PROCEEDINGS OF SCIENCE

Deep learning approaches for prevention of Japanese local monkey trespassing in a sweet potato field

Apirak Sang-ngenchai[a,b,*] **Hayato Ogawa**[a,b,]**and Minoru Nakazawa**[a]

[a] *Kanazawa Institute of Technology,*
7-1 Ohgigaoka, Nonoichi, Ishikawa, Japan

[b] *International College of Technology, Kanazawa,*
2-270 Hisayasu, Kanazawa, Ishikawa, Japan
E-mail: sapirak@neptune.kanazawa-it.ac.jp
E-mail: hogawa@neptune.kanazawa-it.ac.jp
E-mail: nakazawa@infor.kanazawa-it.ac.jp

In rural areas of Hakusan, Ishikawa Prefecture, Japan, the local monkey population has been causing damage to sweet potato farms and surrounding lands. To address this issue, a prototype system has been developed to assist farmers in protecting their fields during the months of September through November in both 2022 and 2023. The system is based on deep-learning models utilizing the you-only-look-once (YOLO) algorithm to classify and localize images of the local monkeys simultaneously. Real-time detection of live monkeys is possible using a Streaming Protocol Camera (RTSP), with the system automatically notifying farmer group members via the Line application when a monkey is detected. The system was trained using data collected from trap cameras placed around the sweet potato fields and successfully operated on a Windows PC with 32 GB RAM, a 64-bit Operating System, and an Intel(R) Core(TM) i9-9900K processor with an Nvidia GeForce RTX 3080 10 GB graphics processing unit (GPU). Performance metrics based on k-fold cross-validation showed precision, recall, and AP@0.5 values of 0.7310, 0.8462, and 0.7421, respectively, indicating high accuracy and classification performance using a real-time streaming camera. By alerting farmers in advance, the system can prevent damage caused by monkeys in sweet potato fields.

International Symposium on Grids and Clouds (ISGC) 2024,
24-29 March 2024
Academia Sinica Taipei, Taiwan

[*] Speaker

1. Introduction

According to a survey conducted by the Ministry of Agriculture, Forestry, and Fisheries (MAFF), the agricultural sector in Japan experienced a loss of approximately 15.8 billion yen in 2020 due to wild animal damage, with monkeys accounting for about 900 million yen of the total damage [1]. Protecting crops from such damage is critical to the agricultural community [2]. In recent years, there has been an increased interest in state-of-the-art object detection technology, particularly in agriculture, with AI and big data used to revolutionize modern agriculture. Deep Learning methods such as YOLO [3] have proven to be reliable and accurate in detecting objects, making it a significant method in computer vision [4]. This paper proposes a system to identify wildlife and assist local farmers in protecting their sweet potato fields from monkeys.

This initiative was launched in response to conflicts between humans and wildlife. To better understand the local farming community's needs and preferences, we interviewed several farmers around Hakusan villages. Our research showed that farmers needed to safeguard their crops and receive alerts when working in other locations. Since farmers typically own multiple small plots of land scattered across the community, it is only possible to be present in some locations simultaneously. This provided an opportunity for monkeys to enter the fields and cause damage. During our interviews, farmers shared that they had attempted to use airsoft guns and fireworks to deter the monkeys, but these approaches were only practical for a short time. While these methods did help to mitigate the damage caused by the monkeys, the farmers were seeking a more permanent solution that would alert them of potential intrusions and allow them to deploy a temporary scare tactic. This would enable them to quickly respond to threats, scare away intruders, and assess the damage.

The proposed system utilizes deep learning algorithms to monitor monkey activity in agricultural settings. The project explores the complexities of using the YOLO version 4 tiny algorithm to classify and localize images to identify local monkeys simultaneously. The prototype was designed to detect live monkeys in images from a real-time streaming Protocol (RTSP) camera and automatically notify farmer group members via the Line application. The system was successfully installed and tested at a sweet potato field in Hakusan, Ishikawa, using four cameras and a solar power system to ensure 24-hour operation from September through November of 2022 and 2023. Data from trail cameras installed around the sweet potato fields were collected to train the model, and the deep learning software called CiRA CORE [5] was used in the process.

This paper aims to provide a seamless reading experience and help the reader navigate the research more efficiently. The background of this research is explained in detail in Section 2, which also elaborates on the main integrated development environment (IDE) used for controlling and monitoring purposes. Section 3 discusses the implementation and engineering of computer vision techniques, including the tools used for model training and evaluation. Section 4 provides detailed methodologies for creating the prototype and information about the hardware upgrade and monitoring system. Experimental results are presented in Section 5, including testing both on-field and off-field. Finally, Section 6 concludes this paper by summarizing the work done and outlining future milestones.

2. Background

2.1 You Only Look Once (YOLO)

Computer vision has seen the rise of various convolution neural networks (CNNs), each tailored to address specific challenges. YOLO (You Only Look Once) [3] is the most widely used and influential among these networks. As its name implies, YOLO approaches frame object detection as a regression problem, mapping objects to spatially separated bounding boxes and corresponding class probabilities. This algorithm is structured to work in three parts. First, it divides the input images into an S x S grid, where S is a pre-defined number.

In each grid cell, the algorithm checks whether the center of an object falls within that cell. If the object falls into the grid cell, that cell is responsible for detecting it. Second, each grid cell predicts bounding boxes and confidence scores for those boxes. The bounding boxes are responsible for locating the object in the image, while the confidence scores represent the probability that the bounding box contains an object. Third, the algorithm multiplies the class probabilities and confidence box, giving each class's confidence score. This final score determines the class of the object detected in the image. With its ability to detect objects in real-time with high accuracy, YOLO has become a popular choice for applications such as self-driving cars, surveillance systems, and object recognition in images and videos.

2.2 CiRA Core Platform

The CiRA Core platform is a cutting-edge visual programming platform that harnesses the power of the Robot Operating System Framework (ROS) created by CiRA Automation and Technology Co., Ltd. This platform is a spin-off of the esteemed Center of Industry Robot and Automatic (CiRA) at King Mongkut's Institute of Technology Ladkrabang (KMITL). ROS [6] is a dynamic open-source framework that enables robots to complete complex tasks, offering many tools and libraries for designing sophisticated robotic systems. CiRA Core platform stands out with its unique integration with the Darknet framework, an open-source neural network framework written in C and CUDA. It is a parallel computing platform and programming model developed by NVIDIA. This integration provides the platform with powerful computer vision tools for image and object recognition and deep learning capabilities for creating and deploying customized neural networks. CiRA Core platform is user-friendly and supports both CPU and GPU computation, making it a perfect choice for those interested in block-based coding and building computer vision skills, as well as advanced users who require working with complex computer vision and external connectivity, such as the Internet of Things (IoT) and Serial Communication.

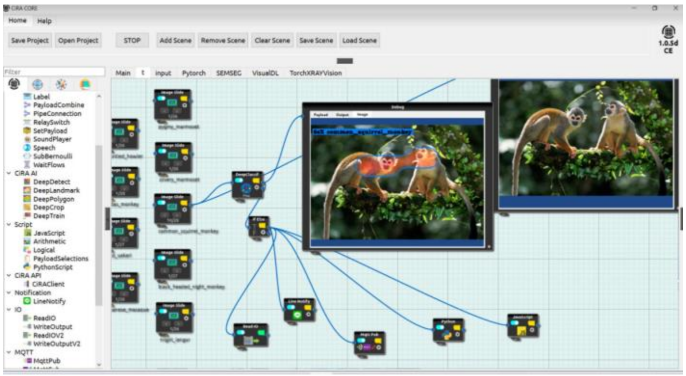


Figure 1: CiRA Core platform.

3. Methodologies

3.1 Evaluation of the Model

Our latest object detection model (2022 edition) is powered by Nvidia Jetson Nano 4GB and utilizes SiPEED's weight model mobilenet for its neural network architecture. We trained this model using the YOLO source code and saved the weight model in Keras format. For training, we utilized the 0.75_mobilenet-224 model, which multiplies the width of the network by 0.75 alpha with an input image size of 224 x 224 pixels. For more information on the architecture, please refer to Table 1.

Layer (type)	Output Shape	Parameter
Input layer	(None, 224, 224, 3)	0
mobilenet_0.75_224 (Model)	(None, 7, 7, 768)	1,832,976
detection_layer_30 (Conv2D)	(None, 7, 7, 30)	23,070
Reshape	(None, 7, 7, 5, 6)	0

Tabel 1: Mobilenet Body Architecture.

For this project, we utilized a pre-trained version of YOLO 4 tiny for object detection. The model runs on the CiRA Core platform's AI Station. To learn more about the model's configuration, please see section 3.3 of the Model Training documentation.

3.2 Image Data Collection and Annotation

We have collected a dataset of monkey videos that can be seen in action from various angles in sweet potato fields in the autumn of 2022 and 2023. We had to label each image individually to generate the annotation file and locate the monkeys in the images. Using a motion-activated trail camera, we captured 2,580 images of monkeys from a video. We utilized a labeling tool from the CiRA Core platform to label all the monkey images and separated them into two categories: Train Images and Validation Images.

Our training model's data annotation file format is XML, which follows the Pascal VOC format. This information is the bounding box description for each monkey present in the images. Each annotation file has the name of the class, the size of the image, and the location of the monkey described in the x-y axis from top left to bottom right, including the parameters as follows: folder, filename, object, center, label_name, label_index, and bounding box. The description of the bounding box is in the following format:

<xmin-top left, ymin-top left, xmax-bottom right, ymax-bottom right>

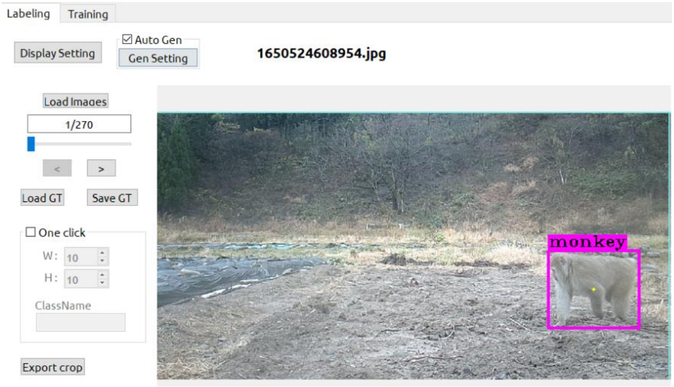


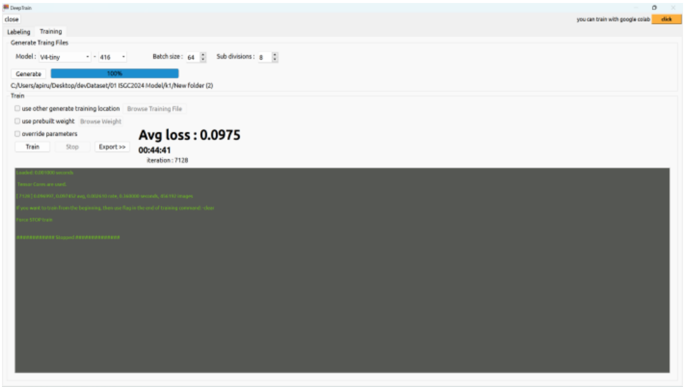
Figure 2: Labeling tool within the CiRA Core platform.

3.3 Model Training

We have chosen to use YOLO version 4 tiny for our custom object detection model. This version offers high accuracy while also being computationally efficient. We have selected and configured several parameters to train our model using this framework

We aim to train a highly accurate and effective object detection model by fine-tuning these parameters. The following configuration parameters are used to specify the input image size before training a model:

- Width, Height, and Channels: The input image is first resized to Width × Height and then processed by the model. Here, we have set the width and height to 416 pixels and the number of channels to 3. Channels = 3 indicates that we would be processing 3-channel RGB input images.
- Batch size: During training, a small subset of images is used to update the weights in one iteration. We have set its value to 64, which means 64 images are used in one iteration to update the parameters of the neural network.
- Subdivision: We have set its value to 8, which means the GPU will process the batch size/subdivision number of images at any time. However, the full iteration will be complete only after all 64 (as set above) images are processed.
- Learning Rate: The parameter learning rate controls how aggressively we should learn based on the current batch of data. We have set the learning rate to 0.00261.
- Momentum and decay: Momentum penalizes significant weight changes between iterations. To avoid overfitting, the parameter decay controls the penalty term applied to large values for weights. We have set the momentum to 0.9 and decay to 0.0005.
- Number of iterations: The number of iterations specifies how long the training process should last. We have trained YOLOv4 for 1,000 iterations on the Nvidia GeForce RTX 3080 10 GB graphics processing unit (GPU).



Generate Traing Files

Model : V4-tiny - 416 Batch size : 64 Sub divisions : 8

100%

C:/Users/apiru/Desktop/devDataset/01 ISGC2024 Model/k1/New folder (2)

Train

□ use other generate training location Browse Training File

□ use prebuilt weight Browse Weight

□ override parameters

Avg loss : 0.0975

00:44:41

iteration : 7128

```
Loaded: 0.001000 seconds

 Tensor Cores are used.

[ 7128 ] 0.096997, 0.097452 avg, 0.002610 rate, 0.360000 seconds, 456192 images

If you want to train from the beginning, then use flag in the end of training command: -clear

Force STOP train


############# Stopped ###############
```

Figure 3: Training Model in CiRA Core platform.

3.4 Model Evaluation

We assessed our custom object detection model using the K-fold cross-validation technique. To ensure the accuracy and robustness of the model, we split our dataset of 2,580 images into five groups. For each k value ranging from 1 to 5, we utilized four groups as the training dataset and one group as the test dataset. We trained the model five times, each time using a different group as the test dataset and the remaining four groups as the training dataset. Following the training, we calculated the precision, recall, and average precision with a 0.5 threshold (IoU = 0.5) for each model and then averaged the outcomes for all five models.

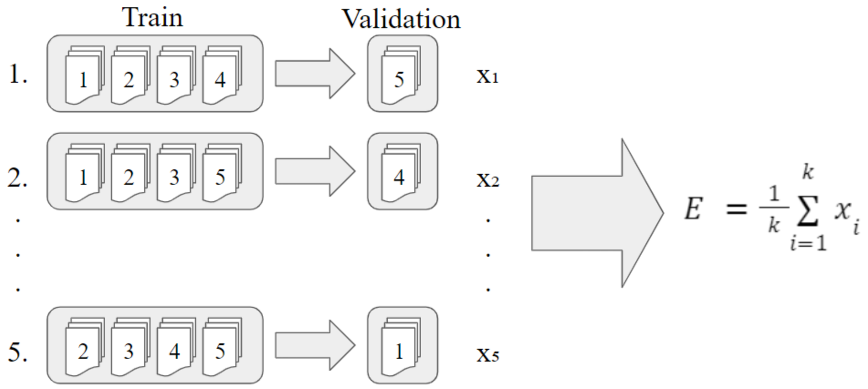


Figure 4: K-Fold Cross-Validation Diagram.

4. Implementation

4.1 Hardware Upgrade and Power Consumption Issue

For our previous hardware model, we utilized the Jetbot powered by the Nvidia Jetson Nano with 4GB memory. This open-source computing system runs on a Nvidia Cuda-X AI computer, offering 472 GLOPS computing performance for modern AI workloads. However, we encountered an issue with the expenses incurred by the Jetson Nano, which runs continuously 24/7, and the system experienced a month-long power outage due to insufficient sunlight to recharge the battery. To address these concerns, we utilized a low-power camera with a built-in solar cell battery, protected by a specially designed box that prevents water, sunlight, and heavy wind exposure. The box's airflow mechanism was also designed to prevent electronics from overheating. By relocating the AI processor outside the box, we reduced power consumption, allowing the solar panel to charge a battery pack that can power four cameras and a ventilation system. To ensure our 4 RTSP cameras remained functional, we used the JVC Kenwood Portable Power Supply, equipped with a rechargeable lithium-ion battery with a 104,400 mAh/375 Wh capacity. This powerful battery allowed our cameras to operate without interruption and ensured they always remained functional. The battery was recharged during the day using a solar panel close to the outdoor prototype, providing a continuous and reliable power supply. Additionally, the portable battery featured a 5V DC USB port, making it incredibly easy to connect with the camera and providing added convenience and ease of use.

4.2 Outdoor Camera House Shelter

The shelter has highly durable features to ensure optimal performance in extreme weather conditions. It features waterproof, heat-resistant, and heavy wind protection attributes that enable it to operate a system continuously for three months, specifically from September to November. The hardware structure is built using MDF wood, which provides a secure and robust shelter for the camera and battery, ensuring their safety and longevity during operation.

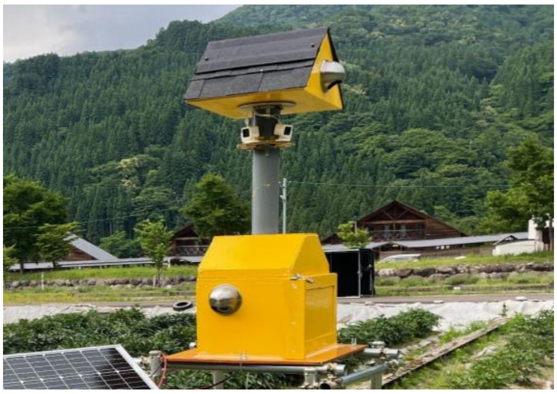


Figure 5: Outdoor Camera House Shelter.

4.3 Monitoring System

The Wyze version 3 camera is utilized in this project with Real-Time Streaming Protocol (RTSP) mode enabled, enabling the provision of an Internet Protocol (IP) address that seamlessly connects with the CiRA Core platform. The camera is connected to an AI station via a private router, allowing real-time video stream analysis to enhance object detection and security. As a result, farmers receive accurate real-time notifications via the Line application, identifying monkeys in images and enabling proactive measures to protect crops while minimizing damage. This revolutionary technology has significantly improved farming productivity by transforming the traditional approach to monkey control.

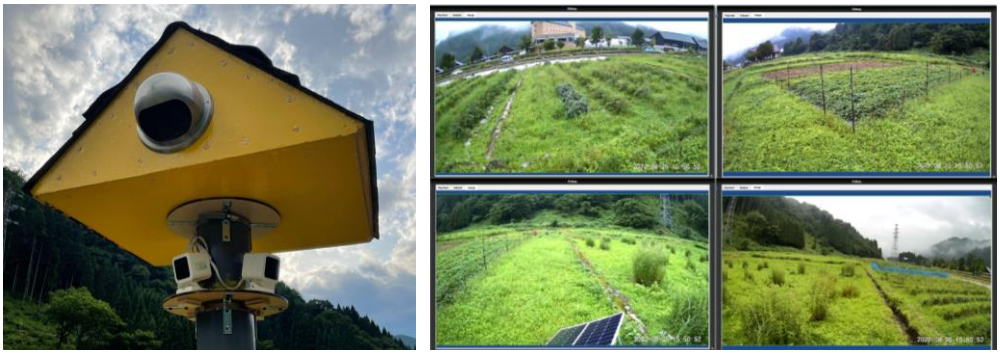
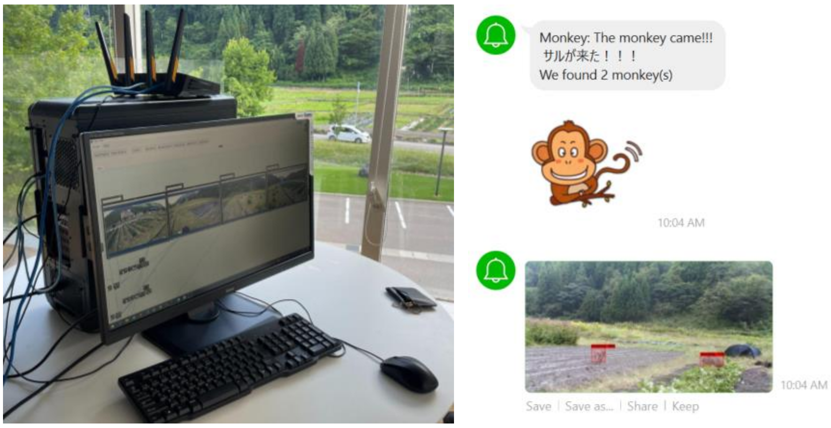


Figure 6: Four cameras are installed beneath the shutter (left) and Farmland Monitoring (right).

5. Experiments and Results

5.1 On-Field Experiments

We have set up a system in our sweet potato field to test its effectiveness. In this setup, we ran the detection model through an AI station computer (CiRA Core platform). The system will be operational 24/7 from September to November 2023. Whenever monkeys are detected, the system will send out a notification to the Line application. We have observed that the monkey often trespasses the farmland during the daytime. The prototype could detect monkeys in real-time, and the system can also send a message, sticker, and image to notify the user. However, misclassification might occur due to the noise in the images and the distance of the monkey from the system.



Monkey: The monkey came!!!
サルが来た！！！
We found 2 monkey(s)

Figure 7: The AI Station is inside the building (left) and Line Notification (right).

5.2 Off-Field Experiments

Table 2 contains the results of our k-fold cross-validation analysis. We were concerned that our system's precision rate was relatively low, particularly in identifying monkeys. This presented a significant challenge for us. To address this problem, we introduced Regions of Interest (ROI) to locate the relevant area and detect the monkey in real-time stream video. By avoiding areas where monkeys were likely far from the farmland, we could prevent misdetection and improve our system's performance.

	x_1	x_2	x_3	x_4	x_5	E
AP@0.5	0.7453	0.7505	0.7281	0.7336	0.7529	0.7421
Precision	0.7021	0.7351	0.6982	0.7357	0.7834	0.7310
Recall	0.8595	0.8315	0.8847	0.8426	0.8128	0.8462

Table 2: K-Fold Cross Validation Result.

6. Summary and Future Work

We have developed a real-time system to detect Japanese Macaque Monkeys, tested in both on-field and off-field settings. The system was successfully trained and operated on a Windows PC with 32 GB RAM, a 64-bit Operating System, an Intel(R) Core(TM) i9-9900K CPU@3.60GHz, and an Nvidia GeForce RTX 3080 10 GB graphics processing unit (GPU). Using a real-time streaming camera, the system has shown high accuracy and classification performance. It can alert farmers in advance to prevent damage caused by monkeys in sweet potato fields. This study explores the use of deep learning to prevent monkey trespassing in agricultural environments. We emphasize the importance of performance analysis and system optimization and offer a comprehensive perspective on developing an effective and practical solution for mitigating human-wildlife conflicts in local agriculture. It is important to note that the system may occasionally misclassify monkey images due to low resolution and monkey distance from the camera. Nevertheless, it remains a valuable tool for alerting farmers to the presence of monkeys and enabling them to take preventive measures.

Moving forward, our team has set its sights on compiling datasets of additional wildlife that may threaten farmland. By doing so, we aim to mitigate the danger of wild animals encroaching on rural areas. Moreover, we attempt to integrate low-power operation cameras into our system, connecting it to the Internet of Things (IoT) network. This approach can revolutionize agriculture, empowering farmers to monitor their crops and livestock remotely. Farmers can proactively identify potential threats by leveraging this technology and take appropriate measures to avert damage.

Acknowledgments

We would like to express our gratitude to the College of Advanced Manufacturing Innovation at King Mongkut's Institute of Technology, Ladkrabang, Thailand, for providing the necessary deep learning platform and software to support our research project.

References

[1] Ministry of Agriculture, Forestry and Fisheries. Summary of the Annual Report on Food, Agriculture and Rural Area in Japan. May 2021. [Online]. Retrieved April 5, 2022, from https://www.maff.go.jp/e/data/publish/.

[2] Ryoki KAMESAKA, Yukinobu HOSHINO (2018), Prototyping and evaluation of the prevention system for beast damage of the agricultural product. https://doi.org/10.14864/fss.34.0_334.

[3] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 779-788).

[4] M, Vamshi & Marupaka, Navateja & Nallamothu, Sri & Naureen, Ayesha. (2023). A Deep Learning Approach – Monkey Detection using YOLOv7. 1-7. 10.1109/EASCT59475.2023.10393326.

[5] Jomtarak, R., Faikhamta, C., Prasoplarb, T., & Lertdechapat, K. (2023). CiRA-Core: The connector for developer teachers and user teachers to artificial intelligence. In Proceedings of the Conference on Artificial Intelligence. https://doi.org/10.1007/978-981-99-8255-4_2.

[6] Morgan Quigley, Brian Gerkey, and William D. Smart. (2015). Programming Robots with ROS. O'Reilly Media.