

Development of an AI-based Image Analysis System to Calculate the Visit Duration of a Green Blow Fly on a Strawberry Flower

Hiroki Taniguchi¹, Yuki Tsukuda², Ko Motoki¹, Tanjuro Goto¹, Yuichi Yoshida¹
and Ken-ichiro Yasuba^{1*}

¹Graduate School of Environmental and Life Science, Okayama University, Okayama 700-8530, Japan

²School of Agriculture Okayama University, Okayama 700-8530, Japan

Pollinator insects are required to pollinate flowers in the production of some fruits and vegetables, and strawberries fall into this category. However, the function of pollinators has not been clarified by quantitative metrics such as the duration of pollinator visits needed by flowers. Due to the long activity time of pollinators (approximately 10-h), it is not easy to observe the visitation characteristics manually. Therefore, we developed software for evaluating pollinator performance using two types of artificial intelligence (AI), YOLOv4, which is an object detection AI, and VGG16, which is an image classifier AI. In this study, we used *Phaenicia sericata* Meigen (green blow fly) as the strawberry pollinator. The software program can automatically estimate the visit duration of a fly on a flower from video clips. First, the position of the flower is identified using YOLO, and the identified location is cropped. Next, the cropped image is classified by VGG16 to determine if the fly is on the flower. Finally, the results are saved in CSV and HTML format. The program processed 10 h of video (collected from 07:00 h to 17:00 h) taken under actual growing conditions to estimate the visit durations of flies on flowers. The recognition accuracy was approximately 97%, with an average difference of 550 s. The software was run on a small computer board (the Jetson Nano), indicating that it can easily be used without a complicated AI configuration. This means that the software can be used immediately by distributing pre-configured disk images. When the software was run on the Jetson Nano, it took approximately 11 min to estimate one day of 2-h video. It is therefore clear that the visit duration of a fly on a flower can be estimated much faster than by manually checking videos. Furthermore, this system can estimate the visit durations of pollinators to other flowers by changing the YOLO and VGG16 model files.

Key Words: deep learning, fly, microcomputer, VGG16, YOLO.

Introduction

Pollinators are necessary for the production of fruits and vegetables. In fact, fruit, vegetable, or seed production of 87 of the leading 125 global food crops depend on animal pollination (Klein et al., 2007). The flowers of many kinds of fruits, trees, and vegetables are pollinated by insects, including the strawberry. Pollinators such as honey bees (Garibaldi et al., 2013) and bumblebees (Goulson, 2010) have been used, but recently, for strawberries, *Phaenicia sericata* Meigen (green blow

fly) has been used as an alternative pollinator to the honey bee (Hanada et al., 2016).

Many strawberry growers have used pollinators in greenhouses, but it has been reported that these pollinators do not work well and inadequate pollen levels reduce marketable yields (Zebrowska, 1998). A decrease in marketable yields is damaging to strawberry growers. However, it is not clear how long strawberry pollinators must visit a flower to pollinate and produce normal fruits. Similarly, it is not known how long a fly must visit a flower for normal fruit production. To determine the duration and number of visits to a flower by flies, we must capture these visits on video and watch the whole video. In an experiment conducted in the month of July by Karbassioon and Stanley (2023), pollinators were observed to be active from approximately 08:00 h to 20:00 h, although there were differences among

Received; March 26, 2024. Accepted; July 9, 2024.

First Published Online in J-STAGE on October 1, 2024.

This study was partly supported by JSPS KAKENHI grant number 23K05208.

* Corresponding author (E-mail: ysbken@okayama-u.ac.jp).



them. In our preliminary experiments, (unpublished data) flies are active from 07:00 h to 17:00 h. Because of these long hours of pollinator activity, it is not easy to observe the visitation characteristics manually. In contrast, if it were possible to have a computer do the work, it would be easier to analyze the activities of the pollinators.

In recent years, machine learning has been attracting attention as computers become more powerful and affordable. In particular, deep learning, which uses three or more intermediate layers for learning, is used in artificial intelligence (AI), and is widely used in the fields of image classification and object detection. In agriculture, AI (Uchimura et al., 2021) has been used for image classification tasks, such as the classification of strawberry fruit shapes (Ishikawa et al., 2018) and the rapid over-softening and shelf life of persimmon fruits (Suzuki et al., 2022), as well as for object detection tasks, such as blueberry fruit detection (Gonzalez et al., 2019). For image classification, various models have been developed (Simonyan et al., 2013; Singh et al., 2018) using convolutional neural networks (CNNs). Object detection models include You Only Look Once (YOLO), which enables real-time object detection (Redmon et al., 2016). In addition, some of the libraries needed to create CNN are available as open source. For example, the source code needed to create YOLO was made publicly available (Bochkovskiy et al., 2020), making it possible to create discriminators cheaply. Recently, seven olive cultivars were classified with 95.91% accuracy using image classification (Ponce et al., 2019), and the number of leaves of *Arabidopsis thaliana* could be detected with approximately 88% accuracy using Tiny-YOLOv3 with object detection (Buzzy et al., 2020). Very accurate models using CNN for crop classification and YOLO for object detection have been reported, proving that it is possible to classify and detect target crops at a high level of accuracy.

Therefore, we developed software that can analyze video of flowers to monitor the activity of pollinators using deep learning. Using the developed software, we created an environment that enables the time a fly visits a flower to be conveniently estimated. We confirmed that a small computer board with a GPU can be used as the hardware to run the software.

Materials and Methods

1. AI development and performance evaluation for the developed software

1) Hardware specifications and software libraries for AI model development

The image classification and object detection AI models were trained on a personal computer running Windows 11 Pro (Microsoft Co., USA) with a CPU (Intel Core i5-12400; Intel Co., USA) and GPU (GeForce RTX 4080; NVIDIA, USA). The object

detection AI model was trained using the Python programming language (version 3.7.9; Python Software Foundation, USA) and several open-source libraries. The open-source libraries OpenCV (version 4.7.0.72; Intel Co.) and Pillow (version 9.4.0) were used for image processing. CUDA (version 11.8; NVIDIA), cuDNN (version 8.7.0; NVIDIA), and Darknet (Joseph Redmon), which is an open-source framework for neural networks, were used as the execution environment. YOLOv4 was used as the object detection AI model. Python (version 3.9.16; Python Software Foundation) and Keras (version 2.9.0) with a TensorFlow GPU (version 2.9.2; Google Inc., USA) were used as the backend for training the neural networks, and VGG16 (Simonyan and Zisserman, 2014) was used to create the image classification AI model.

2) AI Creation Method

The objective of this study was to create software that can determine the duration a pollinator has perched on a flower using two types of AI: YOLO, an object detector, and the other is VGG16, an image classifier.

A video of flowers in bloom in a strawberry field for a certain duration was used as input. YOLO detects the position of the flowers in an image cropped from a captured video and acquires a partial area surrounding the recognized object using a rectangle called a bounding box (BB), and this was used to detect the positions of the strawberry flowers and crop that area of the image, even if the flowers moved. Strawberry bunches move up and down slightly over time. The VGG16 was used to classify the cropped images and determine if the strawberry flowers were being visited by pollinators.

The open-source framework for neural networks called Darknet was used to create the YOLO for object detection. The YOLOv4 architecture was trained. The training method was based on an official document (<https://github.com/AlexeyAB/darknet#how-to-train-to-detect-your-custom-objects>), and a data file and “cfg” file were created. The data file specifies the folder paths of the training and validation images. The cfg file specifies the training conditions for the AI model. To improve learning efficiency, the cfg file was modified to specify 64 batches, 32 subdivisions, a momentum of 0.949, a learning rate of 0.001, and a max_batches of 6,000. For the VGG16, we used the open-source library Keras and a pre-trained VGG16, and performed transfer learning to train convolution layers 15 and above (in the trained VGG16). The pre-trained model was trained using ImageNet, a large training dataset. The basic settings for the model were “SDG” for the solver, a learning rate of 0.0001, and “sparse_categorical_crossentropy” for the loss function. For the “class_mode” argument of the training and validation images, we adopted the “binary” option and specified 15 epochs. These changes were made for training.

To create the AI model, images for training and discrimination were taken from October to December

2021 and from October to December 2022 in a greenhouse (Okayama City, Okayama Prefecture) where strawberries were cultivated using flies as pollinators. The greenhouse was operated by the Faculty of Agriculture, Okayama University. A digital camera (D5100; Nikon Co., Japan) and smartphone (iPhone 12 mini; Apple Inc., USA) were used to take most images. Three types of objects were collected: a fly in a non-flowering area (fly), a strawberry flower with no flies (flower), and a fly on a flower (fly_flower). A total of 4071 digital images in JPG format were acquired. For the object detector (YOLO), the images were classified into the three categories described above and annotated using annotation software (Labellmg, MIT, Computer Science and Artificial Intelligence Laboratory), and the annotated images were randomly divided into a training dataset (1,176 fly images, 1,053 flower images, and 1,028 fly_flower images) and a test dataset (271 fly images, 272 flower images, and 271 fly_flower images). Images of each of the three types are shown in Figure 1.

For the image classifier, 1,300 flower images and 1,145 fly_flower images were selected from the same images used to train YOLOv4, and these images were replicated using OpenCV with the contrast and luminance changed (2,600 flower images, 2,290 fly_flower images). The images in each dataset were randomly divided into two sets using a ratio of approximately 8:2 to obtain a training dataset (2,080 flower images, 1,936 fly_flower images) and a test dataset (520 flower images, 484 fly_flower images).

For the performance evaluation, to verify whether the trained YOLOv4 and trained VGG16 could be used in the developed software, the accuracy, precision, recall, F1-score, and ROC-AUC (VGG16 only) were obtained on the test dataset. We evaluated these performance metrics along with the confidence score, which was calculated by YOLOv4. Confidence score thresholds of 0.5 and 0.75 were evaluated separately.

YOLOv4 can distinguish between flower and fly flower images. However, in this study, YOLOv4 was used to locate the flowers, and VGG16 was used to accurately recognize images of a fly on a flower. Therefore, it is only important for YOLOv4 to accurately locate the flower and its performance evaluation should focus only on the localization of flowers. To verify the

accuracy of BBs, we evaluated a case in which YOLOv4 detects either a flower or fly_flower. In this case, BB confidence scores (0.5 and 0.75) and IoU confidence score (0.75) were used. In addition, the flowerIoU value of the original index was calculated and evaluated separately. The flowerIoU was calculated regardless of whether the recognition result was correct or incorrect when the AI created a BB of a flower or a fly_flower for an image of a flower or fly_flower, and the accuracy when cropping the image was verified by taking the average.

To verify the accuracy of the two AI models, we used YOLO to identify images cropped from the video and VGG16 to identify the cropped image of the area surrounding the flower. The verification method for the developed AI model's performance is described below. Preliminary experiments showed that the active time of the pollinators was mostly between 07:00 h and 17:00 h. Therefore, to create a test dataset, we used videos taken in the greenhouse on three days: 2 Nov., 16 Dec. 2022, and 17 Jan. 2023, using a digital video camera (HC-VX992MS; Panasonic Co., Japan). The videos were converted to MP4 format and included the 10 hours from 07:00 h to 17:00 h. An image for cropping was automatically extracted from the video every 2 min, and the cropped image was saved. After object detection with a trained YOLOv4, we defined an area 1.5 times larger than the recognition area, extracted the area, and saved it. We used an area 1.5 times larger than the recognition area because the size of the background area of the test dataset images was almost the same as that of the training dataset images used for VGG16. This area was updated every 30 min to adjust the flower position because strawberry bunches move over time, as mentioned above. All cropped and trimmed images were saved in JPG format. From these images, 100 flower images and 100 fly_flower images were randomly selected (200 cropped images and 200 trimmed images in total), and these were used as the test dataset. The cropped and trimmed images were then classified using YOLOv4 and VGG16, respectively. We confirmed whether the decision of the AI was correct by checking the images visually.



Fig. 1. Examples of images used to train the AI models (Left: a pollinator (fly), Center: a strawberry flower (flower), Right: a fly on a strawberry flower (fly_flower)).

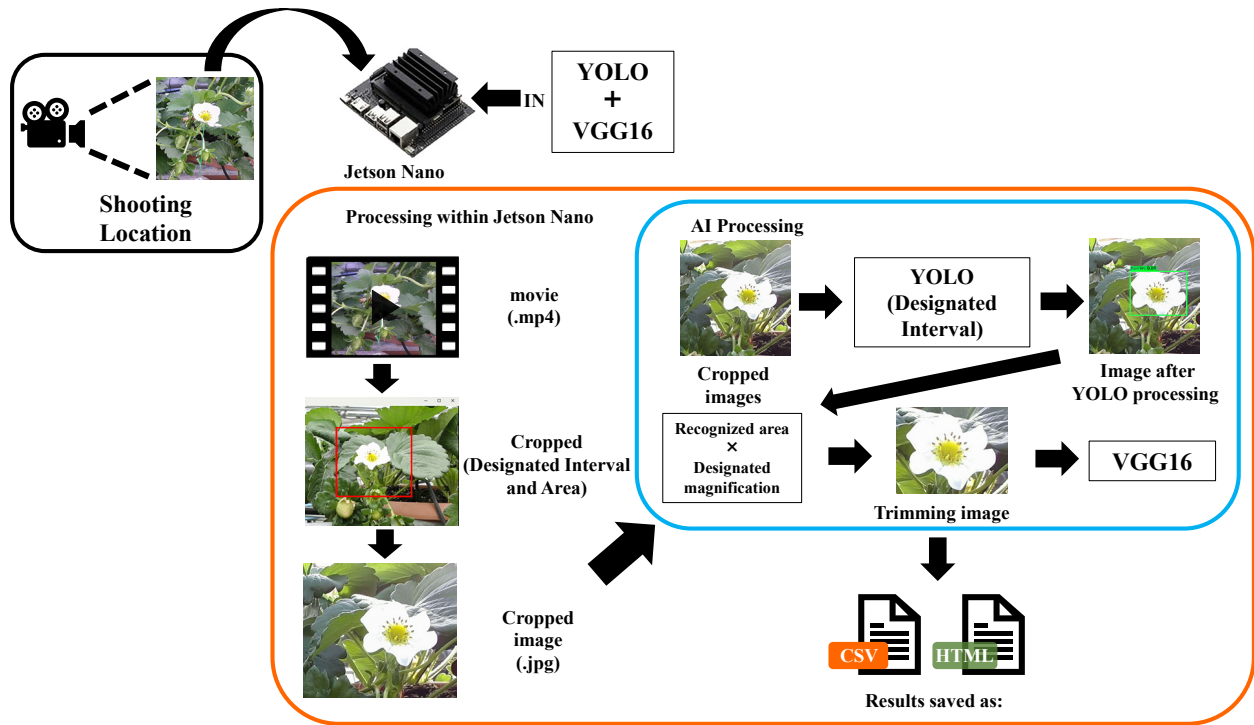


Fig. 2. Overall system overview.

2. Software incorporating the developed AI

1) Development environment for the software

We developed software that incorporated the developed AI models, as mentioned in the previous section. The software was created in the same environment as that of the AI models. The software was designed with a graphical user interface (GUI) so that anyone can operate it, and Tkinter library, a Python standard, was used as the GUI development tool.

2) Software summary

The user must prepare a video of strawberry flowers in advance. The software roughly specifies the area around the flower from the video, and the specified area is cropped as an image at fixed intervals. Then, the YOLOv4 embedded with the software is used at the specified interval to locate the position of the flower. A cropped image is further trimmed in an area slightly larger than the AI-specified area to avoid cropping some of the flowers in the areas identified by the AI model. The trimmed image is classified by VGG16 to determine whether a fly is on the flower in the image. The software allows the user to specify the interval (seconds) used to extract the images from the video, the increase in the area for trimming, and the interval (minutes) used to acquire the flower area information using YOLOv4. Because it takes some time for YOLOv4 to recognize the location of a flower each time it is used, the images are extracted at a certain interval to allow the software to run faster. After the whole video has been processed as described above, the date and time of the cropped images, the VGG16 recognition results,

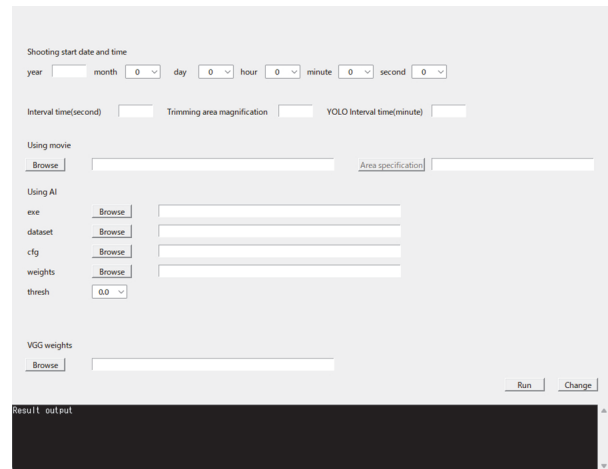


Fig. 3. Software configuration screen.

and the total duration of each flower visit are output to CSV and HTML files (Fig. 2).

3) Hardware and operating environment for running the software

We confirmed that a small computer board with a GPU could be used as the hardware to run the software. The computer board (Jetson Nano; NVIDIA) for embedded use with a GPU was evaluated as the machine for running the software. The software was installed on a Jetson Nano Developer Kit B01 (NVIDIA) using Jetpack 4.6.3.

The software configuration screen is shown in Figure 3. Users must enter the date of the captured

video, trimming interval (in seconds), trimming magnification, and the interval for obtaining the flower area information by YOLOv4 (minutes). After selecting the video to be used (MP4 format), the user selects which region of the video should be cropped and classified by the AI model. To make it easier to select an area, an image is extracted from the video and the area can be selected by dragging the mouse while checking the image on the screen (Fig. 4). YOLOv4 can search flower positions in this selected area. After entering all of the above items, the “change” button is pushed to prepare the software for operation, and then the “run” button is pushed to enable the AI to classify the input until the end of the video. During the classification, the results are output to the operation screen to confirm that the classification is being performed. The cropped and trimmed images are saved as time-stamped + JPG format for confirmation.

4) Software operation verification and performance evaluation

To evaluate the performance of the software by actually running it, we used videos taken in the greenhouse on 2, 17, 23, 29, and 30 Nov. 2022, and 7 Dec. 2022. These videos were different from the ones used to develop the AI. The software accuracy was calculated by comparing the AI recognition results with the results of visual inspection of each image. In addition, we compared the visit duration of a fly to a flower confirmed by visual inspection of the video, with that obtained using the image recognition AI model.

The visit duration of a fly to a flower confirmed by visual inspection of a video was judged by the time a fly flew to a flower up to the time it flew away. The AI models used for both evaluations were the trained YOLOv4 and trained VGG16. The interval for extracting the images from the video was set to 30 s, and YOLOv4’s flower location information was updated when the video advanced by 30 min. The duration of the visit of a fly on a flower was the accumulated cropped image interval over which VGG16 recognized a fly on a flower (duration + 30 s). The trimming magnification was set to 1.5 times the area of the YOLOv4-specified region, and the confidence score was set to



Fig. 4. Selection of the cropped area.

0.5. To measure the differences in the estimated fly visit duration obtained using different video clip intervals, we calculated the results for each video using clip intervals of 10, 20, and 30 s (changing the integration time to 10, 20, and 30 s, respectively).

In addition, a 2-h video was prepared to measure the time it takes for the software to run. The runtimes when the sizes of the images cropped from the video were 200×200 pixels, 400×400 pixels, and 800×800 pixels were evaluated to determine whether the operating time varied depending on the size of the cropped image (interval = 30 s).

Results and Discussion

1. AI model development and performance evaluation for use in software

The accuracy, precision, recall, and F1-score of the trained YOLOv4 were 56%, 0.73, 0.56, and 0.46, respectively, when the confidence score was 0.5. They were 53%, 0.8, 0.53, and 0.43, respectively, when the confidence score was 0.75 (Table 1). For cherry fruit detection using YOLOv4, the F1-score of the model was reported to be 0.935 (Gai et al., 2023), and hence the accuracy of our developed object detection and recognition model was low. Here, the trained YOLOv4 made errors in detection on the test dataset mostly because it misidentified flower images as fly_flower images, and the AI was not able to clearly determine the difference between the flower and fly_flower classes. The results (Table 2) focusing only on the position of the flowers showed that the accuracy, precision, recall, F1-score, and flowerIoU values were 98.5%, 0.99, 0.99, 0.99, and 0.97, respectively, when the confidence score was 0.5. Moreover, they were 84.5%, 0.99, 0.85, 0.92, and 0.97, respectively, when the confidence score was 0.75. Hence, focusing only on the location of the flowers resulted in a significant improvement in AI accuracy. If the confidence score was increased, objects

Table 1. The accuracy, precision, recall, and F1-score of the trained YOLOv4 model (for confidence score thresholds of 0.5 and 0.75).

Confidence Score	Accuracy (%)	Precision	Recall	F1-score
0.5	56	0.73	0.56	0.46
0.75	53	0.8	0.53	0.43

Table 2. The accuracy, precision, recall, F1-score, and flowerIoU of the trained YOLOv4 model when detecting only the position of the flower (for confidence score thresholds of 0.5 and 0.75).

Confidence Score	Accuracy (%)	Precision	Recall	F1-score	FlowerIoU
0.5	98.5	0.99	0.99	0.99	0.97
0.75	84.5	0.99	0.85	0.92	0.97

that had been detected became undetectable and the detection accuracy and F1-score decreased. The trained YOLOv4 could determine the position of a strawberry flower, but could not accurately classify whether the image was a flower or fly_flower image. Stark et al. (2023) reported that YOLOv5nano, YOLOv5small, and YOLOv7tiny could be used for object recognition and classification of eight groups of flower-visiting arthropods. When they set the confidence score for YOLOv5nano, YOLOv5small, and YOLOv7tiny to 0.2, 0.3, and 0.1, accuracies of 94.50%, 96.24%, and 95.08%, respectively, were achieved. This shows that high accuracy can be achieved when the confidence score is set relatively low. In this study, the accuracy was highest when the confidence score was 0.5, which is a value greater than that used in study by Stark et al. (2023). However, the optimal confidence score is likely to vary depending on the growing environment and other factors. Therefore, it should also be possible to change the confidence score in the software.

The discrimination accuracy, precision, recall, F1-score, and ROC-AUC of the trained VGG16 were 95%, 0.96, 0.96, 0.95, and 0.97, respectively (Table 3). Suzuki et al. (2022) reported a test set detection accuracy of 0.87 and an F1-score of 0.85 for the detection of over-softening in persimmons. The performance results obtained in this study were more accurate than those results. The trained YOLOv4 accuracy and F1-score were 56% and 0.46, respectively, indicating the performance of VGG16 improved by 39 percent points and 0.49 with respect to accuracy and F1-score than when only YOLO used. Therefore, we consider it to be more practical than using only YOLO. Two test set samples that the trained VGG16 detected incorrectly are shown in Figure 5. Flowers moved while the video is being taken, so the AI can classify the image incorrectly when the flower itself is oriented horizontally to the camera or when part of the flower is hidden by a leaf.

Focusing only on flower localization, the trained YOLOv4 was found to perform very well, with an accuracy of 98.5% and an F1-score of 0.99. Even with the trained VGG16, high performance was observed, with an accuracy of 95% and an F1-score of 0.95. The trained VGG16 showed no difference in the classification performance for both the flower and fly_flower images. Because both YOLOv4 and VGG16 were found to be highly accurate when used for their intended purposes, we decided to create software using both AI models.

Table 3. The accuracy, precision, recall, F1-score, and ROC-AUC of the trained VGG16 model.

Accuracy (%)	Precision	Recall	F1-score	ROC-AUC
95.5	0.96	0.96	0.95	0.97

2. Software incorporating the developed AI

For the developed software to work, a video of strawberry flowers in the greenhouse was used. The software incorporating the AI models evaluated in the previous section was tested to see if it could estimate the visit durations of the flies.

To determine whether the software's estimated visit duration of flies on flowers was accurate, we compared the AI-calculated daily visit durations with human visual checks of the daily visit durations of flies on flowers (Fig. 6). The visit durations per hour are shown in Figure 7. The percentage errors of the AI models with respect to visual confirmation were 4.3%, 15.4%, 5.8%, 0.9%, 1.7%, and 5.1%, for videos taken on 2, 17, 23, 29, and 30 Nov., and 7 Dec. 2022, respectively, and the average percentage error over the six days was 5.5%. In contrast, the average rate of trimmed images that the software correctly recognized over the six days was approximately 97%. The differences between the overall visit duration of the flies on flowers visually confirmed by video and calculated using software were 426, 1,527, 580, 86, 166, and 506 s, respectively. The average value was 549 s.

The data from Nov. 17, when recognition accuracy was the lowest, was examined to analyze the software performance in the actual growing environment. We found that the flowers were hidden by the strawberry leaves due to movement of the flowers, which prevent-



Fig. 5. An example of a trained VGG16 detecting incorrectly on a test dataset. (Left: part of the flower is occluded by a leaf. Right: the flower is oriented horizontally to the camera).

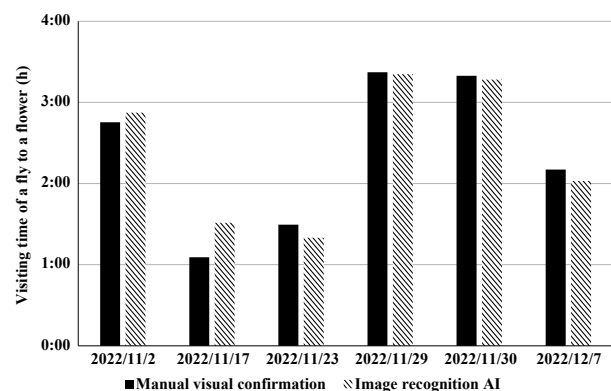


Fig. 6. Comparison of the visit duration of a fly to a flower obtained by visual confirmation and by the image recognition AI model (interval = 30 s).

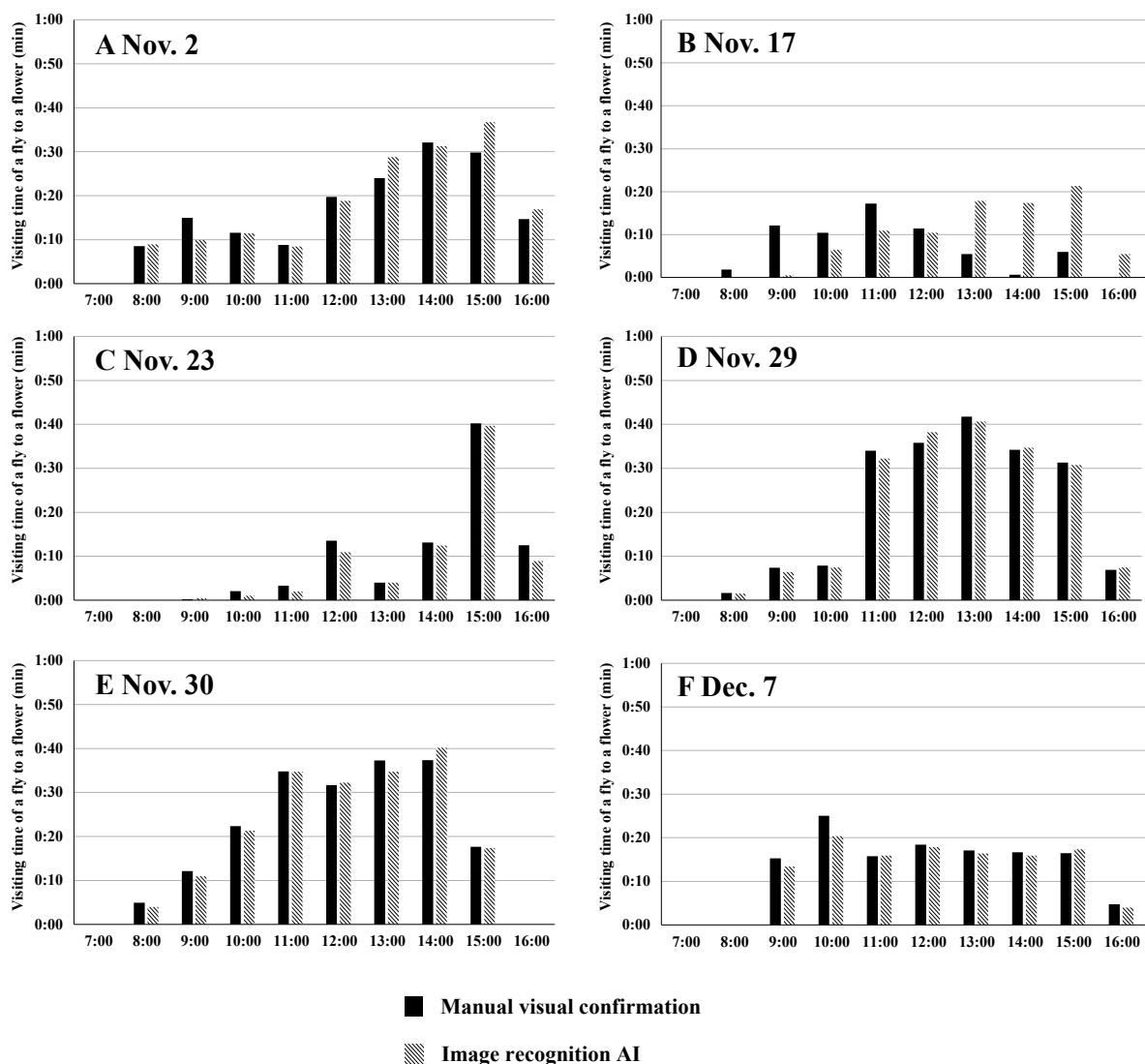


Fig. 7. Total visit duration of a fly to a flower obtained by visual confirmation and by image recognition AI measured every hour (interval = 30 s). A: Nov. 2, 2022, B: Nov. 17, C: Nov. 23, D: Nov. 29, E: Nov. 30, F: Dec. 7.

ed accurate recognition. Other misrecognized images showed that the flowers were oriented horizontally, and in most cases, the colors of the flowers were changed by strong natural light. Overall, differences in the visit duration of flies on flowers were not substantial (Max: 730 s, Min: 8 s, Avg: 131 s, Fig. 7), and it was clear that there were no large changes in the error of the visit durations at any times of the day.

Visit duration was estimated by changing the video cropping interval (10, 20, and 30 s) and comparing with the results visually confirmed by video (Fig. 8). With the exception of the data for 29 Nov. and 7 Dec., when comparing the results for the various interval times the maximum error was 370 s, and the average error was 225 s. The percentage error times for image cropping intervals of 10, 20, and 30 s relative to visual confirmation by video were 6.4%, 5.9%, and 5.5%, respectively. Thus, when the video cropping interval was varied, no noteworthy effect was observed in the estimate of the

visit duration or the percentage error with respect to visual confirmation. When estimating the visit duration, it was not expected for a video cropping interval lower than 30 s to have an effect on the software.

The time required to run the software was approximately 11 min for a 2-h movie, regardless of the image size (200×200 pixels, 400×400 pixels, or 800×800 pixels). This suggests that the software can run at a nearly constant speed regardless of the image size.

Considering that the estimation time was approximately 11 min for a 2-h movie and that the error in the estimated visit duration obtained by image recognition AI is not very large, it was clear that the software can be used to estimate the time that pollinators visit flowers much faster than by visual confirmation. Therefore, the software can automatically estimate the visit duration of a fly to a flower, which reduces the effort compared with doing it manually.

The performance of an AI model is determined by

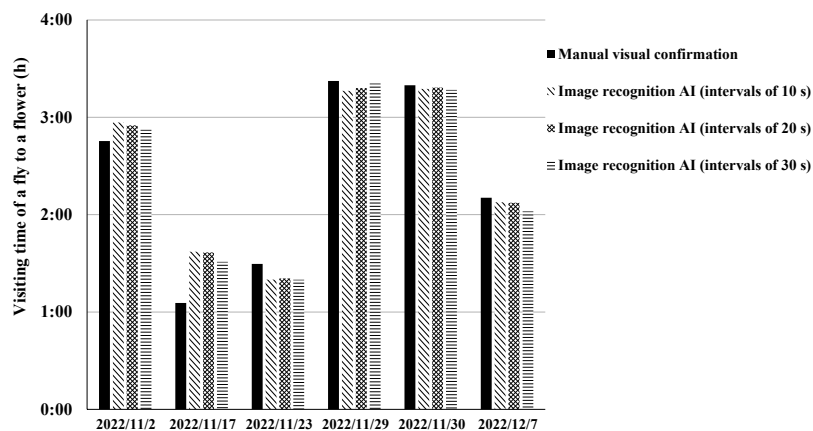


Fig. 8. Comparison of the visit duration of a fly to a flower obtained by visual confirmation and by the image recognition AI model (for video intervals of 10 s, 20 s, and 30 s).

the weights file generated after training per 1,000 epochs (Takayama et al., 2021). AI performance can be modified by changing the weights file used by this software. A feature of this software is that the AI used can be changed only by changing the file specifications, so this software is very flexible. Moreover, this software the confidence score, crop area, and interval time used by the AI model can be adjusted. Hence, users can easily find the best settings for estimating the duration of visits by pollinators on flowers. Moreover, using the weights file created using images of other flowers and other pollinators, the software could be used to estimate the visit durations for other pollinators to other flowers quantitatively.

The hardware, the Jetson Nano, installs its OS from a microSD card. Therefore, image files with pre-configured YOLOv4 and VGG16 models and software executables can be installed on it. It is possible to distribute this image file. In general, building an AI environment is quite challenging. However, if the image file can be distributed as described above, anyone could use this software by downloading the image file to a microSD card in the recommended manner.

Conclusion

The developed software, which uses a trained YOLOv4 to acquire flower location information and a trained VGG16 to classify cropped images to estimate the visit duration of pollinators to strawberry flowers, had a recognition accuracy of 97%, and the average rate of the error was 5.5%. If this recognition accuracy is acceptable, we believe that this software is worth using. Furthermore, the inference time was approximately 11 min for a 2-h video. This means that inference can be performed much faster and automatically as opposed to being done manually.

Because of the software design, it is possible to estimate the visit durations of other pollinators and other flowers by changing the files used for the YOLO and VGG16 models. Moreover, the use of the Jetson Nano

hardware makes it easy to estimate the visit durations of pollinators to other flowers.

Literature Cited

- Bochkovskiy, A., C.-Y. Wang and H.-Y. M. Liao. 2020. YOLOv4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004. 10934. DOI: 10.48550/arXiv.2004.10934.
- Buzzy, M., V. Thesma, M. Davoodi and J. M. Velni. 2020. Real-time plant leaf counting using deep object detection networks. *Sensors* 20: 6896. DOI: 10.3390/s20236896.
- Gai, R., N. Chen and H. Yuan. 2023. A detection algorithm for cherry fruits based on the improved YOLO-v4 model. *Neural Comput. Appl.* 35: 13895–13906.
- Garibaldi, L. A., I. Steffan-Dewenter, R. Winfree, M. A. Aizen, R. Bommarco, S. A. Cunningham, C. Kremen, L. G. Carvalheiro, L. D. Harder, O. Afik, I. Bartomeus, F. Benjamin, V. Boreux, D. Cariveau, N. P. Chacoff, J. H. Dudenhöffer, B. M. Freitas, J. Ghazoul, S. Greenleaf, J. Hipólito, A. Holzschuh, B. Howlett, R. Isaacs, S. K. Javorek, C. M. Kennedy, K. M. Krewenka, S. Krishnan, Y. Mandelik, M. M. Mayfield, I. Motzke, T. Munyuli, B. A. Nault, M. Otieno, J. Petersen, G. Pisanty, S. G. Potts, R. Rader, T. H. Ricketts, M. Rundlöf, C. L. Seymour, C. Schüepp, H. Szentgyörgyi, H. Taki, T. Tschamtker, C. H. Vergara, B. F. Viana, T. C. Wanger, C. Westphal, N. Williams and A. M. Klein. 2013. Wild pollinators enhance fruit set of crops regardless of honey bee abundance. *Science* 340: 1608–1611.
- Gonzalez, S., C. Arellano and J. E. Tapia. 2019. Deepblueberry: quantification of blueberries in the wild using instance segmentation. *IEEE Access* 7: 105776–105788.
- Goulson, D. 2010. Impacts of non-native bumblebees in western Europe and north America. *Appl. Entomol. Zool.* 45: 7–12.
- Hanada, A., Y. Yoshida, T. Sato, T. Goto, K. Yasuba and Y. Tanaka. 2016. Utilization of *Phaenicia sericata* Meig. (Green Blow Fly) as an alternative pollinator to honey bee. *Hort. Res. (Japan)* 15: 161–169 (In Japanese with English abstract).
- Ishikawa, T., A. Hayashi, S. Nagamatsu, Y. Kyutoku, I. Dan, T. Wada, K. Oku, Y. Saeki, T. Uto, T. Tanabata, S. Isobe and N. Kochi. 2018. Classification of strawberry fruit shape by machine learning. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science (ISPRS)* 42: 463–470.

- Karbassioon, A. and D. A. Stanley. 2023. Exploring relationships between time of day and pollinator activity in the context of pesticide use. *Basic Appl. Ecol.* 72: 74–81.
- Klein, A. M., B. E. Vaissiere, J. H. Cane, I. Steffan-Dewenter, S. A. Cunningham, C. Kremen and T. Tscharntke. 2007. Importance of pollinators in changing landscapes for world crops. *Proc. Royal Society* 274: 303–313.
- Ponce, J. M., A. Aquino and J. M. Andújar. 2019. Olive-fruit variety classification by means of image processing and convolutional neural networks. *IEEE Access* 7: 147629–147641.
- Redmon, J., S. Divvala, R. Girshick and A. Farhadi. 2016. You only look once: unified, real-time object detection. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 779–788.
- Simonyan, K. and A. Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*. DOI: 10.48550/arXiv.1409.1556.
- Simonyan, K., A. Vedaldi and A. Zisserman. 2013. Deep inside convolutional networks: visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*. DOI: 10.48550/arXiv.1312.6034.
- Singh, A. K., B. Ganapathysubramanian, S. Sarkar and A. Singh. 2018. Deep learning for plant stress phenotyping: trends and future perspectives. *Trends Plant Sci.* 23: 883–898.
- Stark, T., V. Ştefan, M. Wurm, R. Spanier, H. Taubenböck and T. M. Knight. 2023. Yolo object detection models can locate and classify broad groups of flower-visiting arthropods in images. *Sci. Rep.* 13: 16364. DOI: 10.1038/s41598-023-43482-3.
- Suzuki, M., K. Masuda, H. Asakuma, K. Takeshita, K. Baba, Y. Kubo, K. Ushijima, S. Uchida and T. Akagi. 2022. Deep learning predicts rapid over-softening and shelf life in persimmon fruits. *Hort. J.* 91: 408–415.
- Takayama, T., T. Yashiro, S. Sanada, T. Katsuragi and R. Sugiura. 2021. Evaluation of detection accuracy of image recognition for automatic counting of rice planthoppers captured on sticky boards. *Agric. Info. Res. (Japan)* 30: 174–184 (In Japanese with English abstract).
- Uchimura, Y., Y. Yoshida, T. Goto and K. Yasuba. 2021. Development of cabbage classification system by machine learning. *Hort. Res. (Japan)* 20: 469–475 (In Japanese with English abstract).
- Zebrowska, J. 1998. Influence of pollination modes on yield components in strawberry (*Fragaria × ananassa* Duch.). *Plant Breed.* 117: 255–260.