu, J. Wang, R. Kays, P. A. Jansen, T. Wang, and T. Huang, ‘‘Automated identification of animal species in camera trap images,’’ EURASIP J. Image Video Process., vol. 2013, no. 1, p. 52, Dec. 2013.

[10]. B. G. Weinstein, ‘‘A computer vision for animal ecology,’’ J. Animal Ecol., vol. 87, no. 3, pp. 533–545, May 2018.

[11]. Bushnell. CelluCORE 20 Low Glow Cellular Trail Camera. Accessed: Feb. 22, 2022.

[12]. H. Nguyen, S. J. Maclagan, T. D. Nguyen, T. Nguyen, P. Flemons, K. Andrews, E. G. Ritchie, and D. Phung, ‘‘Animal recognition and identi- fication with deep convolutional neural networks for automated wildlife monitoring,’’ in Proc. IEEE Int. Conf. Data Sci. Adv. Anal. (DSAA), Oct. 2017, pp. 40–49.

[13]. A. G. Villa, A. Salazar, and F. Vargas, ‘‘Towards automatic wild ani- mal monitoring: Identification of animal species in camera-trap images using very deep convolutional neural networks,’’ Ecol. Informat., vol. 41, pp. 24–32, Sep. 2017.

[14]. M. S. Norouzzadeh, A. Nguyen, M. Kosmala, A. Swanson, M. S. Palmer, C. Packer, and J. Clune, ‘‘Automatically identifying, counting, and describing wild animals in camera-trap images with deep learn- ing,’’ Proc. Nat. Acad. Sci. USA, vol. 115, no. 25, pp. E5716–E5725, Jun. 2018.

[15]. R. Thangarasu, V. K. Kaliappan, R. Surendran, K. Sellamuthu, and J. Palanisamy, ‘‘Recognition of animal species on camera trap images using machine learning and deep learning models,’’ Int. J. Sci. Technol. Res., vol. 8, no. 10, pp. 2613–2622, 2019.

[16]. M. Takagi and R. Hirano, ‘‘Proposal and evaluation of a method for automatically classifying images of agricultural work and animals acquired with motion sensor cameras,’’ in Proc. IEEE 2nd Int. Conf. Inf. Comput. Technol. (ICICT), Mar. 2019, pp. 102–108.

[17]. H. Yousif, J. Yuan, R. Kays, and Z. He, ‘‘Animal scanner: Software for classifying humans, animals, and empty frames in camera trap images,’’ Ecol. Evol., vol. 9, no. 4, pp. 1–12, 2019.

[18]. S. Jamil, Fawad, M. S. Abbas, F. Habib, M. Umair, and M. J. Khan, ‘‘Deep learning and computer vision-based a novel framework for Himalayan bear, marco polo sheep and snow leopard detection,’’ in Proc. Int. Conf. Inf. Sci. Commun. Technol. (ICISCT), Feb. 2020, pp. 1–6.

[19]. R. Kamesaka and Y. Hoshino, ‘‘Development of a prevention system for beast damage of agricultural products using deep learning,’’ in Proc. Joint 10th Int. Conf. Soft Comput. Intell. Syst. (SCIS), 19th Int. Symp. Adv. Intell. Syst. (ISIS), Dec. 2018, pp. 747–752.

[20]. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, ‘‘Rethinking the inception architecture for computer vision,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 2818–2826.

[21]. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, ‘‘ImageNet: A large-scale hierarchical image database,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2009, pp. 248–255.

[22]. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, ‘‘MobileNetV2: Inverted residuals and linear bottlenecks,’’ in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 4510–4520.

[23]. Keras Team. Keras. Accessed: Feb. 22, 2022.

[24]. Indoor Corgi Elec. E32-SolarCharger