Research Progress in Mechanical and Manufacturing Engineering Vol. 2 No. 2 (2021) 1023-1034

RPMME

Homepage: http://publisher.uthm.edu.my/periodicals/index.php/rpmme
e-ISSN : 2773-4765

A Pest Monitoring System for Agriculture Using Deep Learning

Syahir Afiq Mohd Faisal[1], Ong Pauline[1*]

[1]Faculty of Mechanical Engineering and Manufacturing Engineering, Universiti Tun Hussein Onn Malaysia, 86400, Johor, MALAYSIA

*Corresponding Author Designation

DOI: https://doi.org/10.30880/rpmme.2021.02.02.112
Received 31 July 2021; Accepted 04 Dec. 2021; Available online 25 December 2021

Abstract: The study developed a pest monitoring system for agriculture using deep learning, pi camera and environmental sensors. The developed system can continuously monitor and calculate the number of pest insects stick on the yellow sticky papers. The detected pest insects were counted using image processing and deep learning model, specifically, the You Only Look Once (YOLO), with an average accuracy of 52.3% and computation time of 8-10 minutes per picture. The environmental information of temperature and humidity were also gathered in the study site, where fruit plants were grown as the main crop. The tests revealed that humidity has the strongest correlation with the number of pest insects. As conclusion, the developed system can effectively collect the insect counts automatically, which provides useful information for efficient pest control in crop cultivation processes.

Keywords: Deep learning, You Only Look Once (YOLO), Agriculture

1. Introduction

Pest management is an important aspect of effective crop management. Pests are unwanted organisms that harm plants in some way. Pest activity and density should be closely monitored for effective pest control [1]. The sticky paper trap is one of the simplest and cheapest ways to monitor pest insect populations in agriculture [2]. A sticky paper trap attracts insects by sticking a sheet of sticky paper to coloured cardboard. Inspection of the sticky paper traps may reveal information about insect density and species.

Counting small insects on sticky paper in controlled lighting conditions takes time and is subject to human error [3]. The image processing technique can be used to simplify the process of counting pests in this regard. Temperature, relative humidity, and light intensity all affect insect activity in the greenhouse [4]. Using artificial light sources, whiteflies can be easily trapped [5].

However, processing the images of sticky papers manually is tedious for large plantations, requiring constant monitoring of insect pest populations and environmental conditions [6]. The frequency of insect pest outbreaks can be predicted and prevented by integrating low-cost embedded systems, such as wireless imaging nodes, with the Internet of Things (IoT) [7]. Wireless images systems with multiple

*Corresponding author: ongp@uthm.edu.my

sensor nodes can be created by these systems use image processing to count insects on sticky paper trap images and collect environmental data.

Pest monitoring is important in agriculture to keep plants safe. Pests damage plants by eating and nesting in plant parts. Its existence is not always obvious. A manual system is used by most farmers. Human labour is required to manually check temperature, humidity, and light intensity in the greenhouse at specific times. Manual practice, it seems, takes a lot of time, effort, and dedication, 24-hour human attention is required to monitor the health of essential plants like vegetables and flowers. This study develops a pest monitoring system using image processing and deep learning. Image processing is used to automatically detect and count insects on sticky paper traps. This developed system could help crop management by collecting quantitative pest and environmental data. This method eliminates the need for tedious manual counting and allows for rapid evaluation of insect pests and environmental details. Pest monitoring requires fewer workers than manual inspection. By reducing pest-related crop losses and improving low-risk food production, environmental factors can reduce the risk of pest-related crop losses and improve low-risk food production.

2. Materials and Methods

The materials and methods section, otherwise known as methodology, presents the methods used to conduct the study. This includes the explanation on the development of the insect pest monitoring device, data collection and insect pest identification.

2.1 Research Methodology

This presents the flowchart of the pest monitoring system. First, assemble of the insect pest monitoring device. Then, data collection and remote server of the monitoring. After that, image processing method being used. Insect pest detection using the deep learning model of You Only Look Once (YOLO) and multiple linear regression (MLR). Study the relationship between environment and number of pests.

2.2 Development of the Insect Pest Monitoring Device

An insect pest monitoring system's as shown in Figure 1 include the hardware and software are divided into two categories. For example, a sticky paper trap as part of the insect pest monitoring system's hardware module. The box contains the Raspberry Pi 4b and an external power supply. Place the device in insect nesting areas and connect it to a Wi-Fi network to collect data. The software module's code includes functions for image capture, image processing, pest insect identification and counting.

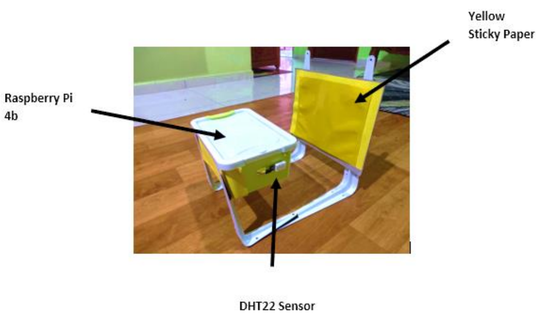

Figure 1: Pest Monitoring Device

2.3 Data Collection

This system collects data using a camera and multiple environmental sensors. Each observation period will be divided into two 7-day sections. 8 cm in front of an A4 sticky paper trap (21.0 cmx 29.7 cm). Photo taken with daylight white balance for automatic colour correction. Every hour from 12:00 to 7:00 p.m., photos are taken. Night images are avoided as they may influence insect pests' decisions and actions. Since most insects are more active during the day, there appears to be little increase in insect pest numbers at night. For data collection using the DHT22 sensor, the temperature and humidity sensors are sent every 10 minutes from 12:00 PM to 7:00 PM for 7 days.

2.4 Remote Server

The RaspController is used to monitor the pest insect, temperature, and humidity in real-time. RaspController is a software programme that allows remote management of the Raspberry Pi. Features like managing files, using GPIO ports, sending commands via the terminal, viewing pictures from a connected camera, and reading data from sensors have been added. Finally, wiring diagrams, pinouts, and other information ensure proper use of the Raspberry Pi. This app is available on the Google Play Store and the Apple AppStore.

2.5 Image Processing

The internet is used to send the captured images and information to a distant location. The images are first scaled to 32 x 32 pixels for use with YOLO insect pest detection software. After the resized images have been manually annotated, the images are divided into two groups: the training set and the testing set. The MATLAB toolbox was used to resize and label the images. The acquired dataset will then be used to identify insect pests using the YOLO algorithm.

2.6 Insect Pest Detection Using YOLO

The insect pest counting, and identification system was created using YOLO. The steps are as follows: pest insect identification, coarse counting of pest insects, feature extraction and pest detection using YOLOv2 algorithm.

2.7 Performance Analysis

The performance of coarse counting and fine counting using the YOLO is evaluated using the following derived indices [4]:

$$Precision = \frac{TI}{TI + FI} \qquad \text{Eq.1}$$

$$Recall = \frac{TI}{TI + FNI} \qquad \text{Eq.2}$$

$$F_1 = 2\,x\,\frac{Precision * \ Recall}{Precision + Recall} \qquad \text{Eq.3}$$

Where,

$TI = true\ insect$

$TNI = true\ non - insect$

$FI = false\ insect$

$$FNI = false\ non - insect$$

3. Results and Discussion

The experiment's results are presented here. Plants were used to test the insect capture method on yellow sticky paper. We compare YOLO-based pest detection and precision counting to component labelling. Then comes counting performance. YOLO models are created using machine learning and MATLAB 2021a. Find the prediction model using Multi Linear Regression between actual and forecast pest count related to environmental parameters.

3.1 Parameters determined from pest count using Multiple Linear Regression (MLR) method

The data for this experiment was collected between 12 p.m. and 7 p.m. on seven days from June 8th to 14th, 2021. The data were chosen to identify insect pests based on temperature and humidity. A manual inspection of sticky paper trap images was performed to further analyse the system's data. Table 1 shows the results of the manual inspection. The data for pest count and environmental parameters were collected on 14/6/2021 between 12 and 7 p.m. to calculate the MSE between the actual and predicted pest count using the MLR model.

Table 1: Pest count and environment sensor data for time (12 – 7 PM) on 14/6/2021

Date	Time	Temperature	Humidity	No of Pest
14/6/2021	12:00	33.1	70.10%	34
14/6/2021	13:00	31.2	70.10%	36
14/6/2021	14:00	31.5	70.20%	39
14/6/2021	15:00	31.8	71.00%	42
14/6/2021	16:00	30.7	72.40%	45
14/6/2021	17:00	31.5	75.70%	49
14/6/2021	18:00	30.8	75.60%	50
14/6/2021	19:00	30.6	77.20%	50

SUMMARY OUTPUT								
Regression Statistics								
Multiple R	0.953438							
R Square	0.909043							
Adjusted R Square	0.87266							
Standard Error	2.276459							
Observations	8							
ANOVA								
	df	*SS*	*MS*	*F*	*Significance F*			
Regression	2	258.9637	129.4818	24.98556	0.002495			
Residual	5	25.91134	5.182268					
Total	7	284.875						
	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
Intercept	-24.0833	60.00721	-0.40134	0.704747	-178.337	130.1701	-178.337	130.1701
Temperature	-1.86837	1.302701	-1.43422	0.210967	-5.21707	1.480334	-5.21707	1.480334
Humidity	172.935	35.96578	4.808319	0.004848	80.48197	265.388	80.48197	265.388

Figure 2: Summary of multiple linear regression model of training data by time

From Figure 2, the relationship between the number of pest and the input variables of temperature and humidity can be modelled as:

$$Number\ of\ pest = \qquad \text{(Eq.4)}$$

$$-24.0833 - 1.86837(Temperature) + 172.935(Humidity)$$

Table 2: Multiple Linear Regression model of training data by time

Date	Time	Actual	Forecast	Error	Error square
14/6/2021	12:00	34	35.30	-1.3	1.69
14/6/2021	13:00	36	38.85	-2.85	8.12
14/6/2021	14:00	39	38.46	0.54	0.29
14/6/2021	15:00	42	39.29	2.71	7.34
14/6/2021	16:00	45	43.76	1.24	1.54
14/6/2021	17:00	49	47.97	1.03	1.06
14/6/2021	18:00	50	49.11	0.89	0.79
14/6/2021	19:00	50	52.25	-2.25	5.06

Table 2 shows the predicted pest numbers using the prediction model in Eq.4 for the data in Table 1. The MSE was 3.24. On June 12, 2021, 12-7pm, the collected pest count and environment sensor data were used as testing data and fitted into the prediction model in Eq.4. Table 2 summarises the prediction values with an MSE of 1.18.

Table 3: Pest count and Environment sensor data for time (12 – 7 PM) on 12/6/2021

Date	Time	Temperature	Humidity	No of Pest
12/6/2021	12:00	32.9	68.80%	27
12/6/2021	13:00	33.1	67.90%	27
12/6/2021	14:00	33.1	66.20%	27
12/6/2021	15:00	33.5	66.20%	28
12/6/2021	16:00	33.8	67.60%	29
12/6/2021	17:00	33.4	65.30%	29
12/6/2021	18:00	33.6	64.80%	29
12/6/2021	19:00	32.5	68.90%	30

The results in Tables 1 and Table 3 are for data collected at different times on the same day. Using the data in Table 4, a prediction model for data collected on multiple days was developed. The data for pest count and environmental parameters were chosen from 7 PM on 8/6/2021 to 12/6/2021 to calculate the MSE between actual and forecasted pest count using the MLR model.

Table 4: Pest count and Environment sensor data for time 7 PM on (8/6/2021 to 12/6/2021)

Date	Time	Temperature	Humidity	No of Pest
8/6/2021	19:00	32.9	64.20%	6
9/6/2021	19:00	32.9	64.50%	9
10/6/2021	19:00	34.5	67.00%	20
11/6/2021	19:00	32.5	70.00%	27
12/6/2021	19:00	32.5	68.90%	30
13/6/2021	19:00	31.5	68.10%	33

14/6/2021	19:00	30.6	77.20%	50

SUMMARY OUTPUT								
Regression Statistics								
Multiple R	0.945944							
R Square	0.89481							
Adjusted R Square	0.842215							
Standard Error	5.980353							
Observations	7							
ANOVA								
	df	*SS*	*MS*	*F*	*Significance F*			
Regression	2	1216.942	608.4708	17.01321	0.011065			
Residual	4	143.0585	35.76462					
Total	6	1360						
	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
Intercept	-128.664	134.2181	-0.95862	0.392037	-501.313	243.9854	-501.313	243.9854
Temperature	-1.49222	2.804464	-0.53209	0.622846	-9.27866	6.29422	-9.27866	6.29422
Humidity	294.8483	78.15111	3.772797	0.019557	77.86603	511.8306	77.86603	511.8306

Figure 3: Summary of multiple linear regression model of training data by date

$$Forecast\ no\ of\ pest = \qquad \text{(Eq.5)}$$

$$-128.664 - 1.49222(Temperature) + 294.8483(Humidity)$$

Table 5: Multiple Linear Regression model of training data by date

Date	Time	Actual	Forecast	Error	Error square
8/6/2021	19:00	6	11.53	-5.53	30.58
9/6/2021	19:00	9	12.42	-3.42	11.69
10/6/2021	19:00	20	17.40	2.6	6.76
11/6/2021	19:00	27	29.23	-2.23	4.97
12/6/2021	19:00	30	25.98	4.02	16.16
13/6/2021	19:00	33	25.12	7.88	62.09
14/6/2021	19:00	50	53.29	-3.29	10.82

Table 5 shows the predicted results using the data in Table 4, with an MSE of 20.44. Then, the data from 3 PM on 8/6/2021 to 12/6/2021 were chosen as testing data (Table 6) and fitted into the prediction in Equation Eq.5, yielding an MSE of 38.94.

Table 6: Pest count and Environment sensor data for time 3 PM on (8/6/2021 to 12/6/2021)

Date	Time	Temperature	Humidity	No of Pest
8/6/2021	15:00	33.8	60.40%	2
9/6/2021	15:00	34.9	62.80%	8
10/6/2021	15:00	34.8	61.70%	14
11/6/2021	15:00	34.6	62.80%	23
12/6/2021	15:00	33.5	66.20%	28
13/6/2021	15:00	34	63.80%	31
14/6/2021	15:00	31.8	71.00%	42

The MLR model had an adjusted R-squared of 0.8727 (Figure 2) and 0.8422 (Figure 3), indicating good predictive performance. As shown in Eq.4 and 5, increasing humidity increases pest numbers.

3.2 Analysis of the environmental parameters with insect pest count

The time for the number of pests, the value of humidity, and the temperature have been simplified from 10 minutes to 1 hour, as shown in Table 7. The result for this analysis uses the pest count and environmental data for time (12 -7 PM) on 14/6/2021 and time at 7 PM on (8/6/2021 to 14/6/2021) shown in Table 4.

Table 7: Pest count and Environment sensor data for time (12 -7 PM) on 14/6/2021

Date	Time	Temperature	Humidity	No of Pest
14/6/2021	12:00	33.1	70.10%	34
14/6/2021	13:00	31.2	70.10%	36
14/6/2021	14:00	31.5	70.20%	39
14/6/2021	15:00	31.8	71.00%	42
14/6/2021	16:00	30.7	72.40%	45
14/6/2021	17:00	31.5	75.70%	49
14/6/2021	18:00	30.8	75.60%	50
14/6/2021	19:00	30.6	77.20%	50

The two samples had a normalised change rate (Δ) in insect count, temperature, and humidity. The time-measured data from the sensors could be combined to get data on changes in each environmental parameter. The data show a correlation between environmental variations and insect count, as shown in Figures 4 and 5.

Table 8: Statistical analysis of the environmental parameters (8/6/2021 – 14/6/2021)

Parameter	Parameter mean		Range		Correlation with insect pest count		
	By Time	By Date	By Time	By Date	By Time	By Date	Mean
Temperature (^{o}C)	31.4	32.4	33.1 - 30.6	64.2 - 77.2	-0.69	-0.72	-0.7
Humidity (%)	72.7	68.6	70.1 - 77.2	64.2 - 77.2	0.93	0.94	0.93

Tables 8 show significant differences in average temperature and humidity between time periods by time and date. As shown in Figures 4 and 5, the temperature and humidity of the time period data change in real-time.

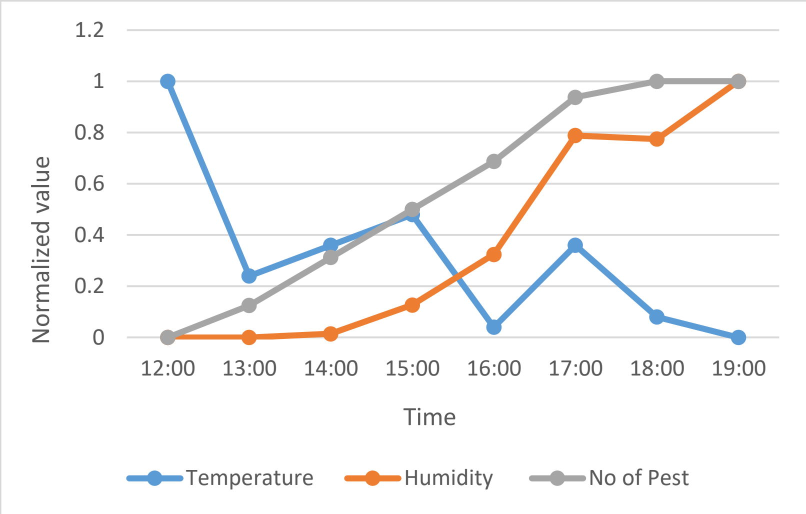

Figure 4: Environmental parameter and insect pest count normalized data by time plot

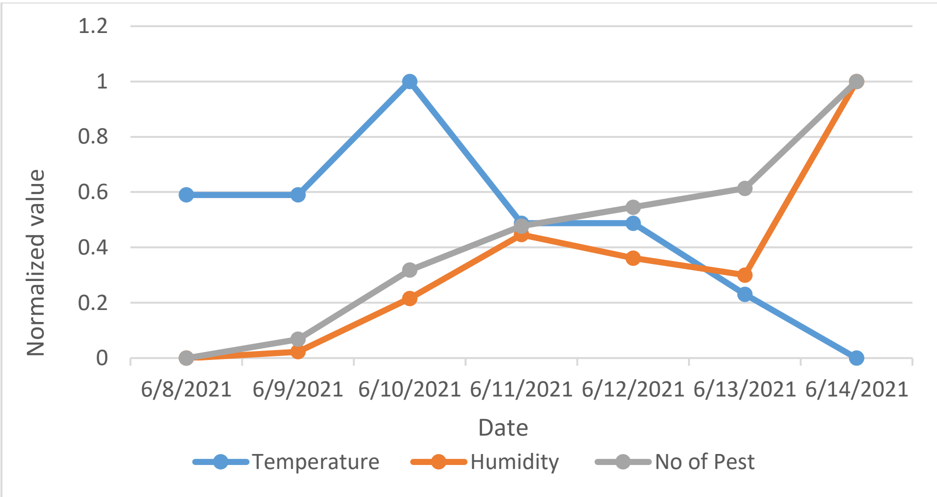

Figure 5: Environmental parameter and insect pest count normalized data by date plot

The first phase has seen a significant difference in humidity and temperature. As shown in Table 8, this event caused an increase in insect pest numbers. Similarly, between 11/6 and 14/6, humidity increased, increasing insect pests. The method can reveal links between insect pest counts and environmental conditions. As a result, more observations of their relationships will be made, with the goal of providing a more detailed analysis of the observed patterns.

3.3 Performance of Training loss function YOLO v2

YOLOv2 are two current object detection methods. In this case, Yolov2ObjectDetector is used to find pests. The addition of scale detection to the YOLO version2 object detector helps identify small objects. The YOLO v2 can detect pests in crops. To begin the YOLO segmentation process, it must first create a ground truth of information. Image Labeler: Create the bounding box for the pest in a picture. Construct bounding boxes for all images currently stored in the database. After constructing the bounding boxes, we will load the network layers. Set the classifier, minimum batch size, starting learns rate, maximum epochs and checkpoint route parameters. The network has been trained for x iterations.

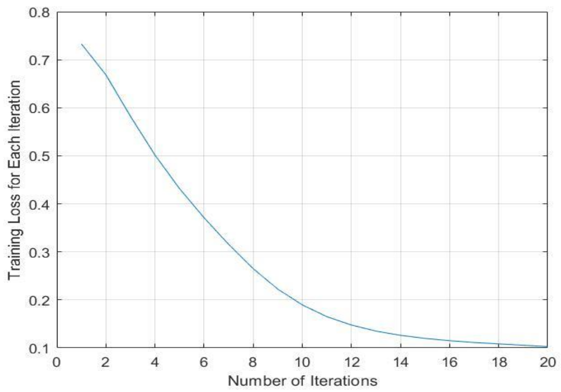

Figure 6: Loss function in YOLO V2 with ResNet-50

```
     TrainingLoss: [0.7330 0.6685 0.5822 0.5014 0.4315 0.3710 0.3156 0.2650
0.2225 0.1896 0.1651 0.1477 0.1351 0.1262 0.1198 0.1150 0.1113 0.1084
0.1056 0.1028]

     TrainingRMSE: [0.8561 0.8176 0.7630 0.7081 0.6569 0.6091 0.5618 0.5147
0.4717 0.4354 0.4063 0.3843 0.3675 0.3553 0.3462 0.3392 0.3336 0.3293
0.3249 0.3206]

    BaseLearnRate: [0.0010 0.0010 0.0010 0.0010 0.0010 0.0010 0.0010 0.0010
0.0010 0.0010 0.0010 0.0010 0.0010 0.0010 0.0010 0.0010 0.0010 0.0010
0.0010 0.0010]
```

Figure 7: Data of the graph loss function in YOLO v2

As shown in Figure 6, the average network loss decreases as the number of iterations increases. The network's performance improved as the prediction error decreased. After 20 iterations, the average loss of improved YOLO v2 was 0.1028 and did not drop further, indicating that training was complete.

3.4 Accuracy and Inference time of YOLO v2

The chosen train sets from the 90% standard set. It does not reduce train images, but rather sends them to the YOLO neural network, which produces the standard model (Ms). Then we use the remaining 10% of the original data to test the standard model. Finally, we perform a single picture degradation operation on the test sets before adding them to the standard model for processing (Ms). Table 4-8 shows the results of the standard model's deteriorated test sets.

Table 9: Evaluation of training and testing dataset by YOLO v2 algorithm

Evaluation	**Training data**	**Testing data**
Number of anchors	9	
Recall	0.053	0.0333
Precision	1	1
Average IoU (%)	0.8518	0.9249
mAP (%)	5	3

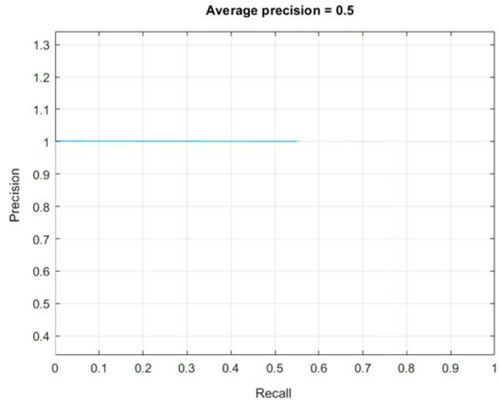

Figure 8: Precision-Recall (PR) curves of training data

	Boxes	Scores	Labels
1	*4×4 double*	[0.5349;0.5190;0.5067;0.5312]	*4×1 categor…*
2	*5×4 double*	[0.5385;0.5035;0.5171;0.5347;0.5267]	*5×1 categor…*
3	[35,32,50,4…	[0.5798;0.5009]	*2×1 categor…*
4	*4×4 double*	[0.5624;0.5030;0.5744;0.5313]	*4×1 categor…*
5	*4×4 double*	[0.5039;0.5011;0.5088;0.5108]	*4×1 categor…*
6	*3×4 double*	[0.5209;0.5628;0.5163]	*3×1 categor…*
7	[36,27,46,51]	0.5060	pest
8	[22,16,59,4…	[0.5142;0.5023]	*2×1 categor…*
9	*5×4 double*	[0.5432;0.5071;0.5671;0.5087;0.5229]	*5×1 categor…*

Figure 9: Accuracy of the pest detection of training data using YOLO v2

Table 10: Summary of pest counting of training data using bounding box detection

Training data	Overall accuracy	Precision	Recall
YOLO v2	52.3	1	0.54

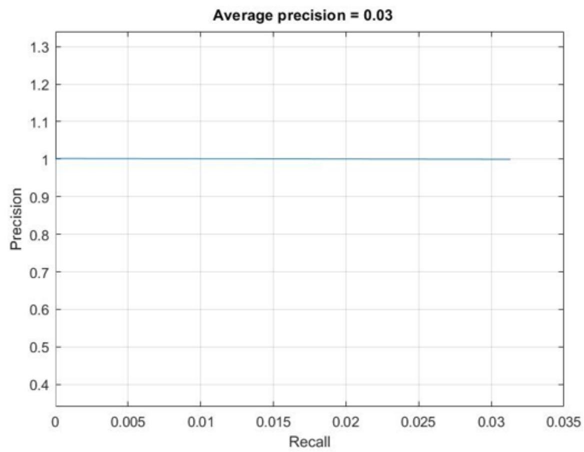

Figure 10: Precision-Recall (PR) curves of testing data

	Boxes	Scores	Labels
1	[36,27,46,51]	0.5060	pest

Figure 11: Accuracy of the pest detection of testing data using YOLO v2

Table 11: Summary of pest counting of testing data using bounding box detection

Testing data	Overall accuracy	Precision	Recall
YOLO v2	0.506	1	0.5

Figures 9 and 10 show the results of detecting the training and testing sets of pest images. Table 10 shows that while 52.3% of insects in the training database are correctly classified, 23 out of 50 instances in the ground truth are incorrectly classified. In the testing data, 50.6 percent of insects are detected (Table 11), but only 15 of 30 in the ground truth. The network found 15 insects not shown in the reference images. The weighted classification accuracy is also computed as a function of TP and total insect count. A lack of hand-drawn labels resulted in rejects due to reduced crossings over unions, but this study revealed that the quality of the ground truth needed to be improved. However, the total number of insects on the traps was manually counted.

3.5 Comparison of training and testing data using YOLO v2

Table 12: Comparison of two machine learning method to find the accuracy of pest detection

YOLO v2	Accuracy
Training data	52.3
Testing data	50.6

Table 12 compares the accuracy of YOLO v2 insect pest detection between training and testing data. But comparing the table result to our suggested method is unfair. The results cannot be compared due to the different frameworks, datasets, and training settings. As shown in the table, the training data has better accuracy than the testing data for YOLO v2 object detection.

4. Conclusion

Insect pest monitoring using an integrated imaging and environmental sensor network was presented. This study surpasses the use of only wireless cameras by combining them with environmental sensors to gather more potentially useful information for various insect pest monitoring applications. This study's major contribution is the creation of an automated, integrated system for collecting insect pests and environmental data for plant growth. This method eliminates the need for tedious manual counting, allowing for rapid evaluation of insect pests and environmental data. In an uncontrolled agricultural environment, it is possible to study the impact of environmental conditions on insect pest presence and activity statistically.

This study proposed an IoT and machine learning approach for remote pest monitoring and automated insect identification. The YOLO v2 object detection automatically identified the insect. This system uses a four-layer Internet of Things architecture to remotely trap insects. This study used various insect image datasets to assess a machine learning technique for insect identification.

Acknowledgement

The authors would like to thank the Faculty of Mechanical and Manufacturing Engineering, Universiti Tun Hussein Onn Malaysia for the support in implementing the research.

References

[1] Vijayalakshmi, B., Ramkumar, C., Niveda, S., & Pandian, S. C. (2019). Smart Pest Control System in Agriculture. *IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing, INCOS 2019*, 1–4. https://doi.org/10.1109/INCOS45849.2019.8951351

[2] Bashir, M. A., Alvi, A. M., & Naz, H. (2014). Effectiveness of sticky traps in monitoring of insects . Short Communication. *Journal of Environmental and Agricultural Sciences*, *1*(5), 2–4

[3] Rustia, D. J. A., & Lin, T. Te. (2017a). An IoT-based wireless imaging and sensor node system for remote greenhouse pest monitoring. *Chemical Engineering Transactions, 58,* 601–606. https://doi.org/10.3303/CET1758101

[4] Rustia, D. J. A., Lin, C. E., Chung, J. Y., Zhuang, Y. J., Hsu, J. C., & Lin, T. Te. (2020). Application of an image and environmental sensor network for automated greenhouse insect pest monitoring. *Journal of Asia-Pacific Entomology*, *23*(1), 17–28. https://doi.org/10.1016/j.aspen.2019.11.006

[5] Shimoda, M., & Honda, K. ichiro. (2013). Insect reactions to light and its applications to pest management. *Applied Entomology and Zoology*, *48*(4), 413–421. https://doi.org/10.1007/s13355-013-0219-x

[6] Rustia, D. J. A., & Lin, T. Te. (2017). An IoT-based wireless imaging and sensor node system for remote greenhouse pest monitoring. *Chemical Engineering Transactions*, *58*(January), 601–606. https://doi.org/10.3303/CET1758101

[7] Varandas, L., Faria, J., Gaspar, P. D., & Aguiar, M. L. (2020). Low-cost IoT remote sensor mesh for large-scale orchard monitorization. *Journal of Sensor and Actuator Networks*, *9*(3). https://doi.org/10.3390/JSAN9030044