

UC San Diego

UC San Diego Previously Published Works

Title

AlertTrap: On Designing An Edge-Computing Remote Insect Monitoring System

Permalink

<https://escholarship.org/uc/item/2qm0b4jz>

Authors

Pham, Duy A

Le, An D

Pham, Dong T

et al.

Publication Date

2021-12-22

DOI

10.1109/nics54270.2021.9701558

Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at <https://creativecommons.org/licenses/by/4.0/>

Peer reviewed

AlertTrap: On Designing An Edge-Computing Remote Insect Monitoring System

Duy A. Pham
Computer Science Department
Bonn-Rhein-Sieg University of
Applied Sciences
Bonn, Germany
duyanhpham@outlook.com

An D. Le
Electrical and Computer
Engineering Department
University California San Diego
San Diego, USA
d0le@ucsd.edu

Dong T. Pham
Graduate School of Bioresource
and Bioenvironmental Sciences
Kyushu University
Fukuoka, Japan
phamtdong0406@gmail.com

Hien B. Vo
Electrical and Computer
Engineering Department
Vietnamese-German University
Binh Duong, Vietnam
hien.vb@vgu.edu.vn

Abstract—Fruit flies become one of the most worrisome insect species to fruit yields. AlertTrap proposes and tests the constituent components to construct an efficient autonomous trap which sends notification to farmers when the number of flies exceeds a predefined threshold. The trap is powered with solar panels, equipped with a Lynfield-inspired sticky trap that is optimized to be attractive to fruit flies and controlled by an Arduino Board to collect data and circulate the energy through the system. The fruit flies are then counted on a Raspberry Pi Board by YOLOv4-tiny and SSD-MobileNet object detection algorithms with over 95% average precision at IoU threshold of 0.5 and an alert signal is sent to the farmers based on the number of fruit flies in the trap.

Keywords—Lynfield sticky trap, Arduino, Raspberry Pi, YOLOv4-tiny, SSD-MobileNet, fruit flies

I. INTRODUCTION

Agriculture is crucial for economic growth, and increasing agricultural output is a major priority in Vietnam [1]. On the one hand, insect insecticides can disrupt agricultural metabolic processes, reducing crop production and quality [2]. Fruit flies, on the other hand, are known to cause 50 to 100 percent crop loss if prompt treatments are not performed. Only a few fruit fly species have been identified, including *Bactrocera dorsalis*, *B. correcta*, *B. cucurbitae*, *B. tau*, *B. latifrons*, *B. zonata*, *B. tuberculata*, *B. moroides*, and *B. albistriga*, with others remaining unknown. The fruit fly species that are damaging to fruits include the common fruit fly species, *B. cucurbitae* and *B. tau* [3].

To maximize crop yields, agricultural personnel often utilize a pesticide scheduler rather than considering the possibility of insect infestation in the crop [4]. As a result, not only are there a lot of pesticide residues in agricultural goods, but there's also a lot of strain on the environment [5]. Pesticides are overused in part because information on pest species and densities cannot be supplied in a timely and accurate manner. Conversely, if the information is supplied in a timely manner, it may be feasible to implement appropriate pest control methods, including the judicious use of pesticides [6, 7].

Historically, information about the environment and pest species has been obtained mostly by hand-crafted feature

extraction [6], in which employees utilize sensors to manually assess a pest's shape, color, structure, and other features with explanation from subject matter experts. Moreover, counting is time-consuming, labor-intensive, and error-prone [8]. Consequently, it is critical to create an autonomous and accurate pest identification system. In the agricultural research area, there is a rising trend toward using machine vision techniques to tackle these challenges with promising results.

Unlike similar applications in the domain, this research concentrates on the use of different kinds of real-time object detection algorithms on emerging edge computing technologies to boost system performance in terms of accuracy rate, lower power consumption, and response time reduction with the goal of detecting alive fruit flies rather than dead ones on the trap. In summary, our contributions are as follows: (1) We built modules for an end-to-end camera-equipped trap called AlertTrap, including a Lynfield-inspired adhesive trap to catch fruit flies quickly and a photovoltaic powering system controlled by a dedicated Arduino Board. A secondary Raspberry Pi Board is in charge of doing vision-based machine learning to detect and count flies, measuring environmental factors as well as delivering notifications to farmers. (2) We tested with several Lynfield-inspired trap setups to find the most appealing one to the fruit flies. (3) Finally, we put three distinct small and quick object identification deep learning models to the test: SSD-MobileNetV1, SSD-MobileNetV2, and the YOLOv4-tiny. With the results, we compare not only their capacity to identify and localize the fruit flies that we had trained them to anticipate, but also the improvement in processing speed and the power saving factor.

II. RELATED WORKS

Advances in electronic traps (e-traps) have lately made it possible to build a decentralized system that monitors fruit flies. In previous research [9–14], a widespread and effective system design was implemented by combining a large number of e-traps with computer vision and wireless communication. Such an e-trap comprises mostly of two parts: a trap mechanism and an embedded computing device. The trap, in general, is in charge of luring and trapping insects. In most cases, the trap contains an implanted device that includes a

camera, meteorological sensors, and a wireless communication module. It is in charge of collecting pictures of flies caught in traps as well as meteorological data and transferring them to a distant server. By placing e-traps at various places, area-wide field data on fruit fly pests may be obtained for statistical analysis [10,14].

An efficient and practical trap is required for observing fruit flies since the amount and species of caught flies on e-traps carry important information on their populations and distribution. There are two types of traps that are widely selected: the McPhail trap [12,15] and the yellow sticky paper [9,16]. The McPhail trap is a plastic and cylindrical piece of equipment that carries liquid attractants. Flies are trapped in the solution, and pictures of the liquid's top are sampled in order to detect them. Yellow sticky paper is often created by smoothly spreading some adhesive materials over a particular type of yellow paper that has attractants integrated into the paper or sticky materials. Fruit flies are placed on adhesive paper, and the pictures of the paper are sampled immediately to monitor the fruit flies. Prior literature has shown that when certain attractants are used, the two types of traps may efficiently capture fruit flies [9,10,12–16]. In addition to McPhail trap, Lynfield trap takes the same structural construction with the replacement of protein food lure with cuelure. Lynfield trap is analyzed to trap more male fruit flies than McPhail one [17]. Both traps are traditionally liquid-based, but we utilize a camera to photograph the inside of the trap, and condensation occurs on the camera lens. The lens seems foggy most of the time, and this effect is especially noticeable in the morning. Thus, the yellow sticky paper is preferred to avoid blurry images and Lynfield is implemented by choice.

Aside from efficiently acquiring high-quality e-trap photos, finding insects in e-trap images and distinguishing their species are the keys to pest statistics. The detection goal would be on finding and sectioning the insects from the e-trap images. In principle, insect detectors are divided into two types: hand-crafted [12,18] and feature-learning-based [8,11,14,19–20]. To recognize the *B. dorsalis* in a hand-crafted way, for example, a mixture of template matching and Kalman filters in the HSV color space was devised [18]. Generally, the detection method relies on image processing, as reported in [10,14,24–25]. While image-processing algorithms are simpler than deep learning approaches, their accuracy is reasonable (70–80 percent) and the system is linked to the lighting environment.

Kaya et al. [26] developed a machine-learning-based classifier that can distinguish between 14 butterfly species. The researchers extract the texture and color qualities. The collected features are processed using a three-layer neural network. The obtained classification accuracy is 92.85 percent.

Zhong et al. [21] developed a deep-learning-based multi-class classifier that can classify and count six distinct species of flying insects. For detection and coarse counting, the You Only Look Once (YOLO) algorithm [24] is employed. The researchers regarded the six kinds of flying insects as a

single class in order to enhance the number of training photos required by the YOLO deep learning model. To increase the size of the data set, the authors use translation, rotation, flipping, scaling, noise addition, and contrast modification on the pictures. On an insect dataset, they also used a pre-trained YOLO to fine-tune its parameters. Kalamatianos et al. [27] developed the *Dacus* Image Recognition Toolkit (DIRT). The toolkit contains Matlab code examples for quick testing as well as a library of annotated olive fruit fly images collected using McPhail traps. The authors evaluated several variants of the pre-trained Faster Region Convolutional Neural Networks (Faster-RCNN) deep learning detection approach using the DIRT dataset. RCNNs are convolutional neural networks that incorporate region proposals that indicate the regions of objects prior to categorization. The mAP of faster-RCNN was 91.52 percent, where mAP is the mean average precision for varied recall thresholds. Xia et al. detect 24 different types of insects in agricultural areas using an end-to-end deep learning neural network [23]. To extract the features, a pre-trained VGG-19 network is used. The position of the bug is then established using the Region Proposal Network (RPN). The proposed model's mAP was 89.22 percent.

Not only is the performance of an insect detection model prioritized, but so is the processing time. As a result of their performance, deep learning models are of interest to us. However, two-stage object identification models such as RCNNs are not considered since they are computationally expensive and hence unsuitable for use on edge devices. Instead, single-stage models such as YOLOv4-tiny, SSD-MobileNetv1 and SSD-MobileNetv2 are developed and evaluated across various types of computer hardware to determine the most likely method for the AlertTrap system.

The remainder is organized as follows: Section III provides an overview of the trap system, describing in detail the system elements; Section IV depicts experiments and system analysis; and Section V ends the research and outlines the prognosis after successfully deploying the AlertTrap.

III. SYSTEM OVERVIEW

AlertTrap is made up of two major components: hardware and a software framework. The following subsections go into further detail on each component of the system.

A. Hardware

Overall, the hardware, which is illustrated in Fig. 1, consists of five components: a power supply, a Lynfield trap, controllers, actuators and a container frame of the system.

1) *Power supply*: Photon energy is converted into direct current by the photovoltaic powering system, which powers the entire trap. The module is made up of a solar panel and a battery, which are responsible for generating direct current and storing it, respectively. There is also a specialized controller to manage the output from the solar panel to be stable at 9V, so that it does not cause the battery to overflow.

2) *Lynfield's trap*: Fruit flies are attracted and trapped by the mechanical trap created in the shape of a Lynfield trap.

The trap is made up of three major parts: a clear cylindrical container with four entry holes equally placed around the container's wall, a top lid that is generally color-coded to the type of attractant used, and a string used as a hanger for the attractant.

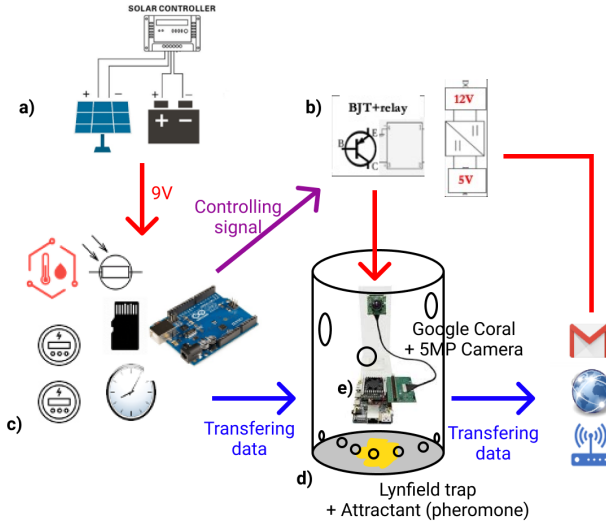


Fig. 1. Overview of the trap system consisting of a) the solar panel system, b) the actuator system, c) the controller system, d) the modified Lynfield trap and e) the system frame

3) *Controllers*: The microcontroller-controlled operation system determines when and how to activate the sensory and object detecting systems. Furthermore, it prevents Raspberry Pi from continuing to take electricity from the solar system after it has been turned off. It also regulates the voltages for running the Raspberry Pi board and camera distinctively.

4) *Actuators*: consist of a sensory module and an object detection module. The sensor system is in charge of sensing three critical factors: temperature, humidity, and light. It also keeps track of the current generated by the solar system and the voltage battery. These two criteria assist the microcontroller in determining whether or not to enable the object detection module. The object detection acquires images and uses a neural image detector to identify the fruit fly. Furthermore, it receives data from the sensor system and delivers it to the notification system then notify or warn farmers to environmental data and the amount of fruit flies.

5) *System frame*: All of the modules listed above are installed on an overall system frame, illustrated in Fig. 2, that ensures mobility and compactness.

B. Software framework

The software framework is in charge of gathering data, detecting fruit flies, and sending out alarms. The data collection and alert notification functions are only available when there is an internet connection. Otherwise, the cloud storage capability for photos and data is disabled, and notifications are provided through SMS texts. The object

detection module is built around a neural object detection method. In this study, three object detection models are examined to see which one best fits the configuration of our trap: SSD-MobileNetv1, SSD-MobileNetv2 and YOLOv4-tiny.



Fig. 2. The system frame

1) *SSD-MobileNet1*: Variants of the Single-Shot Multibox Detector (SSD) are utilized to address the real-time object detection problem in the yellow fly detection challenge. Wei Liu et al. introduced the SSD approach in [28] as a one-stage object identification method that omits the region proposal and pixel/feature resampling phases utilized in region proposal-based techniques such as Faster-RCNN. Furthermore, the network's early layers are built on a conventional image classification without classification layers, which is referred to as the base network [28]. MobileNetv1 and MobileNetv2 are utilized as foundation networks for the SSD detection models in this work. A. Howard et al. originally suggested the technique in [29]. It is based on depthwise separable convolution, which includes a depthwise convolution layer that applies a single filter per input channel and a pointwise convolution layer that generates a linear combination of the depthwise layer's output. Furthermore, to make the model more computationally efficient, a width multiplier, which is used to thin the network uniformly at each layer, and a resolution multiplier, which is applied to input images and the internal representation of each layer, were introduced as hyperparameters to tune and choose the size of the model.

2) *SSD-MobileNet2*: M. Sandler et al. originally proposed the MobileNetv2 method in [30]. Because the technique is based on MobileNetv1, it also employs the depthwise separable convolution architecture, which consists of a depthwise convolution layer and a 1x1 pointwise convolution layer. Furthermore, in order to improve the neural architecture, the technique employs linear bottleneck layers in convolutional blocks [30]. Furthermore, inverted residual

design is utilized in the model to provide shortcuts between bottlenecks in order to improve gradient propagation throughout the multiplier layers. Nonetheless, the inverted design execution resulted in greater speed and considerably higher memory efficiency in the work [30].

3) *YOLOv4-tiny*: YOLOv4 is an object detection method that evolved from the YOLOv3 model [31]. Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao [32] invented the YOLOv4 technique. With equivalent performance, it is twice as quick as EfficientDet. Furthermore, AP (Average Precision) and FPS (Frames Per Second) in YOLOv4 have risen by 10% and 12%, respectively, when compared to YOLOv3. The architecture of YOLOv4 is made up of a CSPDarknet53 backbone, a spatial pyramid pooling extra module, a PANet path-aggregation neck, and a YOLOv3 head. The compressed version of YOLOv4 is YOLOv4-tiny. It is suggested based on YOLOv4 to simplify the network topology and minimize parameters so that it may be developed on mobile and embedded devices. YOLOv4-tiny may be used for quicker training and detection. It contains just two YOLO heads as opposed to three in YOLOv4, and it was trained using 29 pre-trained convolutional layers as opposed to 137 in YOLOv4.

IV. EXPERIMENTS AND EVALUATIONS

A. Hardware

Experiments for AlertTrap hardware include testing the most likely Lynfield-inspired trap design that can attract the greatest number of fruit flies with varying trap colors, sizes and attractants.

1) *Trap color*: The purpose of this section is to assess the effectiveness of various colored traps meant to catch fruit flies. All traps are the same size and kind (Lynfield trap), and each trap has 1ml of attractant. According to the result in Fig. 3, the yellow trap attracts the most flies on the third day, as predicted; flies move forward to yellow material. However, on the 12th day, the same amount of flies were detected on each trap with a different color. They have almost all surpassed their maximum capacity of attracting insects. Hence, yellow paint is utilized to attract fruit flies outside the trap.

2) *Trap size*: The purpose of this trap is to test the impact of different trap sizes on catching. The study used two Lynfield traps: A standard-size trap that was issued by the Food and Agriculture Organization of the United Nations, International Atomic Energy Agency, Vienna, 2018 [33], with a height of 10 cm and a diameter of 7 cm, while a larger trap is twice the size of the above one. Regarding the result in Fig. 4, on the 12th day, the twice-sized Lynfield trap outperforms the conventional trap by nearly double. The size of the trap has an influence on its ability to attract fruit flies. Consequently, the larger one is employed in this investigation to catch flies.

3) *Trap attractants*: Water is the conventional method for trapping flies in a Lynfield trap. In this example, though, we employ a camera to photograph the inside of the trap, and

condensation develops on the camera lens. Most of the time, the lens seems foggy, as shown in Fig. 5, and this effect is especially noticeable in the morning. Therefore, instead of using water, or wet traps, yellow sticky paper is used to capture insects for dry traps. Hence, all training and testing in the project are done with images from dry trap setup.

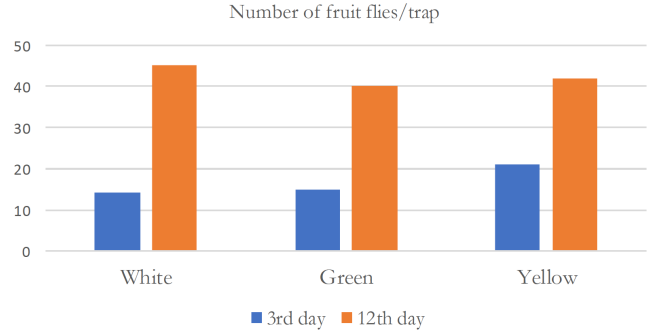


Fig. 3. Number of flies inside the white, green and yellow Lynfield trap

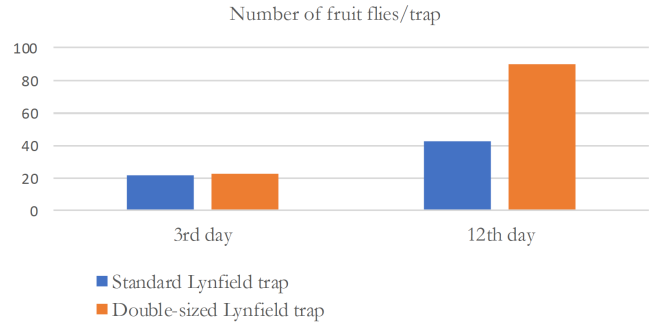


Fig. 4. Number of flies inside the standard- and double-sized Lynfield trap

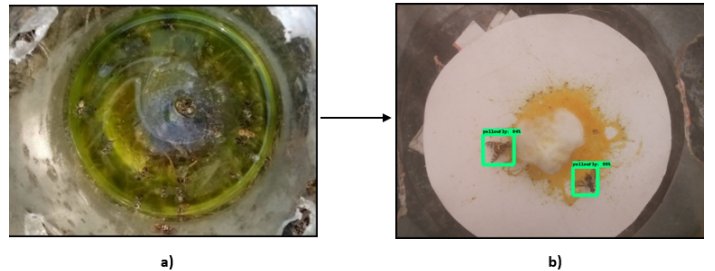


Fig. 5. The conversion from a liquid trap to a dry trap: a) the liquid trap and b) the dry trap. Images from the dry trap are used for models' training and testing. An detection example is also shown in b)

B. Software framework

Initially, the system and its detecting function were built using YOLOv3 on a Raspberry Pi 3 model B computer. Despite the fact that the system detected target fireflies in the field test using the YOLOv3 model, it took 25 seconds to complete the detection for a video frame (0.04 FPS). This creates a problem for the system's real-time application. As a result, lighter models are examined, trained, and tested in simulated settings using CPU, GPU, and TPU devices. Tesla V100-SXM2-16GB, TPU V2, and Intel(R) Xeon(R) CPU @ 2.30GHz devices are utilized for GPU, TPU, and CPU testing, respectively, with Google Colab Pro service. The models are trained and evaluated using a dataset of 248 pictures of yellow flies where 198 of them are utilized for training, while the

remaining 50 are used for testing. The project's potential models are SSD-MobileNetv1, SSD-MobileNetv2, and YOLOv4-tiny. In addition, SSD-MobileNetv1 and SSD-MobileNetv2 are also tested on the edge device, Raspberry Pi 3, to measure their processing time.

1) *Evaluation metrics*: We assess the candidate models in this experiment based on their accuracy, recall, F1 score, mean IoU, and AP at IoU thresholds of 0.25, 0.5, and 0.75. Furthermore, processing time is examined to determine the viability of real-time implementation. The research calculates accuracy, recall, and f1-score metrics by counting the amount of true positives (TP), false negatives (FN), and false positives (FP). While accuracy is defined as the ratio of true detections to total detections, recall is defined as the ratio of true detections to total ground-truths. The F1-score, on the other hand, is used to assess the overall performance of the models. Furthermore, AP is the average precision of the models with varying confidence score criteria. MeanIoU is a measure that may be used to compare the detection results' localisation to the ground-truth bounding boxes. The average processing time across all test sets is utilized for real-time feasibility inspection.

2) *Model Evaluations and Discussion*: TABLE I, TABLE II, and TABLE III demonstrate the evaluation of the candidate models based on precision, recall, F1-score, AP, and meanIoU at IoU thresholds of 0.25, 0.5, and 0.75, respectively. TABLE IV shows the results of evaluating and measuring the processing time of the candidate models.

TABLE I: PERFORMANCE OF THE SSD-MOBILENETV1, SSD-MOBILENETV2, AND YOLOV4-TINY AT IOU THRESHOLD 0.25 ON THE TEST DATASET.

Models	Precision	Recall	F1 Score	Mean IoU	AP
SSD MobileNet v1	1.0	0.924	0.960	0.702	0.983
SSD MobileNet v2	1.0	0.941	0.969	0.811	1.0
YOLOv4-tiny	1.0	1.0	1.0	0.834	1.0

In terms of performance, the YOLOv4-tiny outperforms the SSD version models in terms of recall, accuracy, F1-score, AP, and meanIoU. While the YOLOv4-tiny model outperforms SSD-MobileNetv1 with the extreme IoU threshold 0.75, due to having the different backbone, SSD-MobileNetv2 still has comparable performance to YOLOv4-tiny. Furthermore, SSD models are significantly quicker than the YOLOv4-tiny model, particularly on the GPU device run, where SSD models are 20 times faster than the YOLOv4-tiny model. Because YOLOv4-tiny is faster than YOLOv3, SSD-MobileNetv2 implementation is a viable paradigm for real-time implementation with both high processing time and accuracy.

V. CONCLUSION AND OUTLOOK

Hardware-wise, we establish through practical tests that moving from a liquid-based Lynfield trap to a dry one is a key renovation of the current technique for further statistical investigation in the software framework. Software-wise, the YOLOv4-tiny model is faster than the YOLOv3 model and has the best performance in terms of accuracy, recall, F1-score, AP, and meanIoU. However, even with an extreme localization limitation imposed by the IoU threshold, SSD MobileNetv2 still provides comparable performance. Furthermore, in terms of processing time, SSD models outperform the YOLOv4-tiny on all CPU, TPU, and notably TPU. Because of these features, SSD MobileNetv2 is the most promising choice among the three evaluated models for real-time implementation of the fruit fly detection system with high processing speed and detection accuracy.

TABLE II: PERFORMANCE OF THE SSD-MOBILENETV1, SSD-MOBILENETV2, AND YOLOV4-TINY AT IOU THRESHOLD 0.5 ON THE TEST DATASET.

Models	Precision	Recall	F1 Score	Mean IoU	AP
SSD MobileNet v1	0.982	0.907	0.943	0.707	0.957
SSD MobileNet v2	1.0	0.941	0.969	0.811	1.0
YOLOv4-tiny	1.0	1.0	1.0	0.834	1.0

TABLE III: PERFORMANCE OF THE SSD-MOBILENETV1, SSD-MOBILENETV2, AND YOLOV4-TINY AT IOU THRESHOLD 0.75 ON THE TEST DATASET.

Models	Precision	Recall	F1 Score	Mean IoU	AP
SSD MobileNet v1	0.266	0.246	0.256	0.800	0.082
SSD MobileNet v2	0.774	0.729	0.751	0.847	0.690
YOLOv4-tiny	0.847	0.847	0.847	0.857	0.802

In the short to medium term, it may be worthwhile to experiment with the TFLITE format of SSD models, which is more compatible with TPU devices than inference graphs. Converting the models to the TFLITE format would be a potential way to build even quicker detectors for real-time applications on edge devices like the Raspberry Pi and Google Coral Dev board. In order to completely automate AlertTrap, an automated yellow-sticky-paper replacement system would be designed to trigger anytime overlapping flies are spotted. This may assist to reduce overall fly counting mistakes.

TABLE IV: THE PROCESSING SPEED (FPS) OF THE CANDIDATE DETECTORS ON CPU, GPU, AND TPU HARDWARE.

Models	CPU	GPU	TPU
YOLOv4-tiny	1.282	1.207	0.5445
SSD MobileNet v1	3.141	29.155	1.025
SSD MobileNet v2	3.484	21	1.024
SSD MobileNet v1 (on Raspberry Pi 3)	0.1689	-	-
SSD MobileNet v2 (on Raspberry Pi 3)	0.1874	-	-

REFERENCES

- [1] Unnevehr, L. J. (2007). Causes of and constraints to agricultural and economic development: Discussion. *American Journal of Agricultural Economics*, 89(5), 1168-1169.
- [2] Muralidharan, K., & Pasalu, I. C. (2006). Assessments of crop losses in rice ecosystems due to stem borer damage (Lepidoptera: Pyralidae). *Crop protection*, 25(5), 409-417.
- [3] Nguyen, N. T. (1998). Fruit fly problem in South Vietnam (No. IAEA-CN--71).
- [4] Hazarika, L. K., Bhuyan, M., & Hazarika, B. N. (2009). Insect pests of tea and their management. *Annual review of entomology*, 54, 267-284.
- [5] Miller, G. T., & Spoolman, S. (2014). *Sustaining the earth*. Cengage Learning.
- [6] Cho, J., Choi, J., Qiao, M., Ji, C. W., Kim, H. Y., Uhm, K. B., & Chon, T. S. (2007). Automatic identification of whiteflies, aphids and thrips in the greenhouse based on image analysis. *Red*, 346(246), 244.
- [7] Zhang, W., & Swinton, S. M. (2009). Incorporating natural enemies in an economic threshold for dynamically optimal pest management. *Ecological Modelling*, 220(9-10), 1315-1324.
- [8] Ding, W., & Taylor, G. (2016). Automatic moth detection from trap images for pest management. *Computers and Electronics in Agriculture*, 123, 17-28.
- [9] Shaked, B.; Amore, A.; Ioannou, C.; Valdes, F.; Alorda, B.; Papanastasiou, S.; Goldshtein, E.; Shenderey, C.; Leza, M.; Pontikakos, C.; et al. Electronic traps for population and population monitoring of adult fruit flies (Diptera: Tephritidae). *J. Appl. Entomol.* 2017, 142, 43-51.
- [10] Doitsidis, L.; Fouskitakis, G.N.; Varikou, K.N.; Rigakis, I.I.; Chatzichristofis, S.A.; Papafilippaki, A.K.; Birouraki, A.E. Remote monitoring of the *Bactrocera oleae* (Gmelin) (Diptera: Tephritidae) population using an automated McPhail trap. *Comput. Electron. Agric.* 2017, 137, 69-78.
- [11] Hong, S.J.; Kim, S.Y.; Kim, E.; Lee, C.H.; Lee, J.S.; Lee, D.S.; Kim, G. Moth Detection from Pheromone Trap Images Using Deep Learning Object Detectors. *Agriculture* 2020, 10, 170.
- [12] Wang, J.; Chen, X.; Hou, X.; Zhou, L.; Zhu, C.; Ji, L. Construction, implementation and testing of an image identification system using computer vision methods for fruit flies with economic importance (Diptera: Tephritidae). *J. Appl. Entomol.* 2016, 73, 1511-1528.
- [13] Alorda, B.; Valdes, F.; Mas, B.; Leza, M.; Almenar, L.; Feliu, J.; Ruiz, M.; Miranda, M.; Design of an energy efficient and low cost trap for Olive fly monitoring using a ZigBee based Wireless Sensor Network. In *Proceedings of the 10th European Conference on Precision Agriculture (ECPA)*, Rishon LeTsiyon, Israel, 12-16 July 2015; pp. 1-23.
- [14] Philimis, P.; Psimolophitis, E.; Hadjiyiannis, S.; Giusti, A.; Perello, J.; Serrat, A.; Avila, P. A centralised remote data collection system using automated traps for managing and controlling the population of the Mediterranean (*Ceratitis capitata*) and olive (*Dacus oleae*) fruit flies. In *Proceedings of the International Conference on Remote Sensing and Geoinformation of the Environment (RSGY)*, Paphos, Coral Bay Cyprus, 5 August 2013; pp. 8-16.
- [15] George, S.C. History and use of the McPhail trap. *Florida Entomol.* 1977, 60, 11-16. 11.
- [16] Hall, D.G. An assessment of yellow sticky card traps as indicators of the abundance of adult *Diaphorina citri* (Hemiptera: Psyllidae) in citrus. *J. Econ. Entomol.* 2009, 10, 446-452.
- [17] Dominiak, B. C., & Nicol, H. I. (2010). Field performance of Lynfield and McPhail traps for monitoring male and female sterile *Bactrocera tryoni* (Froggatt) and wild *Dacus newmani* (Perkins). *Pest management science*, 66(7), 741-744.
- [18] Xiao, D.; Yang, Q.; Fu, J.; Deng, X.; Feng, J.; Ye, Y.; Lu, Y. A multi-target trapping and tracking algorithm for *Bactrocera Dorsalis* based on cost model. *Comput. Electron. Agric.* 2016, 123, 224-231.
- [19] Kalamatianos, R.; Karydis, I.; Doukakis, D.; Avlonitis, M. DIRT: The *Dacus* Image Recognition Toolkit. *J. Imaging* 2018, 4, 129.
- [20] Sun, Y.; Liu, X.; Yuan, M.; Ren, L.; Wang, J.; Chen, Z. Automatic in-trap pest detection using deep learning for pheromone-based *Dendroctonus valens* monitoring. *Biosyst. Eng.* 2018, 176, 140-150.
- [21] Y. Zhong, J. Gao, Q. Lei and Y. Zhou, "A vision-based counting and recognition system for flying insects in intelligent agriculture", *Sensors*, vol. 18, no. 5, pp. 1489, May 2018.
- [22] R. Kalamatianos, I. Karydis, D. Doukakis and M. Avlonitis, "DIRT: The *dacus* image recognition toolkit", *J. Imag.*, vol. 4, no. 11, pp. 129, Oct. 2018.
- [23] D. Xia, P. Chen, B. Wang, J. Zhang and C. Xie, "Insect detection and classification based on an improved convolutional neural network", *Sensors*, vol. 18, no. 12, pp. 4169, Nov. 2018.
- [24] P. Tirelli, N. A. Borghese, F. Pedersini, G. Galassi and R. Oberti, "Automatic monitoring of pest insects traps by Zigbee-based wireless networking of image sensors", *Proc. IEEE Int. Instrum. Meas. Technol. Conf.*, pp. 1-5, May 2011.
- [25] C. Sun, P. Flemons, Y. Gao, D. Wang, N. Fisher and J. La Salle, "Automated image analysis on insect soups", *Proc. Int. Conf. Digit. Image Comput. Techn. Appl. (DICTA)*, pp. 1-6, Nov. 2016.
- [26] Y. Kaya and L. Kayci, "Application of artificial neural network for automatic detection of butterfly species using color and texture features", *Vis. Comput.*, vol. 30, no. 1, pp. 71-79, Jan. 2014.
- [27] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You only look once: Unified real-time object detection", *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 779-788, Jun. 2016.
- [28] W. Liu et al., "SSD: Single Shot MultiBox Detector", *Computer Vision - ECCV 2016*, pp. 21-37, 2016. Available: 10.1007/978-3-319-46448-0_2.
- [29] A. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications", *arXiv.org*, 2017. [Online]. Available: <https://arxiv.org/abs/1704.04861>.
- [30] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510-4520, doi: 10.1109/CVPR.2018.00474.
- [31] Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.
- [32] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.
- [33] (IAEA) International Atomic Energy Agency. (2003). Trapping guidelines for area-wide fruit fly programmes. *International Atomic Energy Agency, Vienna, Austria*, 47.