

國立臺灣大學生農學院生物機電工程學系



博士論文

Department of Biomechatronics Engineering

College of Bio-Resources and Agriculture

National Taiwan University

Doctoral Dissertation

應用無線影像環境感測網路於智慧整合蟲害管理

Intelligent and Integrated Pest Management Using Wireless
Imaging and Environmental Sensor Networks

羅傑瑞

Dan Jeric Arcega Rustia

指導教授：林達德 博士

Advisor: Ta-Te Lin, Ph.D.

中華民國 110 年 02 月

February 2021

國立臺灣大學博士學位論文

口試委員會審定書



應用無線影像環境感測網路於智慧整合蟲害管理

Intelligent and Integrated Pest Management Using Wireless
Imaging and Environmental Sensor Networks

本論文係羅傑瑞君 (D05631006) 在國立臺灣大學生物機電工程學系、所完成之碩博士學位論文，於民國 110 年 01 月 20 日承下列考試委員審查通過及口試及格，特此證明

口試委員：

林達德

(簽名)

(指導教授)

邱真志

謝廣文

鄧彥甫

許如名

系主任、所長

陳財弘

(簽名)

Acknowledgements



I am a firm believer that research and learning have no end. But I believe too that I should continue learning on a new path and grow as a person. With heartfelt gratitude, I would like to acknowledge all the people that supported me.

I would like to express my utmost gratitude to my advisor, Prof. Ta-Te Lin, for sharing this joy and excitement of doing research and giving ample contributions to the society. I am very thankful for all the fruitful discussions and ideas that we have shared. I would have not learned a lot of valuable things if it was not for him.

I would like to thank my lab mates that supported this research: 維哲, 麟雅, and 乙澤, and all its hard-working research assistants: Cynthia, Giles, Steven, and Sam. I would also like to extend my gratitude to the members of the Tainan District Agricultural Research and Extension Station. You made me realize that research can never be complete without partners working together to meet a common goal.

I would like to thank my fellow expats here in Taiwan, Sandra and Ngo Nha, for being very supportive and helpful during my studies.

I would like to thank my family, Jamie, Janine, Miko, and my friends back in the Philippines. I am very grateful for the trust that you all have given me as you gave me the strength to work hard. All of your support and encouragement are priceless to me.


Most importantly, thank you God for everything. You are amazing, indeed.

“Consider it pure joy, whenever you face trials of many kinds, because you know that the testing of your faith produces perseverance.” James 1:2-4

摘要



為了推動農民使採用智慧化的蟲害整合管理 (IPM)，本研究開發了一套可以自動監測農業生產場域中害蟲數量及環境相關資訊的智慧整合系統。本系統由無線感測器裝置所組成，這些裝置會將黏蟲紙的影像傳送至遠端伺服器進行分析並回傳訊息給農民作為蟲害整合管理之應用。本研究同時開發了用於黏蟲紙影像上自動偵測及辨識害蟲之演算法，該演算法利用卷積神經網路 (CNN) 深度學習模型以級聯的方式進行害蟲偵測和分類；分別是由一個負責從黏蟲紙影像上定位出物件的物件偵測器模型及一個用於辨識所偵測到的物件種類的分類模型所組成。深度學習模型根據安裝場域的特性分別以兩種不同的深度學習方法來進行訓練。對於溫室場域應用了半監督式學習的技術，利用系統新擷取的黏蟲紙影像自動收集害蟲訓練影像並重新訓練害蟲分類模型。經過一年的持續訓練，分類模型基於物件層次和影像層次的測試 F1 分數都可達到 0.93。至於戶外場域則是對多個分類模型進行訓練並根據分類學進行級聯，將害蟲影像分類到物種級別，此級聯演算法基於物件層次和影像層次測試的平均 F1 分數分別為 0.91 和 0.89。本系統已安裝在多個農業生產場域中驗證與測試並提供農場管理者進行評估與應用於蟲害整合管理。本



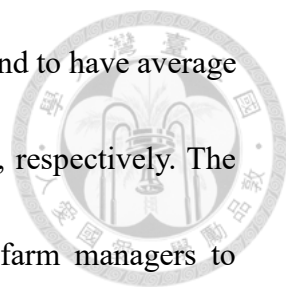
研究並透過數據分析從長期資料中獲取相關資訊開發了警報模型及生物學模型，將蟲害數量數據整理轉換為有意義的資訊，例如可以提供蟲害整合管理行動建議的警報級別，以幫助農場管理者進行決策。在對所有實驗場域的測試中，利用所訓練的生物學模型來描述害蟲的飛行頻率行為，平均 r^2 值可以達到 0.97。這些整理過的資訊是通過網站和行動 APP 與農場管理者共享以便有效應對蟲害狀況。根據農場管理者的使用情況及回饋意見顯示，本系統能夠透過量化的參考依據來協助擬定蟲害整合管理策略，而自動化系統能夠讓研究人員及專家引導農民進入數據驅動和友善環境的蟲害管理。

關鍵詞：整合蟲害管理、黏蟲紙、影像辨識、深度學習、無線感測網路、數據分析

Abstract



To drive farmers into smarter integrated pest management (IPM), an integrated and intelligent system that can automatically monitor the number of insect pests and measure environmental conditions in agricultural production sites was developed. The proposed system is composed of wireless sensor nodes that send images of sticky paper traps to a remote server. An algorithm was developed to automatically detect and recognize insect pests from the sticky paper trap images. The algorithm features a cascaded approach for detection and classification using convolutional neural network (CNN) deep learning models. It is composed of an object detector model, which locates the objects from the sticky paper trap images, and image classifier models that identify the detected objects. The deep learning models were trained using two different deep learning methods fitted for each type of installation site. For indoor sites such as greenhouses, a semi-supervised learning technique was applied to train a multi-class insect classifier model. The proposed technique was used to automatically collect training images from newly acquired sticky paper trap images and retrain the classifier model. After a year of continuously training the classifier model, F_1 -scores of 0.93 was achieved on testing both by object level and image level. For outdoor sites like orchards, multiple classifier models were trained and cascaded taxonomically to classify insect



pest images up to the species level. The cascaded algorithm was found to have average F_1 -scores of 0.91 and 0.89 by object level and image level testing, respectively. The system was installed in several agricultural production sites for farm managers to evaluate and utilize in their IPM routines. Data analytics was applied to extract information from the acquired long term data. Alarm models and biological models were developed that convert insect pest count data into valuable information such as alarm levels that indicate action recommendations for IPM, guiding farm managers in decision-making. Upon validation, the biological models were able to describe the flight rate behavior of the insect pests with an average r^2 of 0.97 based on the fitted models of all the experimental sites. The information was shared through a website and mobile APP which farm managers used to effectively respond to the present insect pest condition. The usage information and feedback given by the farm managers showed that the system was able to help them by having quantitative reference for their IPM strategies. This research presents different ways an automated system in assisting farmers for IPM applications, which can be used by researchers and experts in bringing farmers to data-driven and environmentally friendly insect pest management.

Keywords: integrated pest management, sticky paper trap, image recognition, deep learning, wireless sensor network, data analytics

Table of Contents



口試委員會審定書.....	i
Acknowledgements.....	ii
摘要.....	iii
Abstract.....	v
Table of Contents.....	vii
List of Figures.....	xiii
List of Tables.....	xxii
List of Equations.....	xxiii
Nomenclature.....	xxv
Chapter 1 Introduction.....	1
1.1 Background of the study.....	1
1.2 Statement of the problem.....	3
1.3 Significance of the study.....	4
1.4 Objectives.....	5
Chapter 2 Review of Related Literature.....	6
2.1 Integrated pest management (IPM).....	6
2.1.1 Monitoring.....	6



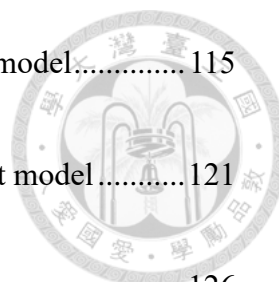
2.1.1.1 Insect pest traps.....	7
2.1.1.2 Insect pests.....	10
2.1.2 Prevention.....	11
2.1.3 Intervention	14
2.2 Artificial Intelligence of Things (AIoT) in IPM	16
2.3 Automatic insect pest detection and recognition.....	19
2.3.1 Image processing and machine learning methods.....	21
2.3.2 Deep learning methods.....	23
2.4 Decision support system (DSS) for IPM.....	29
2.4.1 Alarm model development	29
2.4.2 Biophysical model development	32
Chapter 3 Methodology	36
3.1 System overview	36
3.2 Monitoring system	37
3.2.1 Indoor sensor node	38
3.2.2 Outdoor sensor node.....	43
3.2.3 Network configuration	47

3.2.4 Device configuration and data transmission	48
3.3 Server architecture	50
3.4 Experimental sites	52
3.5 Insect pest detection and recognition algorithm.....	58
3.5.1 Object detector	59
3.5.2 Cascaded multi-class insect classifiers.....	63
3.5.3 Spatio-temporal voting method.....	70
3.6 Image data collection and preparation	72
3.7 Semi-supervised learning method	76
3.7.1 Base model building.....	77
3.7.2 Online image data collection.....	77
3.7.3 Unsupervised pseudo-labelling algorithm.....	79
3.7.4 Unsupervised model fine-tuning	84
3.7.5 Model update and selection.....	84
3.8 Algorithm evaluation.....	86
3.8.1 Object detector image level evaluation	87
3.8.2 Image classifier object level evaluation	87





3.8.3 Integrated algorithm evaluation.....	88
3.8.4 Semi-supervised learning evaluation	89
3.9 System performance indicators	91
3.9.1 System data throughput.....	92
3.9.2 System insect pest trapping efficacy	92
3.10 Data analytics	94
3.10.1 Insect pest count alarm model development	95
3.10.2 Insect pest hotspot detection	100
3.10.3 Insect pest flight rate model development.....	101
3.10.4 Crop growth information.....	103
3.11 System front-end	104
3.11.1 Website	105
3.11.2 Mobile APP	111
Chapter 4 Results and Discussion.....	115
4.1 Insect pest detection and recognition algorithm model training, validation and testing.....	115
4.1.1 Supervised object detector and insect vs. non-insect model	115



4.1.1.1 Indoor object detector and insect vs. non-insect model.....	115
4.1.1.2 Outdoor object detector and insect vs. non-insect model.....	121
4.1.2 Multi-class insect classifier model	126
4.1.2.1 Semi-supervised indoor multi-class insect classifier model.....	127
4.1.2.2 Supervised outdoor multi-class insect classifier models	140
4.1.3 Image-level testing results.....	148
4.1.3.1 Indoor insect pest detection and recognition testing results	148
4.1.3.2 Outdoor insect pest detection and recognition testing results	158
4.2 System performance testing	166
4.2.1 System data throughput analysis	166
4.2.2 System insect pest trapping efficacy analysis	171
4.3 Data analytics model development	186
4.3.1 Insect pest count alarm model.....	186
4.3.2 Insect pest flight rate model	197
4.4 Insect pest count data analysis	206
4.4.1 Indoor site data analysis	206
4.4.2 Outdoor site data analysis	218

4.5 User feedback.....	223
Chapter 5 Conclusions and Recommendations.....	227
5.1 Conclusions.....	227
5.2 Recommendations.....	230
References.....	233



List of Figures



Fig. 3-1. General architecture of the I ² PDM system.....	36
Fig. 3-2. Indoor sensor node functional block diagram.	38
Fig. 3-3. Actual setup of the sensor node in a greenhouse.....	40
Fig. 3-4. 3D schematic diagram of the image acquisition setup of the sensor node. a) Distance from camera to sticky paper trap; b) Camera field of view.	40
Fig. 3-5. 3D schematic diagram of the sensor node showing the sensor locations.....	41
Fig. 3-6. Schematic diagram of the indoor installation setup of the sensor node.	42
Fig. 3-7. Outdoor sensor node functional block diagram.	43
Fig. 3-8. Actual setup of the sensor node in an orchard.....	44
Fig. 3-9. Schematic diagram of the outdoor installation setup of the sensor node.	45
Fig. 3-10. Outdoor sensor node power control schematic diagram.	46
Fig. 3-11. Monitoring system network topologies: a) Star; b) Mesh.	47
Fig. 3-12. Server architecture block diagram.....	50
Fig. 3-13. Geographical locations of the experimental sites.....	52
Fig. 3-14. Installation setup map of Farm TS1.	54
Fig. 3-15. Installation setup map of Farm TS2.	54
Fig. 3-16. Installation setup map of Farm TS3.	55

Fig. 3-17. Installation setup map of Farm T1.	55
Fig. 3-18. Installation setup map of Farm T2.	55
Fig. 3-19. Installation setup map of Farm O1.	56
Fig. 3-20. Installation setup map of Farm O2.	56
Fig. 3-21. Installation setup map of Farm O3.	56
Fig. 3-22. Installation setup map of Farm S1.	57
Fig. 3-23. Installation setup map of Farm M1.	57
Fig. 3-24. Installation setup map of Farm C1.	57
Fig. 3-25. Insect pest detection and recognition algorithm flowchart.	58
Fig. 3-26. Object detector model flowchart.	61
Fig. 3-27. CNN image classifier architecture.	64
Fig. 3-28. Multi-class indoor insect classifier flowchart.	67
Fig. 3-29. Multi-class outdoor insect classifier flowchart.	69
Fig. 3-30. Spatio-temporal voting method (STVM) sample flowchart.	71
Fig. 3-31. Semi-supervised learning method flowchart.	78
Fig. 3-32. Unsupervised pseudo-labelling method flowchart.	81
Fig. 3-33. Sensor node trapping efficacy experimental setup.	93
Fig. 3-34. Insect count alarm model flowchart.	97
Fig. 3-35. I ² PDM website home page.	107





Fig. 3-36. I²PDM website temporal analysis page..... 108

Fig. 3-37. I²PDM website spatial analysis page. 109

Fig. 3-38. I²PDM website report page. 110

Fig. 3-39. I²PDM mobile APP home page. a) Insect pest count; b) Weather, and
pesticide calendar. 112

Fig. 3-40. I²PDM mobile APP analysis page. a) Map view; b) Table view; c) Insect
pest count analysis; and d) Environmental data analysis..... 113

Fig. 3-41. I²PDM mobile APP alarm notification sample. a) Notification; b) Insect
pest count alarm and pesticide information. 114

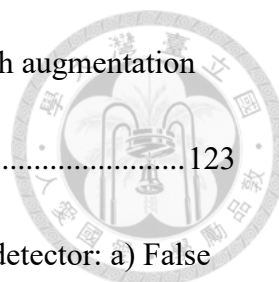
Fig. 4-1. Indoor object detector model training curve. 116

Fig. 4-2. Indoor insect vs. non-insect model validation and testing results: a)
Classification threshold tuning; b) Object-level testing confusion matrix without
augmentation; and c) Object-level testing confusion matrix with augmentation
(7200 samples)..... 118

Fig. 4-3. NMS threshold tuning curves of the indoor insect object detector: a) False
positive rate; b) Miss rate; c) F_1 -score..... 120

Fig. 4-4. Outdoor object detector model training curve..... 121

Fig. 4-5. Outdoor insect vs. non-insect model validation and testing results: a)
Classification threshold tuning; b) Object-level testing confusion matrix without



augmentation; and c) Object-level testing confusion matrix with augmentation
(8000 samples)..... 123

Fig. 4-6. NMS threshold tuning curves of the outdoor insect object detector: a) False
positive rate; b) Miss rate; c) F_1 -score..... 125

Fig. 4-7. Indoor multi-class insect classifier base model performance: a) Model
validation results; b) Object-level testing results using the best model..... 128

Fig. 4-8. Two-dimensional PCA feature projection of the indoor multi-class insect
classifier base model..... 130

Fig. 4-9. Sample results of pseudo-labelling mothfly image samples from images of
Farm TS1: a) Pseudo-labelled images via base model CNN classification; b)
Images closer to another centroid; and c) Images outside cluster ellipse $\omega = 3$.
..... 131

Fig. 4-10. Temporal semi-supervised learning results per class..... 133

Fig. 4-11. Temporal evaluation of the adaptive model based on the semi-supervised
learning online and offline metrics..... 135

Fig. 4-12. Semi-supervised indoor multi-class insect classifier model object-level
testing confusion matrix..... 138

Fig. 4-13. Supervised indoor multi-class insect classifier model object-level testing
confusion matrix..... 138

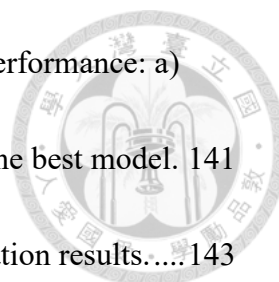


Fig. 4-14. Single stage outdoor multi-class insect classifier model performance: a) Model validation results; b) Object-level testing results using the best model. 141

Fig. 4-15. Multi-stage outdoor insect classifier Stage 1 model validation results.....143

Fig. 4-16. Multi-stage outdoor insect classifier Stage 2A model validation results. . 143

Fig. 4-17. Multi-stage outdoor insect classifier Stage 2B model validation results. . 144

Fig. 4-18. Multi-stage outdoor insect classifier Stage 3 model validation results..... 144

Fig. 4-19. Multi-stage multi-class outdoor insect classifier object-level testing confusion matrix results..... 147

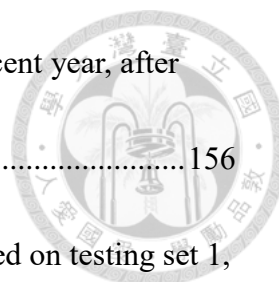
Fig. 4-20. Indoor object detector and insect vs. non-insect classifier model sample processed images: a) Image with detected images; b) Image with insect and non-insect images..... 149

Fig. 4-21. Indoor multi-class insect pest classifier model sample processed images: a) Image with mosquito, gnat, mothfly, fly, and whitefly images; b) Image with cranefly, midge, and thrips images. 151

Fig. 4-22. Scatter plots of the manual and automatic insect counts of the sticky paper images collected from the indoor installation sites, from the recent year..... 153

Fig. 4-23. Algorithm image level F_1 -score boxplots per class as tested on testing set 1. 154

Fig. 4-24. Scatter plots of the manual and automatic insect counts of the sticky paper



images collected from the indoor installation sites, from the recent year, after
 applying the spatio-temporal voting method 156

Fig. 4-25. Algorithm image level F_1 -score boxplots per class as tested on testing set 1,
 after applying the spatio-temporal voting method. 157

Fig. 4-26. Outdoor insect pest detection and recognition sample processed images: a)
 Image with leafhopper, fruitfly, gnat, and thrips; b) Image with oriental fruitfly,
 mothfly and mango leafhopper. Sample errors were marked with dotted circles.
 159

Fig. 4-27. Scatter plots of the manual and automatic insect counts from the sticky
 paper images collected in Farm M1, from the recent 6 months..... 161

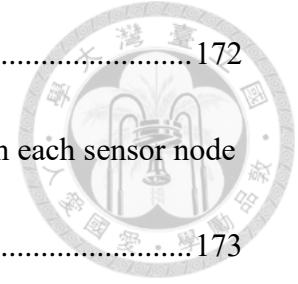
Fig. 4-28. Algorithm image level F_1 -score boxplots per class as tested on testing set 2.
 162

Fig. 4-29. Scatter plots of the manual and automatic insect counts from the sticky
 paper images collected in Farm M1, from the recent 6 months, after applying the
 spatio-temporal voting method. 163

Fig. 4-30. Algorithm image level F_1 -score boxplots per class as tested on testing set 2.
 164

Fig. 4-31. System data throughput analyses of Farms a) TS1, b) TS2, and c) O1..... 167

Fig. 4-32. Whitefly trapping efficacy experiment results obtained from each sensor



node of Farm TS1.	172
Fig. 4-33. Thrips trapping efficacy experiment results obtained from each sensor node of Farm TS1.	173
Fig. 4-34. Total whitefly trapping efficacy experiment results obtained from Farm TS1. a) Raw and normalized insect pest counts; b) Linear regression analysis.	175
Fig. 4-35. Total thrips trapping efficacy experiment results obtained from Farm TS1. a) Raw and normalized insect pest counts; b) Linear regression analysis.	176
Fig. 4-36. Whitefly trapping efficacy experiment results obtained from each sensor node of Farm TS2.	178
Fig. 4-37. Thrips trapping efficacy experiment results obtained from each sensor node of Farm TS2.	179
Fig. 4-38. Total whitefly trapping efficacy experiment results obtained from Farm TS2. a) Raw and normalized insect pest counts; b) Linear regression analysis results.	180
Fig. 4-39. Total thrips trapping efficacy experiment results obtained from Farm TS2. a) Raw and normalized insect pest counts; b) Linear regression analysis results.	181
Fig. 4-40. Whitefly trapping efficacy experiment results obtained from each sensor node of Farm TS3.	182

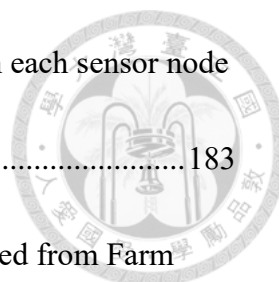


Fig. 4-41. Thrips trapping efficacy experiment results obtained from each sensor node of Farm TS3. 183

Fig. 4-42. Total whitefly trapping efficacy experiment results obtained from Farm TS3. a) Raw and normalized insect pest counts; b) Linear regression analysis results. 184

Fig. 4-43. Total thrips trapping efficacy experiment results obtained from Farm TS3. a) Raw and normalized insect pest counts; b) Linear regression analysis results. 185

Fig. 4-44. Insect pest count alarm models of Farm TS1. 187

Fig. 4-45. Insect pest count alarm models of Farm TS2. 188

Fig. 4-46. Insect pest count alarm models of Farm O1. 190

Fig. 4-47. Insect pest count alarm models of Farm M1. 192

Fig. 4-48. Fitted temperature vs. flight rate model of insects found in Farm TS1. ... 198

Fig. 4-49. Fitted temperature vs. flight rate model of insects found in Farm TS2. ... 199

Fig. 4-50. Fitted temperature vs. flight rate model of insects found in Farm O1. 201

Fig. 4-51. Fitted temperature vs. flight rate model of insects found in Farm M1. 203

Fig. 4-52. Sample effective pesticide application data of Farm TS2. 207

Fig. 4-53. Long-term temperature and insect pest count data collected from Farm TS1. Insect pest counts are presented as moving averages with a window size of 7. 209

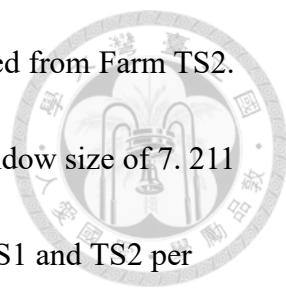


Fig. 4-54. Long-term temperature and insect pest count data collected from Farm TS2.

Insect pest counts are presented as moving averages with a window size of 7. 211

Fig. 4-55. Daily accumulated thrips and whitefly counts of Farms TS1 and TS2 per year. The total counts per year are showed in the legends.214

Fig. 4-56. Weekly insect pest count, growth stage, and rain level data of Farm M1. 219

Fig. 4-57. Relative frequencies of increase in insect pest count per class during each growth stage of the crops in Farm M1.221

Fig. 4-58. I²PDM APP user login activity data.223

List of Tables



Table 2-1. Major flying insect pests of economically important crops.....	12
Table 3-1. Specifications of the environmental sensors.....	42
Table 3-2. Data transmission frequency of the sensor node.	49
Table 3-3. Basic information on the installation sites.....	53
Table 3-4. Sticky paper trap image dataset partition information.....	73
Table 3-5. Indoor insect pest detection and recognition model training, validation, and testing image dataset statistics.	74
Table 3-6. Outdoor insect pest detection and recognition model training, validation, and testing image dataset statistics.	75
Table 4-1. Final total number of training images of the indoor multi-class insect pest classifier model after applying semi-supervised learning.....	139
Table 4-2. Data throughput analysis of the sensor nodes of each installation site.....	168
Table 4-3. Insect pest count alarm threshold values of all the installation sites.	193
Table 4-4. Insect pest flight rate model niche temperature per installation site.....	204
Table 4-5. Yearly reduction in thrips and whitefly counts of Farms TS1 and TS2....	215
Table 4-6. Thrips and whitefly counts alarm level statistics per year.	216
Table 4-7. Pesticide usage data of Farm TS2.....	217

List of Equations



Eq. 3-1. Intersection over union (IoU).....	62
Eq. 3-2. Majority voting class label	71
Eq. 3-3. Euclidean distance (d)	80
Eq. 3-4. Gaussian Mixture Model (GMM) covariance matrix (A)	82
Eq. 3-5. Cluster ellipse angle of rotation (θ_c)	83
Eq. 3-6. Inequality equation of a point inside an ellipse.....	83
Eq. 3-7. Precision.....	85
Eq. 3-8. Recall.....	85
Eq. 3-9. F_1 -score	85
Eq. 3-10. Miss rate (MR).....	87
Eq. 3-11. False positive rate (FPR).....	87
Eq. 3-12. Average F_1 -score	88
Eq. 3-13. Average counting accuracy (ACC)	89
Eq. 3-14. Cluster density (CD)	90
Eq. 3-15. Silhouette score (SS)	90
Eq. 3-16. Pseudo-labelling accuracy ($ACC_{pseudo,c}$).....	91
Eq. 3-17. Data throughput (%).....	92

Eq. 3-18. Data loss (%).....	92
Eq. 3-19. Normalize insect pest count (IC^*).....	94
Eq. 3-20. Anomaly score (S).....	96
Eq. 3-21. Isolation forest average path length (E).....	96
Eq. 3-22. Unsuccessful search average path length $c(n)$	96
Eq. 3-23. Exponential distribution function.....	98
Eq. 3-24. Change in insect count alarm level threshold ($TH_{\Delta Ca}$).....	98
Eq. 3-25. Moving average outlier condition.....	101
Eq. 3-26. Five-parameter double Weibull function.....	101
Eq. 3-27. Daily light integral (DLI) in ppfd.....	103
Eq. 3-28. Growing degree days (GDD).....	104



Nomenclature



Abbreviations

AI	Artificial Intelligence
AIoT	Artificial Intelligence of Things
AM	Adaptive Model
ANN	Artificial Neural Network
BM	Base Model
CCL	Connected Components Labeling
CL	Convolutional Layer
CNN	Convolutional Neural Network
DNN	Deep Neural Network
DSS	Decision Support System
GMM	Gaussian Mixture Model
HoG	Histogram of Gradients
I ² PDM	Intelligent and Integrated Pest and Disease Management
IoT	Internet of Things
IPM	Integrated Pest Management
LED	Light Emitting Diode
LM	Levenberg-Marquadt
PC	Principal Component
PCA	Principal Component Analysis
R-CNN	Region-based Convolutional Neural Network
ReLU	Rectified Linear Units
RPN	Region Proposal Network
SL	Softmax Layer
SSD	Single Shot Multibox Detector
STVM	Spatio-temporal Voting Method
SVD	Singular Value Decomposition
SVM	Support Vector Machine
TCP	Transmission Control Protocol
TCSV	Tomato Chlorotic Spot Virus

TCV	Tomato Chlorosis Virus
UDP	User Datagram Protocol
UM	Unsupervised Model
VGG	Visual Geometry Group
WSN	Wireless Sensor Network
YOLO	You Only Look Once



Variables

a	Alarm level
A	Gaussian mixture model covariance matrix
ACC	Average counting accuracy
$ACC_{pseudo,c}$	Pseudo-labelling accuracy of class c
c	Class index/number
C	Number of classifier target classes
CD	Cluster density
CI	Confidence interval
CL_c	Classification label of class c
d	Euclidean distance
DLI	Daily light integral
F	Insect flight rate
FPR	False positive rate
GDD	Growing degree days
ΔIC	Increase in insect pest count
IC	Insect pest count
IC^*	Normalized insect pest count
IC_{auto}	Automatic insect pest count of sticky paper trap held by sensor node
IC_{device}	Manual insect pest count of flat sticky paper trap held by sensor node
IC_{below}	Manual insect pest count of cylindrical sticky paper trap below sensor node
I_f	Final image index
I_i	Initial image index
IoU	Intersection-over-Union
k	Total number of data points in the isolation forest exit node

K	K-means clustering number of clusters
L_1	Initial pseudo-label
L_2	Second pseudo-label
LI	Light intensity
M	Total number of isolation forest binary splits
MR	Miss rate
m_x	Five-parameter double Weibull function parameters
$N_{clean,c}$	Number of clean pseudo-labelled images of class c
$N_{images,c}$	Number of images of class c
$N_{pseudo,c}$	Number of pseudo-labelled images of class c
N_{trees}	Number of isolation forest trees
p	Classification probability
R	Cluster ellipse radius
r^2	Coefficient of determination
S	Anomaly score
s	Number of samples
SS	Silhouette score
t	Time
T	Temperature ($^{\circ}\text{C}$)
$T_{baseline}$	Growing degree days baseline temperature ($^{\circ}\text{C}$)
$TH_{\Delta Ca}$	Insect count alarm threshold of alarm level a
TH_{CNN}	Classification probability threshold
TH_{IF}	Isolation forest anomaly threshold
TH_{MODEL}	Model fit r^2 threshold
T_{max}	Daily maximum temperature ($^{\circ}\text{C}$)
T_{min}	Daily minimum temperature ($^{\circ}\text{C}$)
$T_{niche,max}$	Insect flight rate peak niche temperature ($^{\circ}\text{C}$)
$T_{niche,min1}$	Insect flight rate lower niche temperature ($^{\circ}\text{C}$)
$T_{niche,min2}$	Insect flight rate upper niche temperature ($^{\circ}\text{C}$)
TP	True positive
u	Number of unique values of a dataset
v	Singular vector
ε	Euler constant



θ_c	Cluster ellipse angle of rotation
λ	Exponential distribution function scale parameter
σ_F	Insect flight rate standard deviation
ω	Pseudo-labelling algorithm scaling factor



Chapter 1 Introduction



1.1 Background of the study

Integrated pest management, or commonly known as IPM, is one of the most sought techniques for agricultural production management (Lamichhane et al., 2016; Potamitis et al., 2017). IPM can provide solutions for most common agricultural crop problems through systematic management of pests. This includes techniques such as insect pest population monitoring, environmental control, pesticide scheduling, and more.

A survey conducted by Parsa et al. (2014), shows that IPM is having a slow progression in terms of application and awareness among countries. They were able to assess the current status of IPM in different countries which showed that IPM is not fully implemented and known by other people. Currently, IPM is only known to selected countries due to inefficient orientation and lack of technical support to farmers (Parsa et al., 2014). IPM plans in different countries end up in a brainstorming stage that were never executed due to government policies, funding shortages, etc. Even in developed countries, IPM is still facing a challenge to prove its worth. Therefore, IPM technologies and solutions should be capable enough to prove its potential for actual

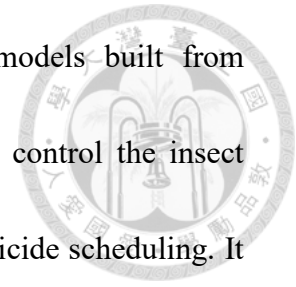
implementation and demonstrate its benefits.



With the rise of technology and the urgent need for IPM technologies, researchers worked on developing IPM solutions that can be widely accepted and used. Most researchers focused on discovering and developing efficient ways to identify insect pests from traps using image processing, machine learning, and deep learning methods (Espinoza et al., 2016; Miranda et al., 2014; Xia et al., 2015; Zhong et al., 2018). Following the rising demand of artificial intelligence (AI) in different industries, the application of machine learning in agriculture was also widely discussed. Through the use of AI, identification of insect pests was done more extensively compared to previous methods developed (Ding & Taylor, 2016; Xia et al., 2015). The availability of imaging and wireless technology made it more possible to design automatic wireless monitoring systems that was found to be useful for wide scale monitoring (Lamichhane et al., 2016; López Granado et al., 2012; Rustia et al., 2020b). Identification of insect pests and data gathering were carried out by software and in real-time. The integrated systems can support the farmers and provide possible solutions to protect their crops from potential damage caused by insect pests.

Data analytics may also help in providing valuable information for the farmers and help in their decision-making (Donatelli et al., 2017; Ferguson et al., 2016; Qin et al., 2017). A system should include data analytics that transforms numerical data into

information that users can easily comprehend. Mathematical models built from different data sources can act as references for systems that can control the insect population by several ways such as environmental control and pesticide scheduling. It can also explain the possible causes of insect pest outbreaks and predict its possibility in the future.

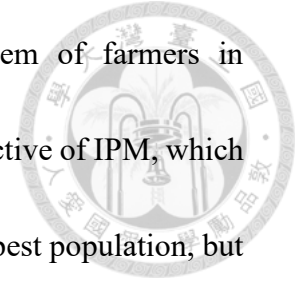


Implementation of smarter IPM in agriculture is a challenge not too far from reality due to the efforts of researchers, government agencies, and industries. This research aims to contribute to this progress by demonstrating the advantages of having quantitative insect pest information through an intelligent and integrated system. This study involves not only laboratory tests but also actual distribution of the system to farmers that need data-driven IPM information.

1.2 Statement of the problem

The concept of IPM is still unfamiliar to some farmers all over the world. Crop damage and economic loss are usually caused by insect pest outbreaks. However, there are times when farm managers cannot respond immediately due to lack of reliable quantitative information that can be only obtained through detailed analyses and monitoring of insect pest related data. This calls for the need of an automated system

that can provide data-driven knowledge to alleviate the problem of farmers in conducting IPM. To date, only a few systems meet the over-all objective of IPM, which is to develop an integrated system that can not only monitor insect pest population, but also to understand their behavior as well. Most of the studies related to IPM systems were done based on controlled environments or by simulations. In this work, data from uncontrolled environments were collected using an automated monitoring system to implement a fully adaptive system for IPM support.



1.3 Significance of the study

This study provides an automated system for monitoring the insect pest population in agricultural production sites. The system can reduce the required time for farmers to develop solutions for insect pest related problems. Through the data-driven system, holistic integrated pest management strategies can be developed. It will be useful in conducting deeper studies on insect pest behavior and assist farmers in addressing their insect pest related concerns.

1.4 Objectives



This research aims to develop an intelligent system that can be utilized for IPM application through automated insect pest population monitoring and decision-making support. The specific objectives to fulfill this goal are as follows:

- To design a monitoring system, composed of wireless sensor nodes, that can acquire sticky paper trap images and measure environmental conditions remotely
- To develop an algorithm that can automate the detection and recognition of insect pests on sticky paper trap images
- To provide data analytics service to the users of the system, through the use of alarm and biological models, for assistance in IPM decision-making
- To discover the actual benefits of using an intelligent IPM system by data analysis and collection of user feedback

Chapter 2 Review of Related Literature



2.1 Integrated pest management (IPM)

Integrated pest management (IPM) is a holistic and systematized approach for managing insect pests by applying multi-disciplinary agricultural techniques. IPM aims to optimize agricultural production without causing harm to people and the environment. According to Barzman et al. (2015), IPM can be divided into three key components: monitoring, prevention, and intervention. Each component plays an important role in an effective IPM program. This section discusses the techniques and concepts used in each component as practiced and studied by farmers and researchers.

2.1.1 Monitoring

The core of an effective IPM program is monitoring. Monitoring intends to quantify the population of insect pests in a farm. It is done by strategically placing traps on suspected insect outbreak hotspots in a farm. After finding out which kind of insects and the number of insect pests exist in the farm, farmers can evaluate the gravity of the insect pest situation. Without monitoring, farmers will not have any reference for decision-making. This sub-section discusses about common types of insect traps and

insect pests found in different agricultural production sites.



2.1.1.1 Insect pest traps

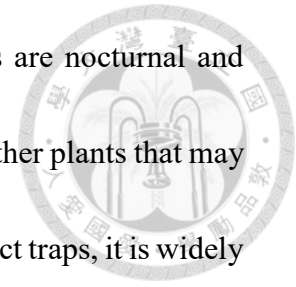
There are three common kinds of insect traps: pheromone traps, light traps and sticky paper traps (Böckmann & Meyhöfer, 2017; Stukenberg et al., 2015; Witzgall et al., 2010). Each kind of insect trap is designed based on the visual and olfactory sensual cues that attract insects.

A pheromone trap is a species-specific insect trap that uses chemical attractants released by an insect's gender counterpart (Yin & Maschwitz, 1983). A simple pheromone trap design is an open container that includes a pheromone with its interior covered with adhesive material. Insects are attracted by the pheromone forcing them to fly or crawl towards the inside of the container (Witzgall et al., 2010). Another design is composed of an anti-insect net hung on a tree or post with a pheromone inside. Using nets can be used for trapping more specific sizes of insect pests (Miluch et al., 2013). In some cases, a pheromone trap is used for controlling insect populations instead of pesticides since it can be used for a long period of time (Witzgall et al., 2010). Most importantly, it is an environmentally friendly approach since it does not require the use of harmful chemicals. It may also help in preventing an outbreak of insect pests that

have developed immunity against pesticides. However, while selectivity is considered as its advantage, it becomes a disadvantage for other kinds of crops (Lewis et al., 1997; Yin & Maschwitz, 1983). Other types of crops attract more than one species of insect pests making other insect pests not detectable by pheromone traps. In such case, the use of light traps or sticky paper traps is recommended.

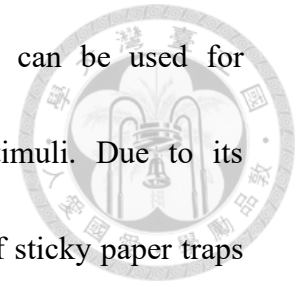
A light trap targets the visual photoreception of insects. Like any other animal, insect pests are attracted by specific wavelengths of light (Shimoda & Honda, 2013; Stukenberg et al., 2015). A light trap is placed near plants to attract and zap the insects as they touch the light trap's cage. A light trap is commonly found in a household to trap mosquitoes and flies. Such kind of trap uses ultraviolet light because common household insect pests have positive attraction to it (Shimoda & Honda, 2013). In other cases, specific wavelengths of light are used. In the work of Chu et al. (2003), lime-green (530 nm) LEDs were installed inside plastic cup traps for monitoring whiteflies, one of the most sought greenhouse insect pests. A similar study was done by Stukenberg et al. (2015) which showed that using green (517 nm) light emitting diodes (LEDs) with ultraviolet (368 nm) LED support had the highest efficiency in trapping whiteflies compared to other color combinations of LEDs. Similar to pheromone traps, the use of light traps can be an alternative to spraying pesticides as well. Its only disadvantage is its reliance to power sources which are not commonly available in some farms. It is

also less precise compared to pheromone traps since most insects are nocturnal and easily attracted to light. In worst cases, it may affect the growth of other plants that may wilt due to excessive exposure to light. But unlike other types of insect traps, it is widely available in local stores and convenient for home use.



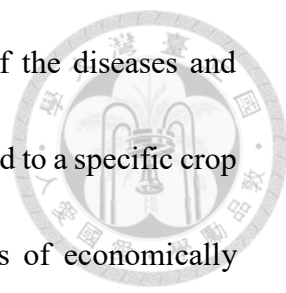
The use of sticky paper traps is the most cost-effective and commonly used insect pest monitoring method (Martin & Moisan, 2017; Moreau & Isman, 2011; Sampson et al., 2018). A sticky paper trap is a paper or board coated with adhesive material on either side. Similar to light traps, sticky paper traps have different colors such as blue or yellow based on the colors that the target insect pests are attracted to. The most commonly used sticky paper trap color is yellow. Yellow sticky paper traps attract insect pests such as whitefly, aphid, thrips, midges, and more. Due to the variety of insect pests it can attract, it is a more generalized option compared to light traps and pheromone traps. Using sticky paper traps does not completely affect the behavior of the insect pests in the farm; making it applicable for research purposes. Moreau and Isman (2011) compared the reliability of using sticky papers and trap crops for trapping whiteflies. Their study showed that whiteflies had better response and attraction to the sticky paper traps. Prema et al. (2018) analyzed the effects of different sticky paper trap colors for attracting thrips, a major pest of cotton. It was found in their study that thrips were most attracted to yellow sticky paper traps next to blue and white sticky paper

traps. Their results proved that the use of sticky paper traps can be used for understanding the behavior of thrips in response to visual stimuli. Due to its applicability for both on-field and research applications, the use of sticky paper traps was used as the trapping method of the proposed system in this study.



2.1.1.2 Insect pests

An insect pest is a kind of insect that can inflict damage or transmit diseases to crops. Species of insect pests vary from place to place depending on the crops grown. For an instance, tomato, one of the high-value crops in the world, is a host crop of insect pests such as whitefly, thrips, and aphid (Navas-Castillo et al., 2011). One of the most devastating diseases of tomatoes inflicted by such insect pests is tomato chlorosis virus (TCV). TCV causes shrinking of tomato fruit yields and brittleness of leaves that eventually kills the plant. TCV is very common in Asian countries such as Taiwan and Japan (Navas-Castillo et al., 2011). Meanwhile, a common disease that damages flower crops is tomato chlorotic spot virus (TCSV). TCSV mainly affects flower crops such as calla lilies, *Phalaenopsis*, and *Lisianthus*. TCSV can cause wilting and deformation of leaves; making flower crops unmarketable (Jones, 2005). Most diseases, including the aforementioned, are transmitted within a day or two by insect pests. This means that



properly identifying the insect pests can prevent the occurrence of the diseases and avoid economic loss. Knowing which kind of insect pests are attracted to a specific crop will help farmers in decision-making. A list of major insect pests of economically important crops globally (Csizinszky et al., 1995; Hazarika et al., 2009; Navasero & Navasero, 2015) is presented in Table 2-1.

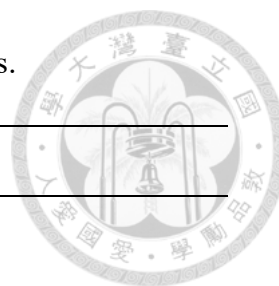
2.1.2 Prevention

Prevention aims to manage the population of insect pests through a cultural, physical, or biological approach. Some of the most effective prevention methods are selection of crop varieties, environmental control, and introduction of parasitoids.

In the study of Pelgrom et al. (2014), wild variants of cabbage crops were tested for resistance from cabbage whiteflies. Their results revealed that wild cabbage species have high levels of resistance and immunity against damages of cabbage whiteflies as compared to other cabbage species. Mirnezhad et al. (2010) also had similar findings by testing wild species of tomato for resistance and immunity to thrips damage while cultivated tomatoes did not. Rossetto et al. (2017) searched for mango plant varieties that are resistant to fruit fly damage. Their work showed that local varieties from Brazil such as *Alfa*, *Espada Varmelha*, and a few more, were resistant. It was also found that

Table 2-1. Major flying insect pests of economically important crops.

Insect name	Order and family	Host crop/s
Whitefly	Hemiptera: Aleyrodidae	Tomato
		Cucumber
		Cabbage
		Potato
		Orchids
		Tea
Thrips	Thysanoptera: Thripidae	Tomato
		Mango
		Cucumber
		Pepper
		Cabbage
		Potato
		Onion
		Orchids
Aphid	Hemiptera: Aphididae	Tomato
		Cucumber
		Pepper
		Cabbage
		Orchids
		Tea
Leafhopper	Hemiptera: Cicadellidae	Tea
		Rice
Gnat	Diptera: Sciaridae	Orchids
Fruit fly	Diptera: Tephritidae	Mango
		Cucumber
		Guava
Diamondback moth	Lepidoptera: Plutellidae	Cabbage



the resistance of mango varieties was correlated to the amount of milk volume that can be extracted from the cut fruits. The cited works demonstrate the selection of crop varieties to prevent thrips damage. It was observed that choosing wild species of tomato can be a very efficient way to prevent damage of insect pests. However, this may be subject to availability as not all countries may have the specific crop variety.

An alternative and safe approach for preventing insect pest damages in indoor agricultural production sites is through environmental control. By setting the temperature and humidity levels in a greenhouse or orchard, insect pest reproduction can be controlled. van Lenteren and Martin (1999) mentioned that greenhouse design specifications such as its size, height, shape, cladding materials, and other factors, largely affect the environmental condition and the behavior of insect pests. By properly considering the development requirement of insect pests and the crops grown, a balance can be achieved in which crops grow well while insect pests rarely show up. According to them, this is also the same when considering the incidence of plant disease infection.

Another approach for prevention is by introducing parasitoids in the agricultural site (van Lenteren & Martin, 1999). Parasitoids are insects that feed on targeted host insect pests. One example of a parasitoid is the *Encarcia Formosa*, which feeds on greenhouse whiteflies. Upon release, *E. formosa* preys on whitefly larvae found on plant leaves (van Lenteren et al., 1996). The use of parasitoids for whitefly control was

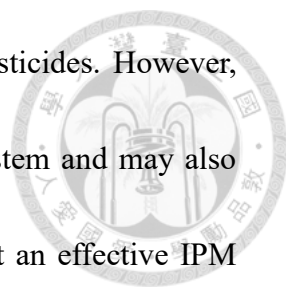
found capable of reducing the population of whiteflies in a farm. Murai and Loomans (2001) mass-reared *Ceranisus menes*, a parasitoid of thrips. They performed experiments to optimize the reproduction of *C. menes* such that the population of thrips in a farm can be completely controlled. This describes one constraint in using parasitoids for insect pest control. Apparently, there should be an adequate number of parasitoids introduced to an agricultural production site to effectively exterminate selected insect pests.

All the prevention methods presented involved safe and environmentally friendly approaches for insect pest population control. However, in extreme cases in which these solutions are not applicable, IPM proposes the use of intervention methods.

2.1.3 Intervention

The intervention component of IPM aims to exterminate insect pests by means of chemical control. This includes the use of pesticides, fungicides, and repellants. Applying intervention techniques are very effective for totally controlling the population of insect pests but may cause several negative ecological implications.

According to Lewis et al. (1997), there is a so-called treadmill in which farmers highly rely on intervention techniques to solve their insect pest issues. The most



common intervention technique applied by farmers is spraying pesticides. However, conventional pesticide spraying can cause disruption in the ecosystem and may also cause insect pests to gain resistance to pesticides. This means that an effective IPM strategy should not often come to a point in which intervention is required. Intervention should be conducted only for extreme scenarios such as when the population of insect pests can no longer be controlled by natural methods. They also compared the amount of pesticides used by IPM and by conventional methods which showed that there was around 80-90% improvement in terms of efficiency in spraying pesticides. This proved that IPM did not only prevent damages to the environment, but also reduced the budget used for controlling insect pests.

To make spraying pesticides more efficient, pesticide scheduling can be proposed. Jones et al. (2016) demonstrated a method for optimal pesticide scheduling using control theory. A control model was developed based on previous pesticide schedules to know the best times for spraying pesticides. However, it was mentioned in their work that problems still occurred whenever the model notified the user prematurely. Their results show a significant reduction in the frequency of pesticide application. To further improve the method, it was suggested that real-time data should be used to update and optimize the schedule from time to time.

2.2 Artificial Intelligence of Things (AIoT) in IPM



A novel concept called Artificial Intelligence of Things (AIoT) has been introduced in the recent years to assist in accelerating the development of smart farming. It combines the power of connectivity from Internet of Things (IoT) and the data-driven knowledge obtained from AI. AIoT rose due to the increasing popularity of AI as modern devices started to be capable of performing more complex and high-speed computations for many different applications. With the convergence of IoT and AI, more advanced and smarter systems can be developed.

IoT is a system composed of devices or machines that are connected and working together without human intervention. In the past decade, IoT has been introduced as a way to invite researchers, government agencies, farm owners, and other related parties, to work together in solving agricultural challenges. Lamichhane et al. (2016), highly suggested using IoT to address common challenges in agriculture, specifically IPM. They were convincing people to form a collaborative network to discuss and share information about IPM. IPM networking was also proposed from an article written by Parsa et al. (2014). They mentioned that if more people became aware of the benefits of IPM, the usage of IPM will be shared to fully utilize IPM. This was demonstrated from the scenario presented by McKee (2011) which showed that coordinated insect

pest management decisions among farmers helped in preventing insect pest outbreaks.

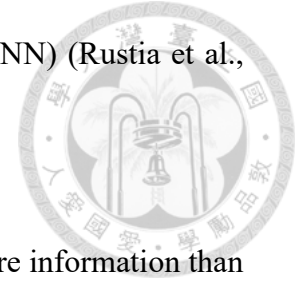
Farmers worked together by regularly inspecting their own crops and informing other farmers if there was a possibility of having a whitefly outbreak in their farms. IoT has a very important role in conducting IPM by being a platform to unite the ideas of farmers and experts for convincing more people to apply IPM.

Wireless sensor networks (WSN) serve as the ears and eyes of IoT. A WSN is a group of wirelessly connected devices that records spatial and temporal data. In agriculture, the basic function of a WSN is to measure different environmental conditions such as temperature, humidity, soil moisture, CO₂ levels, and a lot more. A WSN can also be used for counting insect pests for IPM application. Potamitis et al. (2017) presented several WSN systems that count insect pests from traps using detection sensors. The systems mentioned in their work used optical sensors, such as photodiodes, that were triggered whenever an insect pest fell into a pitfall trap. Meanwhile, Iqbal et al. (2019) demonstrated an automatic fly monitoring system for greenhouses. Their device used a Hall effect sensor which was triggered by flies that were attracted to a light source. Parsons et al. (2019) wrote an article about several commercially-available IPM WSN systems. One of the products they featured included Trapview, a system that uses cameras for capturing images inside a pheromone-based trap. The insect pest counts recorded by the system are all sent to a remote server so

that users can view their data through a user interface. Another product was the Spensa Z-trap, which is also a pheromone-based trap that included a series of electrified rods that electrocutes insect pests. The trap lures insect pests to fall into a bucket and are counted using bioimpedance sensors. The cited examples are systems that were designed for counting specific kinds of insect pests. In other cases, it is necessary to identify more kinds of insect pests. This is the time AI steps in to improve current insect pest monitoring systems.

AI is a field in computer science that simulates human intelligence through machines such as computers. AI encompasses other specific fields in computer science such as machine learning and deep learning. Nowadays, AI has been a very useful tool for image analysis replacing and/or improving traditional image processing methods. AI has been used in IPM to develop algorithms for detection and recognition of multiple insect species (Espinoza et al., 2016; Rustia et al., 2019; Zhong et al., 2018). Together with IoT, image-based insect pest monitoring systems were built with the support of AI. Zhong et al. (2018) used AI to count and recognize outdoor flying insect pests from wireless imaging devices. Miranda et al. (2014) used Kohonen self-organizing maps neural network for recognizing insect pests in a rice field. Their system included multiple wireless cameras which sends images of sticky paper traps to a remote server for processing. We also applied AI in our previous work for detecting and recognizing

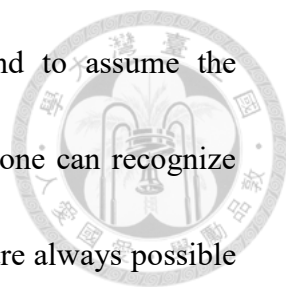
greenhouse insect pests through convolutional neural networks (CNN) (Rustia et al., 2019; Rustia et al., 2018).



AI can also be used for interpreting patterns and extracting more information than statistical methods. As a matter of fact, AI was used to create forecasting models for predicting occurrences of insect pest outbreaks (Cai et al., 2019b; Lu et al., 2019; Xiao et al., 2019) In the work of Cai et al. (2019b), multi-dimensional data from a meteorological station was used to raise alarms of possible insect outbreaks. They trained a model to predict and classify insect pest damages into four different alarm levels. On the other hand, Skawsang et al. (2019) developed a method for predicting the rice insect pest population as measured using light traps. They trained an artificial neural network (ANN) model based on meteorological data and satellite-derived plant variables. From the examples, AI was able to utilize the collected data through IoT and form more meaningful interpretations useful for decision-making in IPM. AI get rids of the limitations of IoT as it builds a bridge between the users and the information they will need.

2.3 Automatic insect pest detection and recognition

Normally, insect pests on sticky paper traps are counted by manual inspection. But



since insects are relatively small to the naked eye, people tend to assume the approximate number of insects captured from the traps. Not everyone can recognize which kind of insect pests are found from the traps. Human errors are always possible and may lead to risky insect pest management decisions. This is also a trouble for agricultural government agencies in conducting farm inspections (Pundt, 2013). Regardless of the insect trap used, monitoring is considered as a very cumbersome and time-consuming routine. This inspired a lot of researchers to develop automatic insect pest detection and recognition methods to obtain accurate quantitative data.

Numerous methods for automatic insect pest detection and recognition in sticky paper traps had already been done in the past. Cameras were used to obtain images of the sticky paper traps or of the plant leaves. In other works, flatbed scanners were used to obtain images with higher resolution (Cho et al., 2007; Xia et al., 2015). Algorithms were developed based on these different image acquisition methods which involve the use of several automatic image analysis methods such as image processing, machine learning, and deep learning. This section discusses concepts and examples of applying the different methods for automatic insect pest detection and recognition.

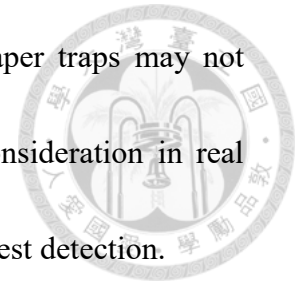
2.3.1 Image processing and machine learning methods



Image processing is a technique for image analysis which involve operations that are used to extract information from images. In insect pest detection and recognition, image processing can be used to perform operations such as image segmentation, feature extraction, and image classification.

Image segmentation is an operation for separating the background and foreground of an image using threshold values based on its color components. As applied on a sticky paper trap image, image segmentation is used to remove the yellow background of the image and obtain the location of the target insect pest objects based on its non-zero pixels (Barbedo, 2014; Qiao et al., 2008; Rustia et al., 2018). In the work of Cho et al. (2007), image segmentation was performed by static color thresholding. Removal of the yellow background was carried out by using pre-set threshold values. The insect pest objects in the image were detected and classified based on their size, color, and morphological features by object analysis. Meanwhile, Miranda et al. (2014) applied background subtraction for image segmentation. Background subtraction is an operation that computes the difference between an input image and a reference image. Unlike thresholding, this is a simpler approach since it does not require tuning of any threshold value. However, the bottleneck of using image segmentation for detection is the presence of impurities in the image. Images from recent studies were taken from

controlled lighting environments which means that the sticky paper traps may not contain dirt or other foreign objects. This should be taken in consideration in real applications that will require more adaptive approaches for insect pest detection.



After getting the locations of the insect pest objects from a sticky paper trap image, feature extraction and image classification is performed. Feature extraction is used to convert the RGB pixel data of an image into features such as color, size, shape, and others. A machine learning model is fit based on the features for image classification. Kumar et al. (2010) presented a method for automatic classification of whiteflies and greenflies using Gabor filters, histogram of gradients (HoG), and color features. The selected features were able to withstand differences in image acquisition lighting condition. Meanwhile, Mundada and Gohokar (2013) used shape and morphological features as input to a support vector machine (SVM) classifier model for classifying whiteflies and aphids. The image features used in their work were image entropy, mean pixel values, contrast, and more. In our previous work, we tested image features such as size, color, shape, and morphological features for classifying segmented objects from a sticky paper trap image into insect or non-insect (Rustia et al., 2017). The features were fitted to an SVM model that had a classification accuracy of 0.85. However, we observed that the features of the objects changed from time to time and were dependent on the image acquisition condition. As a more adaptive approach, RGB pixel values

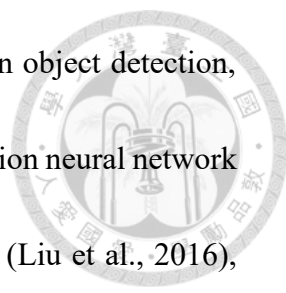
were used as input features to boost the accuracy up to 0.93 (Rustia et al., 2020b).

It was shown from the cited works that image processing is a useful tool for detecting and recognizing insect pests from sticky paper trap images. But, there are still some instances that it is not adequate. In cases in which the appearance of the insect species on the sticky paper traps are too similar, classifying the insects by size, color, and other features may be more difficult. Therefore, more complex operations such as the use of deep learning techniques is recommended as a robust and adaptive approach.

2.3.2 Deep learning methods

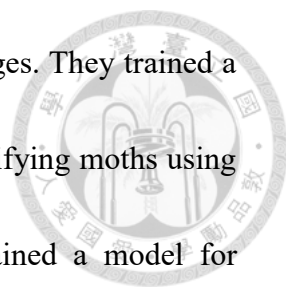
Deep learning has been very effective for insect pest detection and recognition as it can be used to further improve the accuracy of current algorithms (Espinoza et al., 2016; Rustia et al., 2019). Unlike in image processing or machine learning, object detection, feature extraction, and image classification can be performed in deep learning by merely training a neural network model to do such tasks. This gets rid of the trouble in developing feature extraction techniques or segmentation methods. It also makes the algorithm more robust to variations in the image samples.

The challenge in deep learning is selecting the appropriate neural network structure for the target application. This includes tuning and optimization of the network



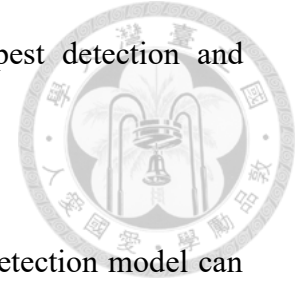
parameters such as size, complexity, number of layers, and more. In object detection, some of the well-known structures are Faster Region-based convolution neural network (R-CNN) (Ren et al., 2015), Single Shot Multibox Detector (SSD) (Liu et al., 2016), and YOLO (Redmon & Farhadi, 2018). Object detection model structures are different in terms of their detection algorithm approaches. The detection approach affects the models' computation speed and accuracy. SSD uses a so-called MultiBox strategy in proposing bounding boxes and detecting objects with a single step. It performs classification by computing the class scores for each detected object. In contrast, Faster R-CNN initially extracts features from the image using a convolutional layer followed up by a region proposal network (RPN). The RPN is a network that slides through the extracted feature map while detecting and classifying the objects. This makes SSD faster than Faster R-CNN by optimizing the detection speed with a few sacrifices in accuracy due to potentially missed detections. On the other hand, YOLO divides an image into grid cells. For each grid cell, bounding boxes are predicted and class probability are computed. Compared to the two other object detector structures mentioned, YOLO is the fastest as it was developed intentionally for real-time detection.

For insect pest detection and recognition, the object detection models are trained to detect custom objects such as of insect pests. The usual approach in doing so is to detect and classify the insect pests using a single network model. Ding and Taylor (2016)



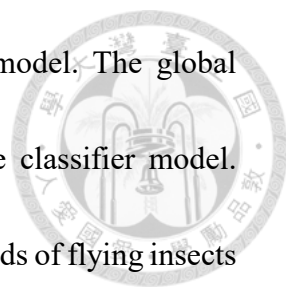
used this approach for automatic detection of moths from trap images. They trained a convolutional neural network (CNN) model for detecting and classifying moths using a sliding window approach. Similarly, Fuentes et al. (2017) trained a model for detection of insect pests and diseases on plant leaves. The model architecture they used for object detection and classification was Faster R-CNN (Ren et al., 2015), with VGG-16 (Simonyan & Zisserman, 2014) as feature extractor. In the work of Partel et al. (2019), a system was designed for monitoring Asian citrus psyllids in orchards. They used CNN to detect and count insect pests from a viewing board that holds a sticky paper trap in place. Their results showed that using CNN produced reliable results even as used on a moving system. Hong et al. (2020) tested several object detection models for detection of moths from pheromone traps. It was found from their results that Faster R-CNN has the highest accuracy compared to other architectures tested. From the related works, it shows that deep learning produced satisfactory results that are reliable for different environments and devices. The only downside of using a single stage or model object detection and recognition model is the preparation of training samples and the time required for training. Normally, training samples for object detection models are prepared by annotating or marking the objects in input images. In large-scale applications, this is very time consuming and confusing. A solution to this problem is to separate the detection and recognition components of the model and train two

separate models, which is an alternative approach for insect pest detection and recognition using deep learning.



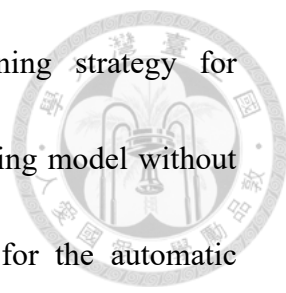
By using separate models for detection and recognition, the detection model can be used as a black box detector, same strategy as when image segmentation by image processing is done. On the other hand, the recognition model can be retrained easily based on newly sorted images to improve the current model or add new classes. Some of the famous neural networks structures for image recognition are AlexNet (Krizhevsky et al., 2017), VGG (Simonyan & Zisserman, 2014) and Inception (Canziani et al., 2016). The difference between neural network structures is their levels of deepness, which is based on the number of convolution layers a neural network has. For faster computation, AlexNet may be a good option since it only has 8 layers. On the other hand, Inception, which has 48 layers, may be more capable of classifying complex images but will need a longer processing time. Depending on the input images, it is recommended to use a neural network suitable for the application (Chevalier et al., 2016).

The strategy of using a separate object detection model and image recognition model for insect pest detection and recognition was seen from the work of Zhong et al. (2018). They used You Only Look Once (YOLO) model for detecting flying insect pests in sticky paper trap images, while using SVM as classifier model. The locations of the



insect pests were obtained through the trained object detection model. The global features of each insect pest image were extracted as input to the classifier model. Through this approach, they were able to classify up to 6 different kinds of flying insects with an average accuracy of 0.90, higher than using the object detection model for both detection and recognition. In our previous work, we also used the same approach by using Tiny YOLO v3, as object detection model, and two low resolution CNN classifier models for greenhouse insect pest detection and recognition in sticky paper trap images (Rustia et al., 2019). First, the object detector model was used to detect the objects from the sticky paper trap image. The first stage CNN model classifies the objects into insect or non-insect to exclude impurities and foreign objects from classification. The second stage CNN model then classifies the insect objects according to five kinds of greenhouse insect pests. In the same work, we also applied an online learning approach in which the second stage classifier was retrained from time to time to collect new samples and continuously improve the accuracy of the model. This allowed the system to continuously learn from the new insect pest images acquired.

Another factor to consider in deep learning is the learning strategy. In most cases, deep learning models are trained by so-called supervised learning. In supervised learning, all the training samples of the model are prepared manually. But as used in uncontrolled environments, an adaptive approach such as by self-supervised learning is



more recommended. Self-supervised learning is a deep learning strategy for continuously collecting training samples for updating a deep learning model without human supervision. A study on online self-supervised learning for the automatic segmentation of dynamic objects was demonstrated by Guizilini and Ramos (2015). Their work proposes a method for autonomously collecting images using an uncalibrated moving camera for training a segmentation model. This is a method used for self-driving vehicles that need to continuously learn from unpredictable environments. In medicine, self-supervised learning has been applied by Chen et al. (2019) in training models for the detection and segmentation of medical images. They developed an image context restoration strategy that can learn mapping from paired images to automatically annotate unlabeled medical images. We also presented a method for self-supervised learning in our previous work for automatically training image recognition models of our insect pest monitoring system. Our method uses a pre-trained convolutional neural network for automatically sorting and classifying and insect pest images and updating the classifier model on each cycle (Rustia et al., 2019).

From the cited works, it was proven that deep learning is a more reliable and adaptive approach for detection and classification of insect pest images. Unlike image processing and machine learning, deep learning reduces the time in developing models without sacrificing the performance of the algorithm.

2.4 Decision support system (DSS) for IPM




A decision support system (DSS) is one of the most important components of an insect pest monitoring system. By the use of statistical and machine learning methods, the data obtained from the system can be further transformed into information that most users can comprehend. This section discusses about how a DSS is developed using techniques in data analytics for defining alarms and developing biophysical models that allow users of the system make data-driven decisions.

2.4.1 Alarm model development

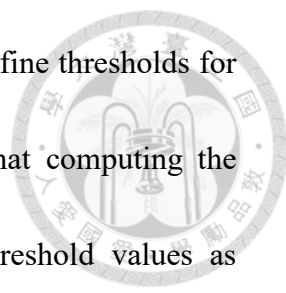
One way to extract meaningful data from the recorded insect pest counts is through the development of an alarm model. An alarm model transforms the insect pest count into several alarm levels that specify different suggested actions for insect pest management. Up-to-date, there is no standard way for defining alarm levels for insect pest counts collected from sticky paper traps (Pinto-Zevallos & Vänninen, 2013). Guidelines are provided by several agricultural agencies but not all are suitable for different farm environments and conditions.

Cohen et al. (2008) proposed a method for defining alarms and action recommendations for Mediterranean fruit fly control. A decision tree was defined based



on the trap counts manually collected from different farm regions and temperature thresholds for reproduction of fruit flies. They provided different recommendation levels for spraying pesticides and evaluated the reliability of the recommendations based on the users' feedback. It shows that the farmers agreed with the recommendations of the system 56% of the time. Ferguson et al. (2016) proposed a threshold determination method for pollen beetle management. They used the historical data, recommended threshold values of government agencies, and public weather data for defining threshold levels that indicate possible migration of pollen beetles. Miranda et al. (2019) presented a system that uses an automated insect pest monitoring system for developing a DSS for olive fruit fly management. They used the collected trap count data, weather data, and tree information for determining the risk of fruit fly infestation. Their proposed system also sent out recommended spraying times for an automated tractor. The system was able to reduce the amount of pesticides used for controlling the population of the olive fruit flies.

There were also several attempts on automatically defining alarm thresholds for decision making. Zhang et al. (2018) used K-means clustering, an unsupervised machine learning technique, for grouping automatically collected insect pest counts into four different levels. They used the four different levels to develop an insect outbreak severity forecasting method using multivariate data. In medicine, Guagliardo



et al. (2018) proposed and compared mathematical methods that define thresholds for public response to monkeypox. It was found from their work that computing the cumulative sum was the most adaptive method for defining threshold values as compared to other methods such as a method provided by the World Health Organization and a so-called Cullen method. They concluded that it was too simplistic to use a single method for defining the threshold values. Burr et al. (2013) compared different parametric distribution functions for finding anomaly thresholds in process monitoring. Their results showed that normal, lognormal, or a mixture of normal functions produced reasonable anomaly threshold values. They also recommended probability values of 0.001 or 0.025 for maintaining a low false alarm probability. The study showed that a common problem in defining alarm thresholds is finding out reasonable threshold levels that can actually help in decision-making.

Apparently, there were only selected works on developing an automated insect pest monitoring system with DSS capability. This calls for the development of a holistic system that can both monitor the population of insect pests and provide recommendations to the farmers (Böckmann et al., 2015).

2.4.2 Biophysical model development



A biophysical model is used to describe the behavior of biological systems through the use of mathematical models. In IPM, biophysical modelling is a helpful tool for knowing the possible influences of the environment, control methods, and other external factors so that farmers can plan insect pest management strategies more effectively.

It was shown in recent studies that one of the factors that has the most influence to insect pest behavior is temperature. Temperature largely affects the reproduction, development, and flight activity of insect pests (Damos & Savopoulou-Soultani, 2012; Haridas et al., 2016; Kingsolver et al., 2015). It was found by Bonsignore (2015) that whiteflies are most active at temperature levels 25°C to 30°C. It was also found that whiteflies had no activity at temperature levels below 12°C. On the other hand, aphids and thrips were most active at temperature levels from 28°C to 30°C (Ramalho et al., 2015) and 27°C (Rhains et al., 2007), respectively. It can be noticed that the three common greenhouse insect pests had close active temperature requirements. Meanwhile, studies show that outdoor insect pests had similar characteristics. According to a study by Wang et al. (2012), most fruit fly species had development and survival temperature requirements of 18°C to 30°C. Leafhoppers had a narrower range

of 30°C to 35°C, as found from an experiment by Rigamonti et al. (2014) performed in vineyards.



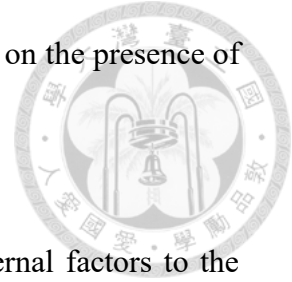
It was found in most of the related works that insect pests cannot survive under extreme levels of temperature. It was commonly seen that this phenomenon can be best described through skewed non-linear functions (Damos & Savopoulou-Soultani, 2012). Wang et al. (2012) found that the development rate of fruit flies appeared to slowly increase from 10°C to 30°C but immediately declined to zero probability at 34°C. They fitted the relationship using a non-linear extreme value model. In our previous work, a double Weibull function was used to describe the relationship between the increase in the whitefly count obtained by our insect pest monitoring system and the measured ambient temperature inside a greenhouse (Rustia & Lin, 2019). Through the fitted model, it was found that whiteflies drastically have a higher flight probability as it increases from 9°C to 26°C but reaches a sharp decrease at 27°C.

Differently, humidity was found to have less effect to the reproduction rate of insect pests as compared to temperature (Butler et al., 1988). It was found from recent studies that effects of humidity to the reproduction rate of insect pests were only found at extreme levels. Sengonca and Liu (1999) performed experiments analyzing the effects of different humidity levels to the reproduction rate of whiteflies. They were able to find out that at constant temperature and extreme humidity levels, the

development time of whiteflies increased from 20 days to 30-40 days. A study conducted by Yee (2013) showed a different phenomenon as some species of fruit fly emerged more frequently at high humidity levels. The study also shows that at extremely low humidity levels, fruit flies flew less frequently and moderately at normal levels of humidity.

Light, which has a direct relation to the time of day, was also found to have a direct effect to the flight activity of insect pests (Shimoda & Honda, 2013). In a study by Jha et al. (2009), it was shown that thrips have specific flight peak times of around 10:00 AM and 2:00 PM. It was found that thrips were generally inactive without the presence of light. Close findings were also found in the study of (Liang et al., 2010) about the flight behavior of thrips where it was found that thrips were most active at light intensities from 4000 lux to 6000 lux. On the other hand, whiteflies were more active during the morning from 6:00 AM to 9:00 AM and less active during the afternoon and at night (Butler et al., 1988). However, this was found different in the case of fruit flies. An experiment was conducted by Rieger et al. (2007) where the activity of fruit flies were compared during the morning and at night. In both cases, the results of the experiment show that fruit flies were more active when there was almost no presence of light, specifically at 10 lux. This means that outdoor in a fruit orchard, fruit flies tend to infest the crops more often at early dawn or late dusk. These works show that insect

pest have different diel flight activity and flying patterns depending on the presence of a light source.



The related works confirm the seasonal effects and other external factors to the over-all behavior of insect pests. It showed the importance of studying such relationships to be able to effectively control the population of insect pests. Unfortunately, most of the models developed were dependent on the datasets and methods of collections used. The model should be adaptive and applicable for different environments. The proposed automated monitoring system in this work can help in solving this problem by gathering on-field data that can be used to develop models that describe the insect pest behavior regardless of the location.

Chapter 3 Methodology



3.1 System overview

An integrated system called Intelligent and Integrated Pest and Disease Management (I^2PDM) System was developed in this research. A general overview of the I^2PDM system is illustrated in Fig. 3-1.

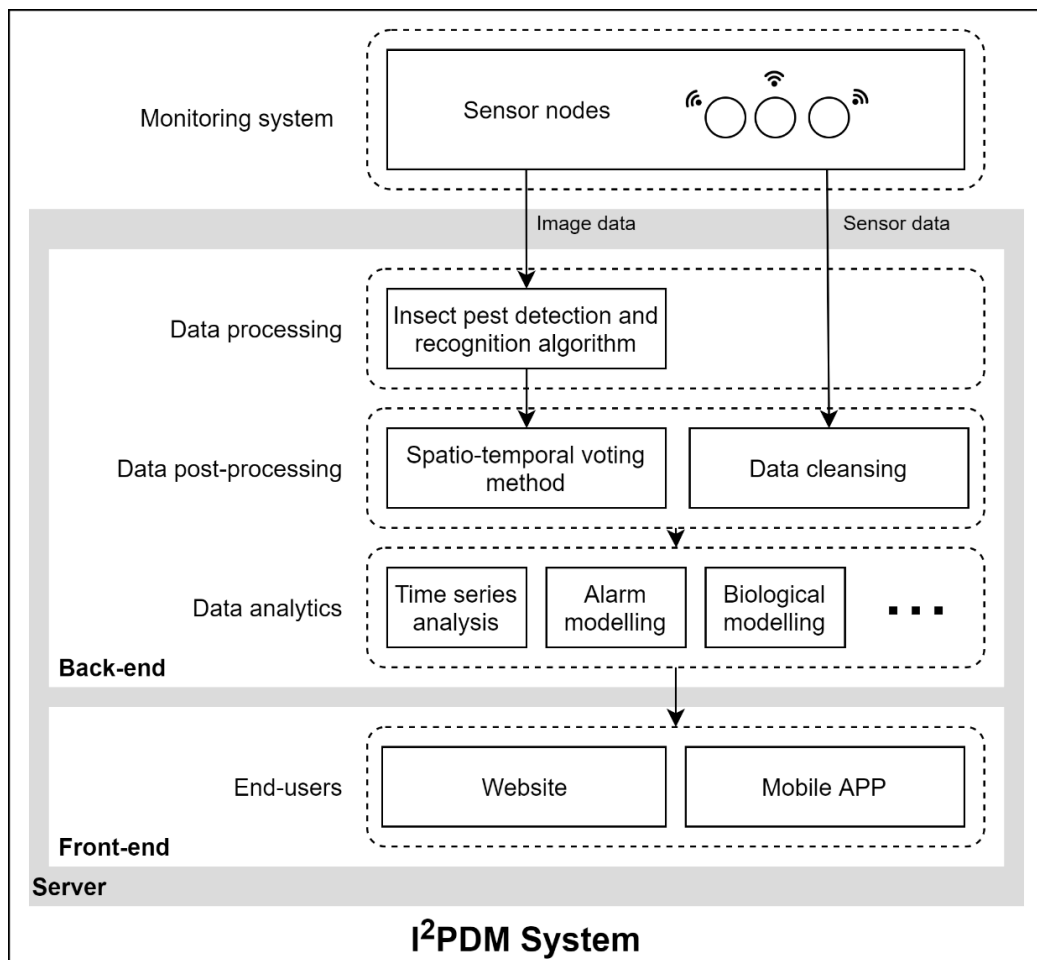
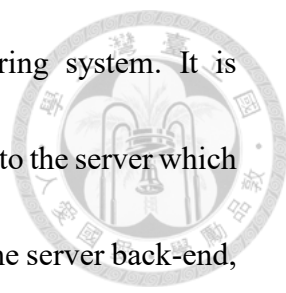


Fig. 3-1. General architecture of the I^2PDM system.



The main component of the I²PDM system is the monitoring system. It is responsible for collecting and sending data from the installation sites to the server which include sticky paper trap images and environmental sensor data. In the server back-end, the sticky paper trap images were processed through an insect pest detection and recognition algorithm while a spatio-temporal voting method was applied for post-processing. The resulting insect pest counts and environmental data were analyzed using data analytics which include methods such as time series analysis, alarm modelling, biological modelling, and more. Finally, end-users accessed their data and information in the server front-end via website or mobile APP. Technical considerations and discussions regarding the components of the I²PDM system and their corresponding sub-components are discussed in the next sections.

3.2 Monitoring system

The monitoring system is composed of sensor nodes that automatically acquire images of sticky paper traps and measure relevant environmental conditions. Indoor and outdoor versions of the sensor node were designed to accommodate different agricultural production setups. It realizes the monitoring component of IPM; to collect quantitative data that can be used to extract information from the insect pest condition

in a farm. This section discusses about the hardware design and development of the sensor node.



3.2.1 Indoor sensor node

The indoor version of the sensor node is made up of a Raspberry Pi 4 single-board computer, Raspberry Pi v2 camera, SHT20 temperature and humidity sensor, and BH1750 light intensity sensor, as illustrated in the block diagram in Fig. 3-2.

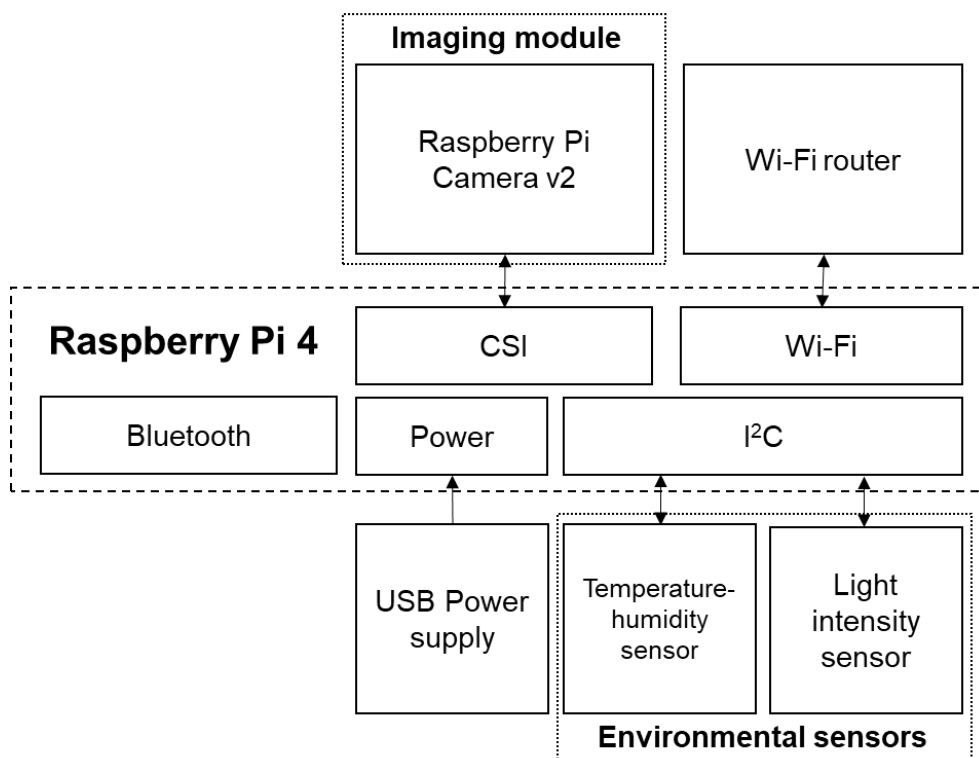
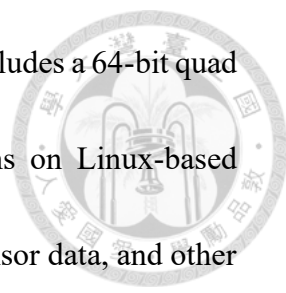


Fig. 3-2. Indoor sensor node functional block diagram.



Raspberry Pi 4 is an ARM-based single-board computer that includes a 64-bit quad core processor with a clock frequency of up to 1.4 GHz. It runs on Linux-based operating systems and is capable of acquiring images, collecting sensor data, and other low memory processing operations. The Raspberry Pi v2 camera, an 8-megapixel RGB camera with a maximum resolution of 3280 x 2464 pixels, was connected to the Raspberry Pi 4 via camera serial interface (CSI). The environmental sensors, SHT20 and BH1750, were connected via I²C interface. The device was powered by a 5V 3A USB power supply connected to an AC power source. An actual setup of the sensor node in a greenhouse is shown in Fig. 3-3.

The sensor node was enclosed in a plastic box. An arm made out of 1 mm stainless steel was screwed to the box for holding the sticky paper trap in place. An A5-sized (14.8 x 21.0 cm) sticky paper trap was placed over a flat board and secured by transparent clips to flatten the trap, assuring clear acquisition of the sticky paper trap image. The camera was protected by a glass cover and was positioned approximately 8 cm away from the sticky paper trap, allowing a field of view covering 11.5 cm x 15 cm of the sticky paper trap. Based on initial testing, it was found to be the best image acquisition setup that can maximize the number of insects detected without sacrificing the quality of the images. The 3D schematic diagrams illustrating the image acquisition setup of the sensor node is shown in Fig. 3.4.

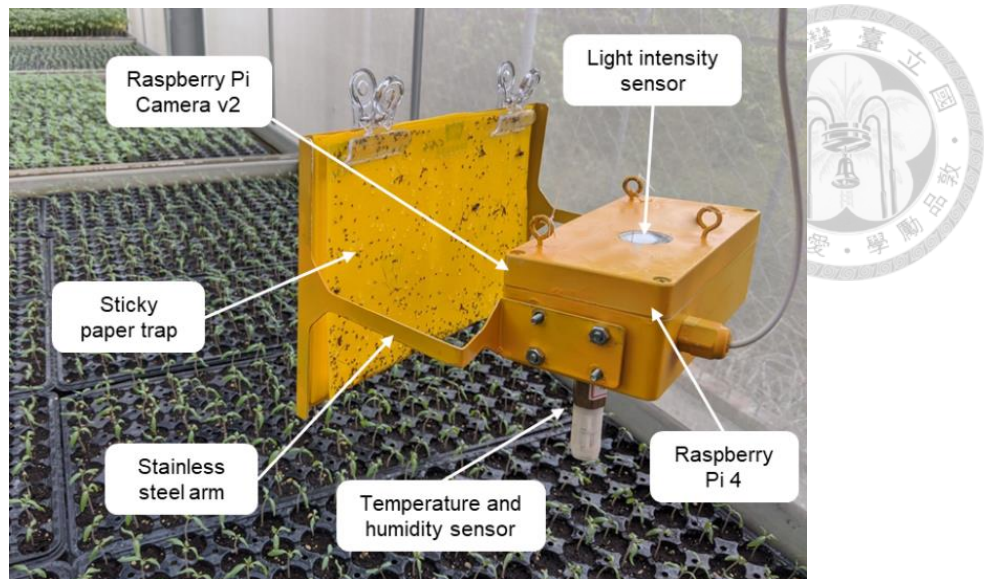


Fig. 3-3. Actual setup of the sensor node in a greenhouse.

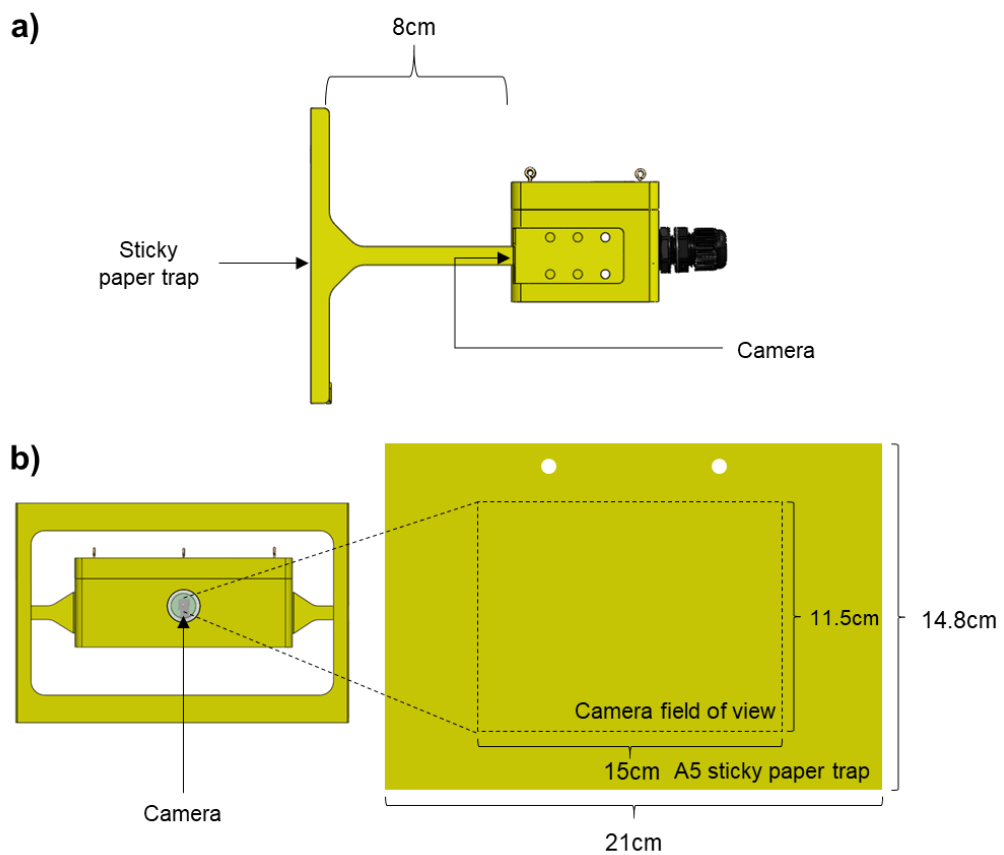


Fig. 3-4. 3D schematic diagram of the image acquisition setup of the sensor node. a)

Distance from camera to sticky paper trap; b) Camera field of view.

The temperature humidity sensor was installed below the box to minimize the effects dealt by sunlight to the sensor readings. The light intensity sensor was screwed to a PCB board attached to the Raspberry Pi 4 board, facing the top of the box, while a glass covers it. A 3 mm red LED indicator was installed below the box to show whether the device was on, sending data, or ready for device configuration via Bluetooth. Placements of the sensors and the LED is illustrated in a 3D schematic diagram in Fig. 3-5, while the specifications of the environmental sensors are shown in Table 3-1.

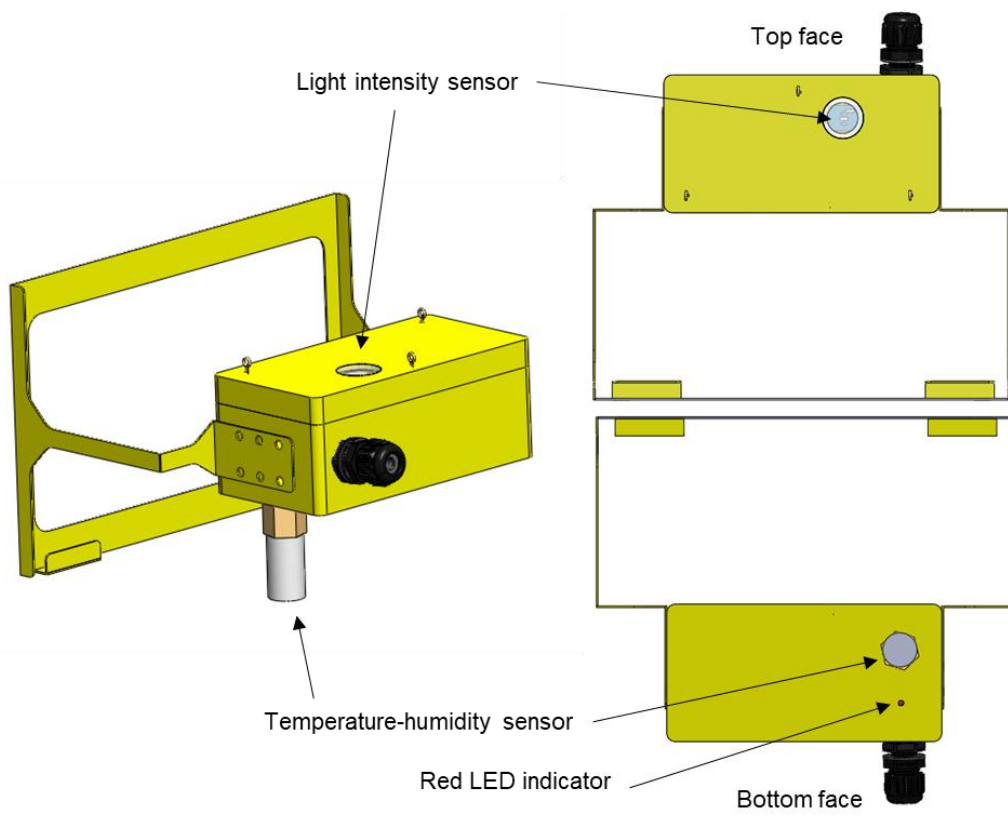


Fig. 3-5. 3D schematic diagram of the sensor node showing the sensor locations.

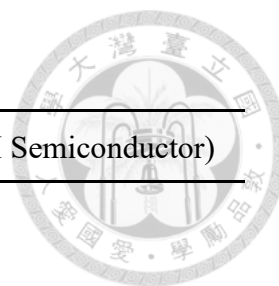


Table 3-1. Specifications of the environmental sensors.

Specification	SHT20 (Sensirion)	BH1750 (ROHM Semiconductor)
Resolution	Temperature: 0.01°C Relative humidity: 0.04%	1 lux
Accuracy	Temperature: $\pm 0.3^\circ\text{C}$ Relative humidity: $\pm 3.0\%$	1.2 (Sensor out/Actual lux)
Range	Temperature: -40 to 125°C Relative humidity: 0 to 100%	1-65535 lux

The sensor node was installed in an indoor environment by hanging it using a carabiner hooked to a rope or chain, 8-10 cm above the plants. The distance of the device and the plants was determined based on the height of the crops as they grow.

The installation setup is illustrated in Fig. 3-6.

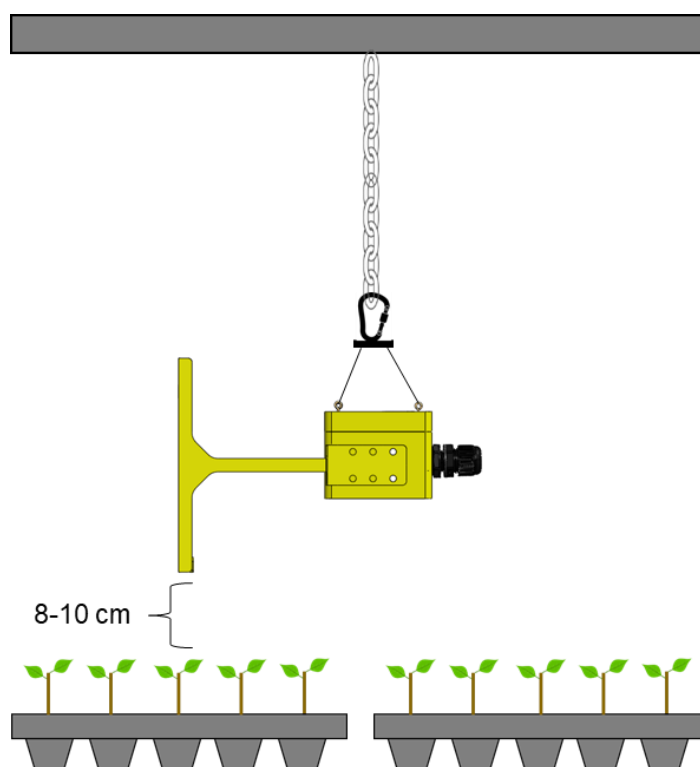
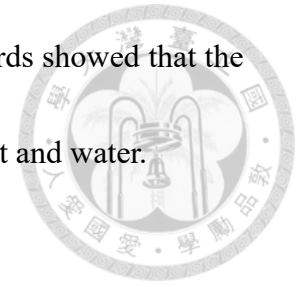


Fig. 3-6. Schematic diagram of the indoor installation setup of the sensor node.

Tests performed based on the Ingress Protection Rating standards showed that the sensor node can be rated up to IP66 in terms of protection from dust and water.



3.2.2 Outdoor sensor node

The outdoor sensor node performs similar functions as the indoor sensor node but focusing on minimizing its power consumption. Raspberry Pi Zero was used as the processing unit instead of Raspberry Pi 4 since it has less power consumption with a lower maximum clock frequency of up to 1GHz. Raspberry Pi Zero consumes a maximum of 0.85W (5V, 170mA) while Raspberry Pi 4 consumes 2.7W (5V, 510mA).

The functional block diagram of the outdoor sensor node is shown in Fig. 3-7.

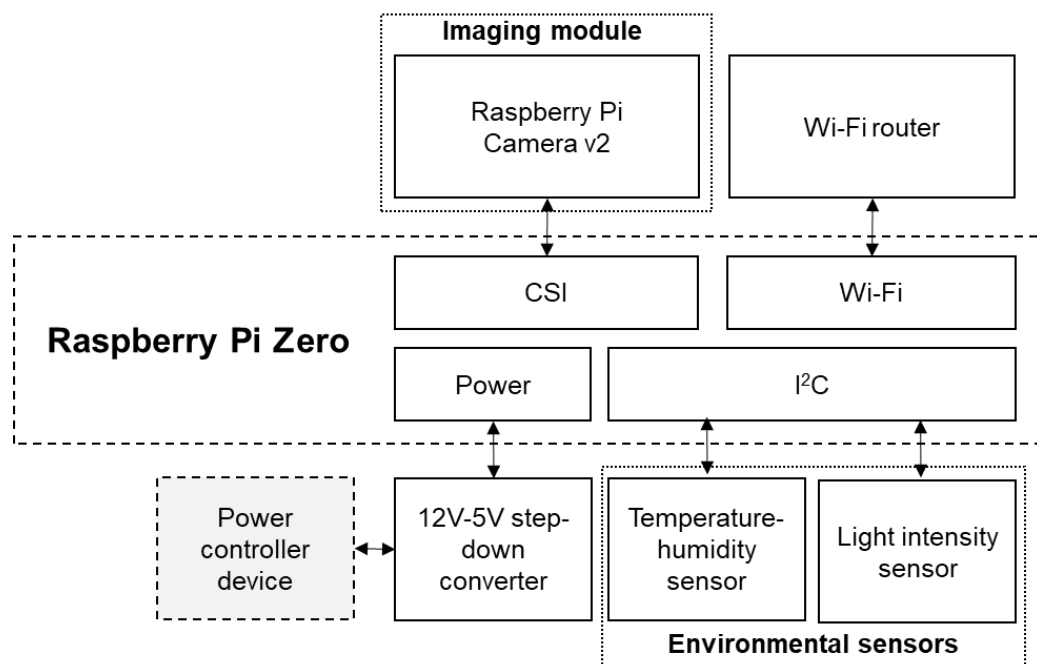


Fig. 3-7. Outdoor sensor node functional block diagram.

The outdoor version of the sensor node had a similar dustproof and waterproof hardware design as the indoor version. In contrast, the stainless-steel arm holder included a roofing that protects the sticky paper trap from rain droplets. The box was connected to a telescopic rod that can be extended up to 1.5 m in height. A water filled pole base, with a maximum weight of 10 kg, was used as the device stand that prevents the device from falling or be driven away by strong gusts of wind. The outdoor sensor node was designed in such manner so that the farm managers can move the device depending on their preferred location, height of the trees, and terrain. The actual device setup of the outdoor sensor node and its three-dimensional setup schematic diagram are shown in Fig. 3-8 and Fig. 3-9, respectively.

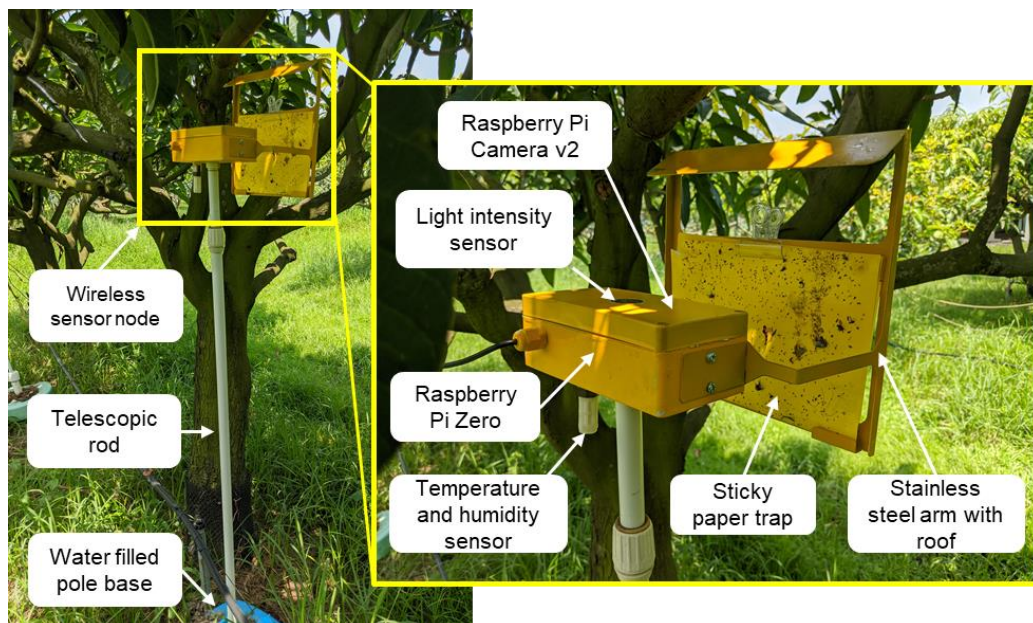


Fig. 3-8. Actual setup of the sensor node in an orchard.

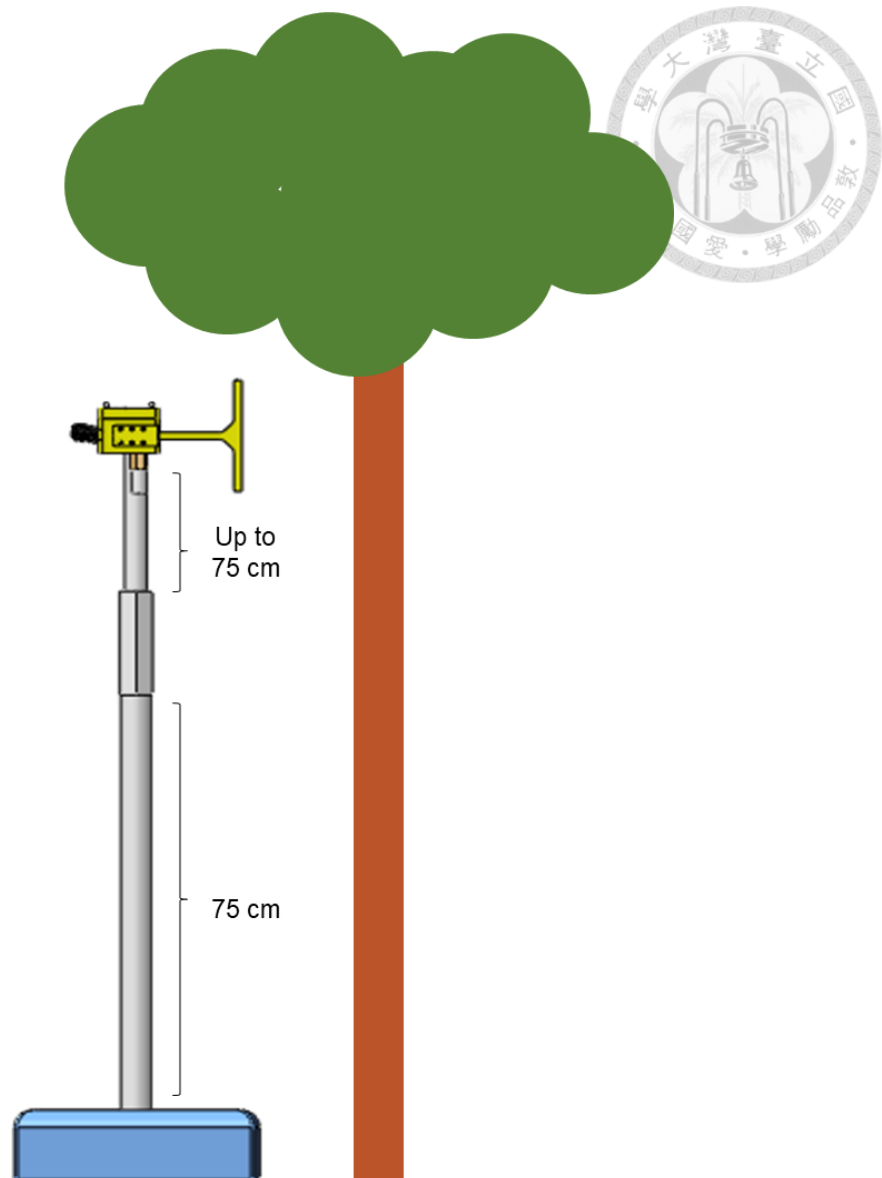


Fig. 3-9. Schematic diagram of the outdoor installation setup of the sensor node.

The power source of the outdoor sensor node came from a 12V 26Ah lead acid battery which was simultaneously charged by an 18V 100W solar panel connected to a solar charge controller module. The connection of the power source and the outdoor sensor node was controlled by a power controller device which includes a Raspberry Pi 4 and 5V latching relay. The power controller device was enclosed in an IP67 graded

waterproof and dustproof box and has the same set of power source as the outdoor sensor node. The power controller device triggered the relay that connects the power source of the outdoor sensor node based on pre-set time periods. Through this power control strategy, a single outdoor sensor node can last up to 5 days without being charged by the solar panel. Schematic diagram of the outdoor sensor node and the power controller device is shown in Fig. 3-10.

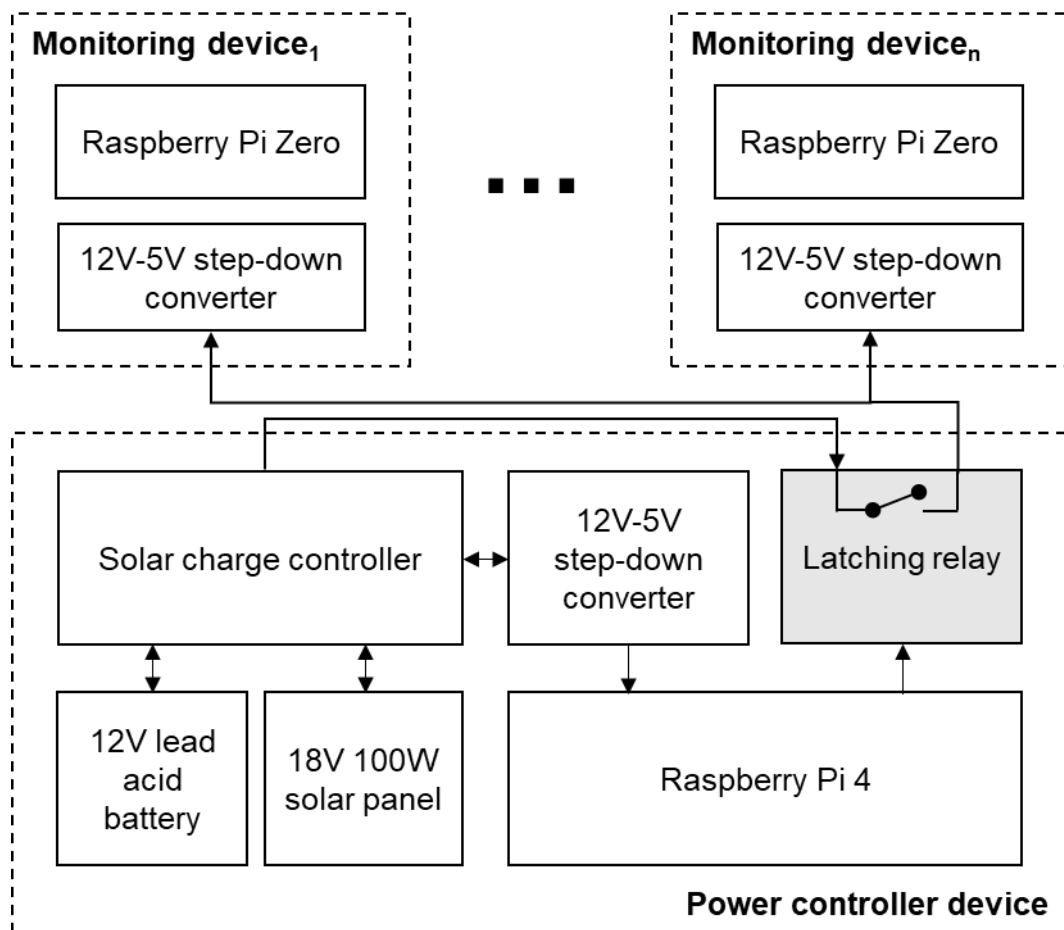


Fig. 3-10. Outdoor sensor node power control schematic diagram.



3.2.3 Network configuration

Each sensor node was wirelessly connected to the internet via Wi-Fi using different network topologies depending on the size of the installation site. In smaller sites, the system followed a star topology in which the sensor nodes were connected to a single internet-enabled Wi-Fi router, as shown in Fig. 3-11a. But in larger sites, a mesh topology was used similar to Fig. 3-11b which consisted of a single internet-enabled Wi-Fi router with several Wi-Fi signal extenders connected to it. Through mesh topology, the sensor nodes connected to the router with the best signal quality and strength. In sites where internet connection was unavailable, a 4G LTE router was used as main Wi-Fi router.

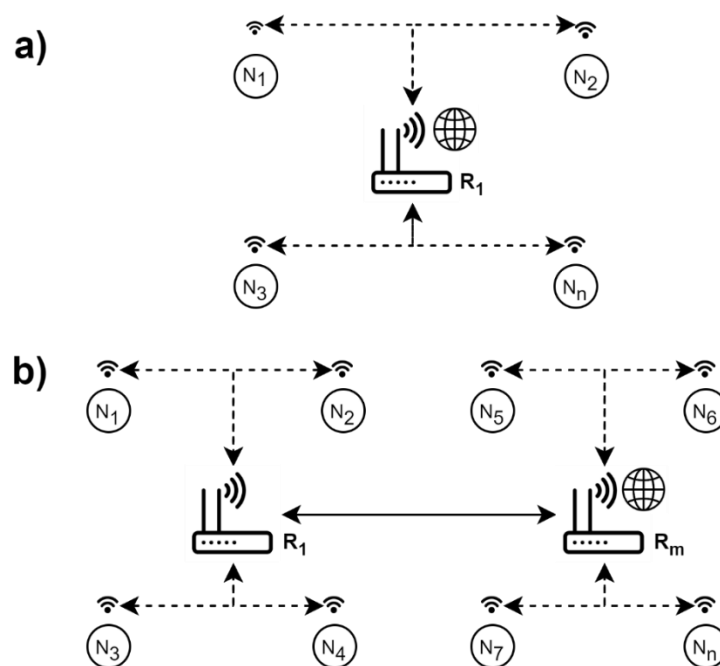


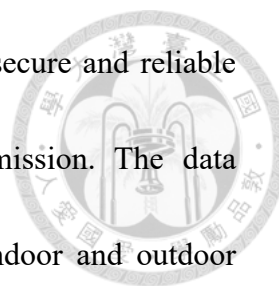
Fig. 3-11. Monitoring system network topologies: a) Star; b) Mesh.

3.2.4 Device configuration and data transmission



One essential step in IoT is to configure the identity of a device. Each sensor node had to be configured to determine its location and number. This was carried out by running a program in the sensor node that automatically turns on its Bluetooth and waits for a smart phone to pair with it. A user can use a smart phone and open the I²PDM mobile APP to scan for configurable sensor nodes. The user can input information such as the WiFi router SSID the sensor node will connect to and its password, location name, and number. The sensor node receives the configuration via Bluetooth serial interface and checks for program updates in Docker Hub, as described in more detail in the next section.

After receiving the updates, the sensor node registers itself to the server. The sensor node sends a registration packet, which includes the sensor node location and number, to the server via User Datagram Protocol (UDP). UDP is a low-latency communication protocol for sending small packets of data through the internet. A self-defined handshake protocol was developed to secure the throughput of data received and transmitted. After registration, it begins the data transmission routine. The environmental data such as temperature, humidity, and light intensity were sent via UDP. Meanwhile, the images were sent to the server via Transmission Control Protocol (TCP).



TCP is a slower sending protocol compared to UDP but is more secure and reliable since it also requests a response from the server after transmission. The data transmission frequency configuration of the sensor node in an indoor and outdoor environment is shown in Table 3-2.

The sensor node sends the environmental data less frequently outdoors. This was done to decrease its energy consumption. Moreover, the outdoor environment condition does not change as fast as the indoor environment. Indoors, the environmental data was measured in shorter intervals, since some farm owners apply several environmental control strategies. The image data was sent every hour since the number of insect pests detected does not change abruptly, as found from previous tests. The program ran by the sensor node was written using Python 3.5.

Table 3-2. Data transmission frequency of the sensor node.

Device	Data	Protocol	Frequency
Indoor	Temperature	UDP	Every 10 minutes
	Humidity		
	Light intensity		
	Image	TCP	Every hour from 5AM to 7PM
Outdoor	Temperature	UDP	Every hour
	Humidity		
	Light intensity		
	Image	TCP	Every hour from 5AM to 7PM



3.3 Server architecture

The server was responsible for storing all the data collected by the monitoring server and distributing it to the end-users. A block diagram illustrating the server architecture is shown in Fig. 3-12.

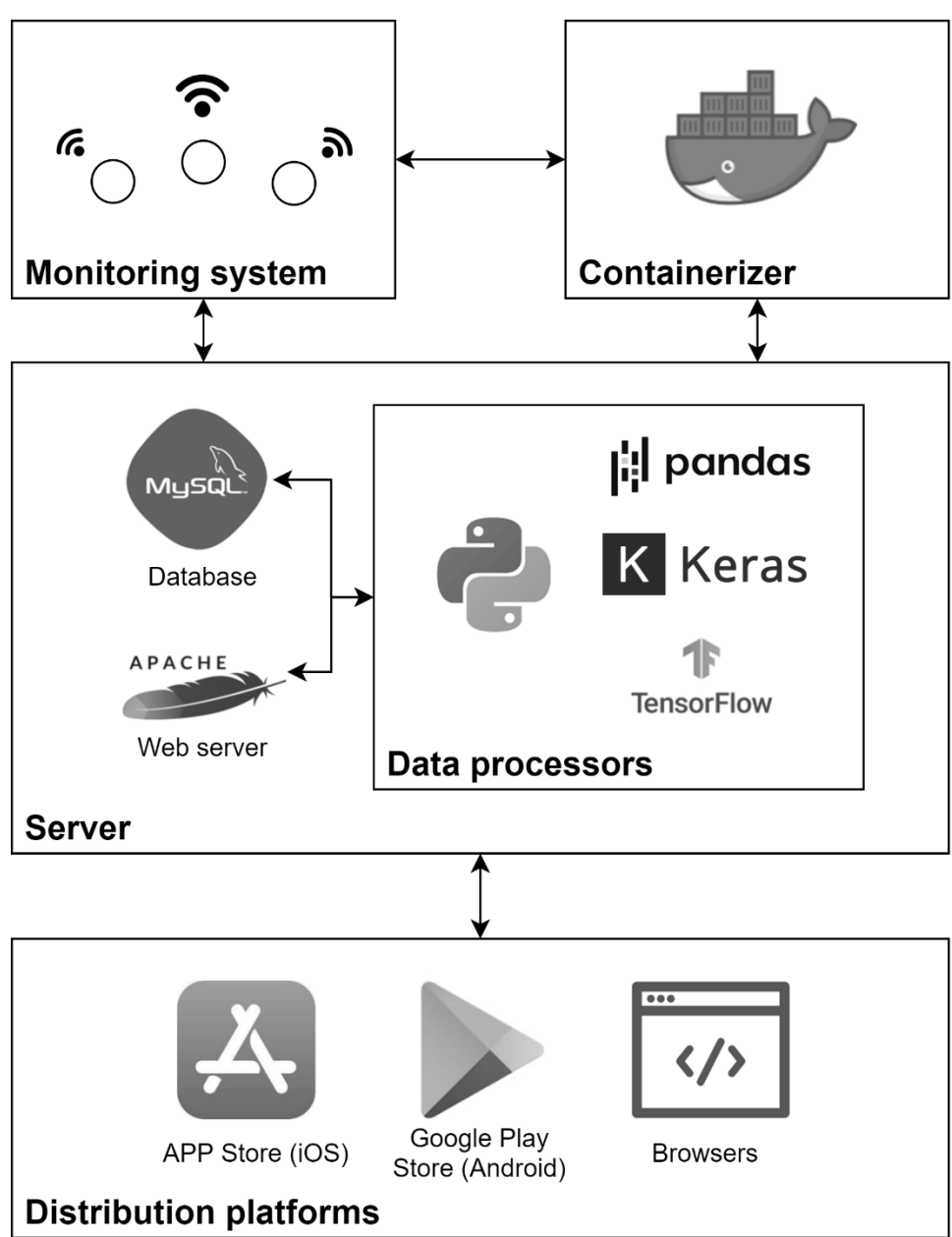


Fig. 3-12. Server architecture block diagram.

The environmental data, sensor node information, location information, user account information, etc. were saved in the MySQL database of the server. MySQL is an open-source relational database management system currently used by well-known software companies such as Facebook, Twitter, and YouTube. Meanwhile, image data were stored locally into the server. Data processors, written in Python 3.5 programming language, were used to pre-process, clean, and analyze the collected data.

The monitoring system follows a remote update protocol via Docker containerization. A Docker container was set-up using the server. A Docker container is a standalone software package that may include applications, codes, libraries, and files, to be run on another device. This prevents users from hacking or modifying the source codes of the sensor nodes as it was installed in a site. Most importantly, it allows the server to initiate a system update to the sensor nodes without accessing each node individually. This was done by pushing the Docker container to the Docker Hub, a public repository for storing Docker containers.

Finally, the system was distributed to the end-users via mobile APP and website. The mobile APP is downloadable in both APP store for iOS and Google Play Store for Android. Users may also access the website using any web browser. The server is running under Windows 10 operating system, with an Intel Core i7-7700 CPU @ 3.60 GHz, 16 GB RAM, and Nvidia GTX 1070 GPU.



3.4 Experimental sites

The system has been installed in 10 indoor sites and 1 outdoor site. The approximate geographical locations of the experimental sites are shown in Fig. 3-13 while basic information on each site is shown in Table 3-3.

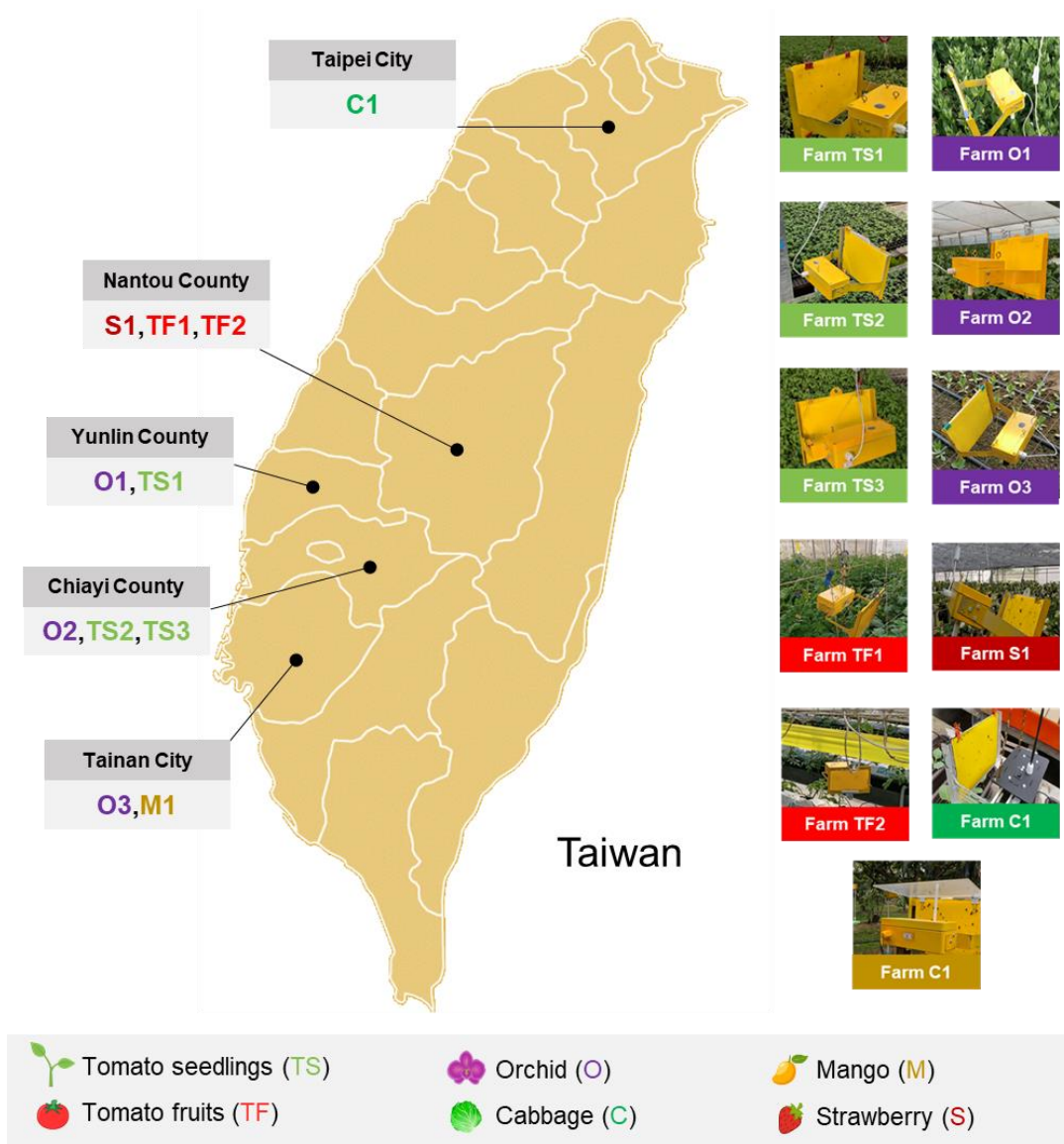
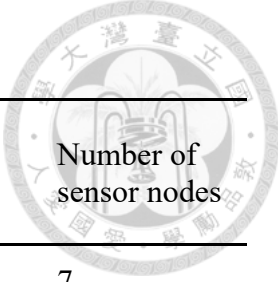


Fig. 3-13. Geographical locations of the experimental sites.

Table 3-3. Basic information on the installation sites.



Farm name	Geographical location	Land area (m ²)	Crops	Installation duration (days)	Number of sensor nodes
TS1	Chiayi, Taiwan	529	Tomato seedlings	1260	7
TS2	Yunlin, Taiwan	2208	Tomato seedlings	970	10
TS3	Yunlin, Taiwan	600	Tomato seedlings	382	6
T1	Nantou, Taiwan	630	Tomato	268	8
T2	Nantou, Taiwan	1673	Tomato	268	8
O1	Yunlin, Taiwan	550	<i>Eustoma</i>	851	8
O2	Tainan, Taiwan	500	<i>Phalaenopsis</i>	674	6
O3	Tainan, Taiwan	250	<i>Eustoma</i>	786	6
S1	Nantou, Taiwan	550	Strawberry	654	8
C1	Taipei, Taiwan	100	Cabbage	601	4
M1	Tainan, Taiwan	10000	Mango	645	8

The experimental sites were all independent of each other in terms of crops grown, management strategies, weather, etc. In each site, the number of installed devices vary depending on the farm area, insect pest management strategy of the farmers, and the number of suspected insect pest hotspots in the farm. Most of the crops grown by each farm are major economic crops in Taiwan (Lee et al., 2020). As shown previously in Table 2-1, major insect pests such as whiteflies, thrips, and aphids are attracted by the crops listed in Table 3-3. Farms O3, M1, and C1 were managed by our research group

and the Tainan District Agricultural Research and Extension Station (TDARES), Council of Agriculture, Taiwan ROC. The installation setup maps of each location are shown in Figs. 3-14 to 3-24.

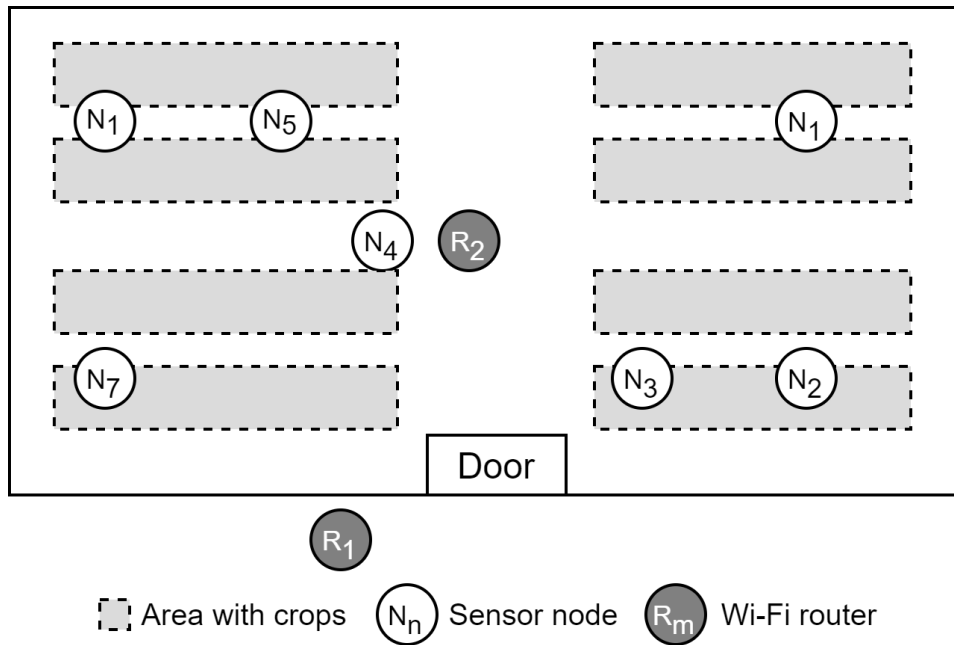
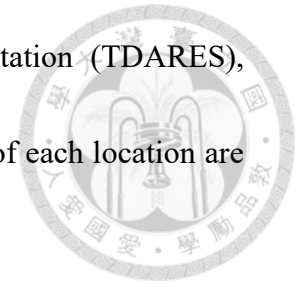


Fig. 3-14. Installation setup map of Farm TS1.

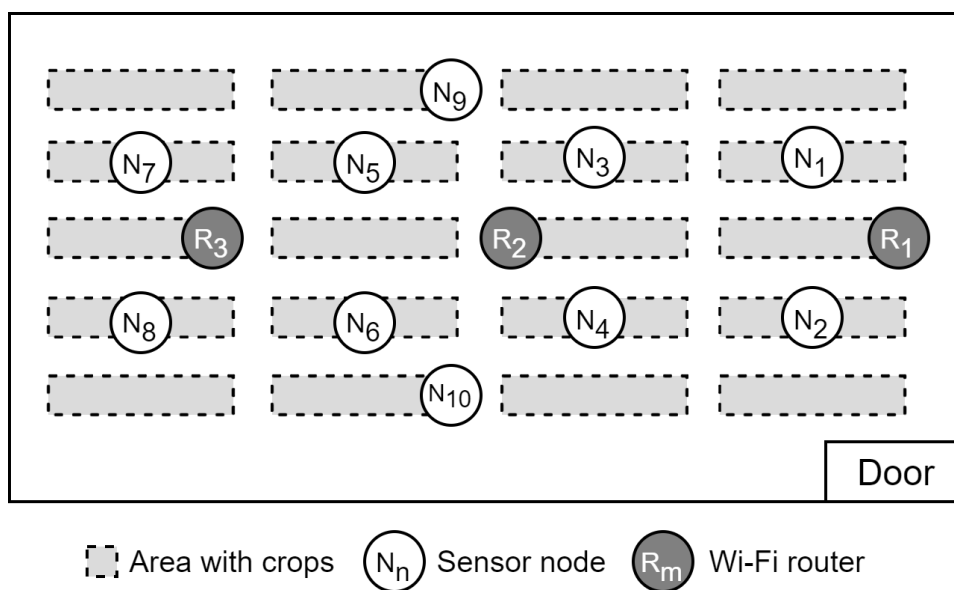


Fig. 3-15. Installation setup map of Farm TS2.

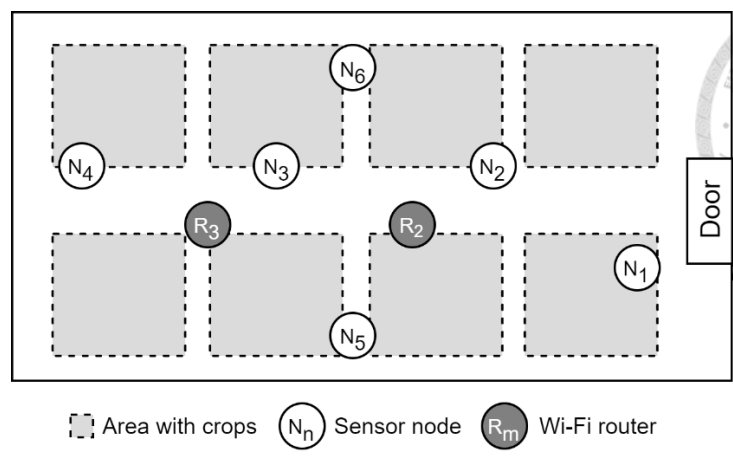
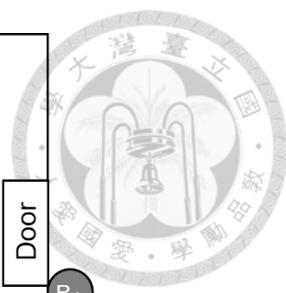


Fig. 3-16. Installation setup map of Farm TS3.

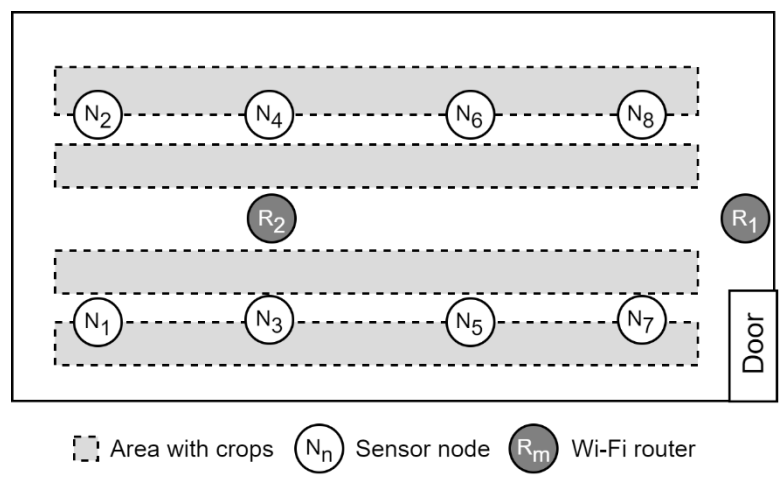


Fig. 3-17. Installation setup map of Farm T1.

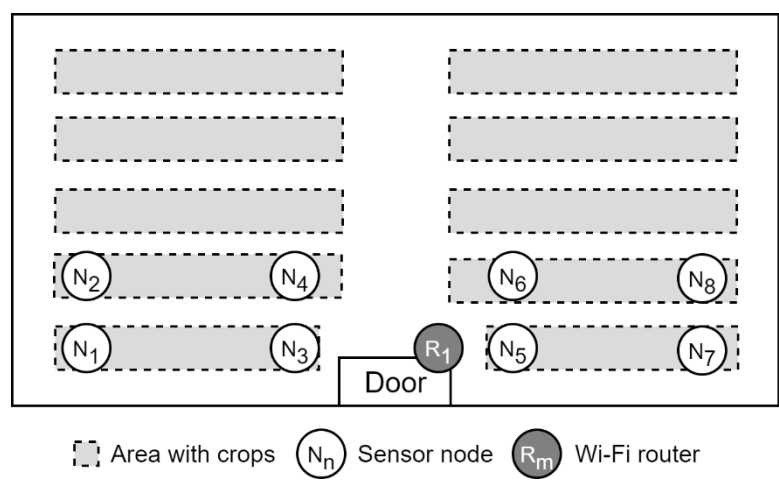


Fig. 3-18. Installation setup map of Farm T2.

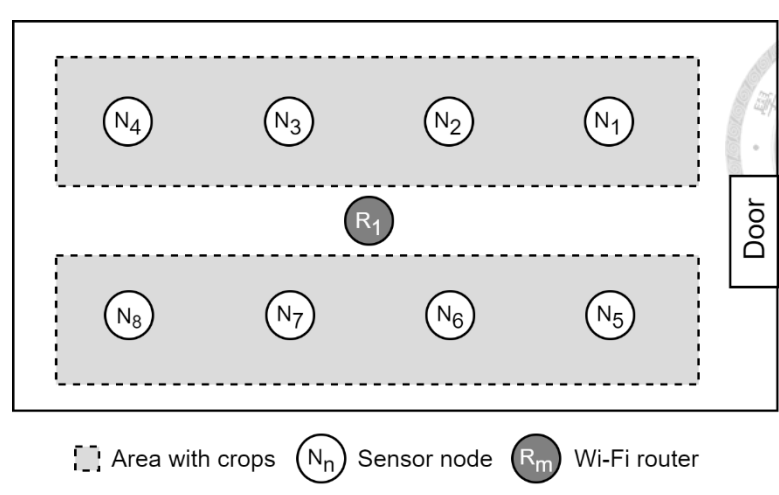
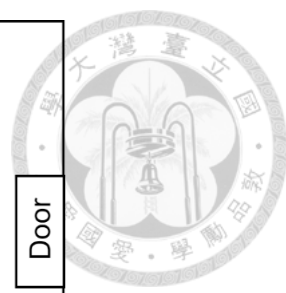


Fig. 3-19. Installation setup map of Farm O1.

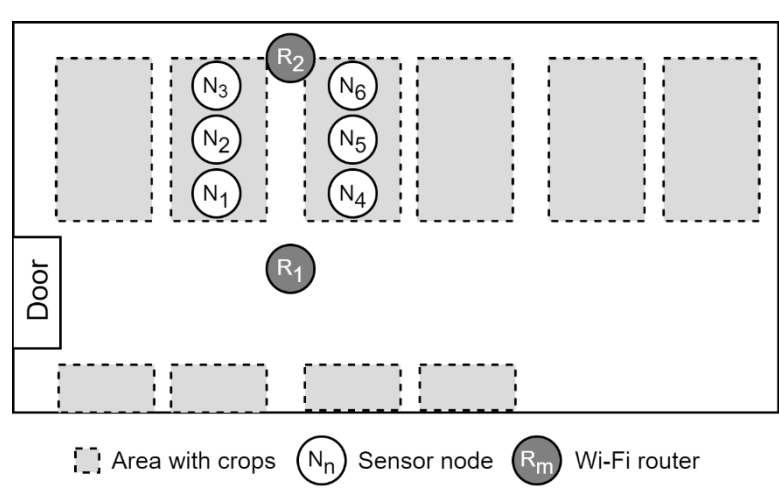


Fig. 3-20. Installation setup map of Farm O2.

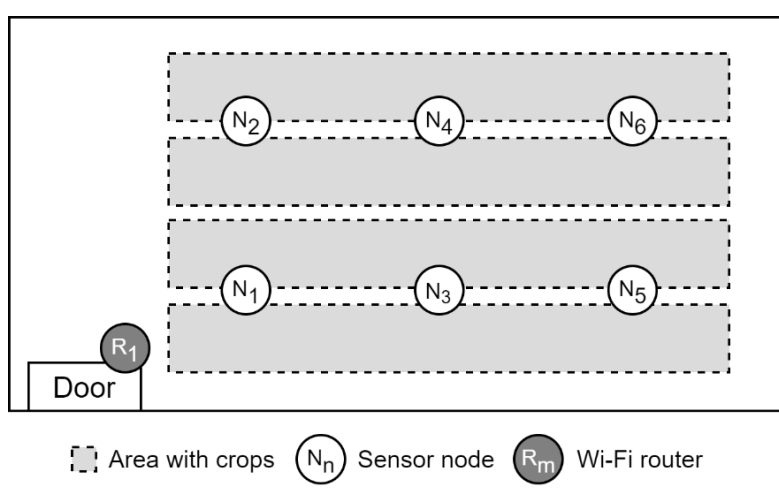
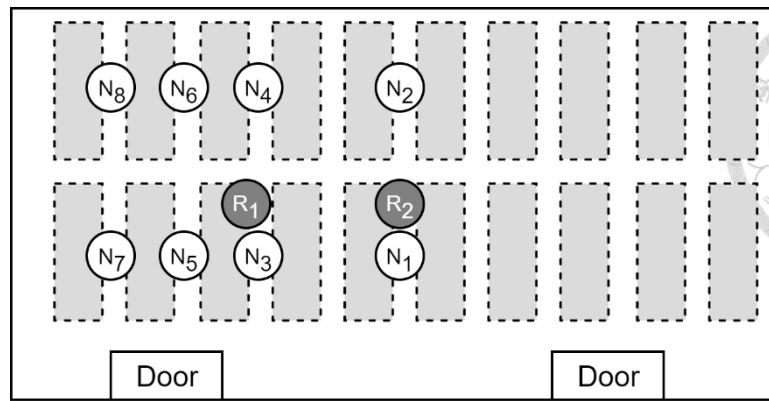
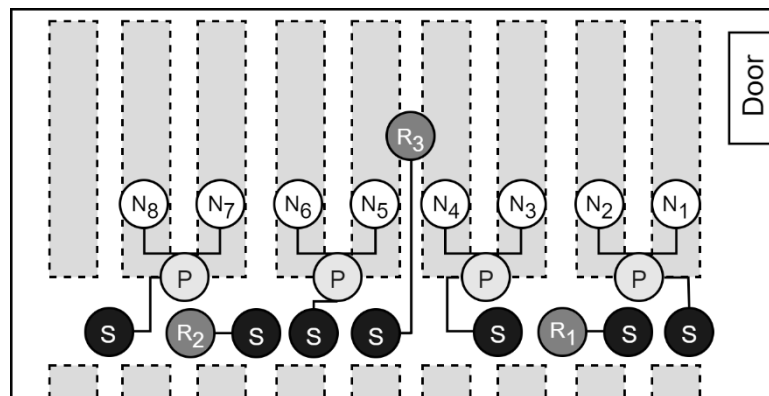


Fig. 3-21. Installation setup map of Farm O3



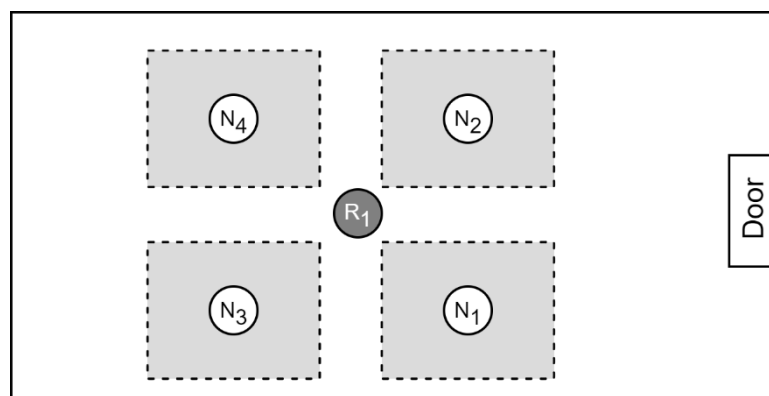
Area with crops
 Sensor node
 Wi-Fi router

Fig. 3-22. Installation setup map of Farm S1.



Area with crops
 Sensor node
 Wi-Fi router
 Power controller device
 Solar panel

Fig. 3-23. Installation setup map of Farm M1.



Area with crops
 Sensor node
 Wi-Fi router

Fig. 3-24. Installation setup map of Farm C1.

3.5 Insect pest detection and recognition algorithm



The images were acquired by the monitoring system and were sent via internet to the server for batch processing. Each image was processed with an insect pest detection and recognition algorithm composed of three major stages: object detection, insect vs. non-insect classification, and multi-class insect classification. An overview of the algorithm is shown in Fig. 3-25.

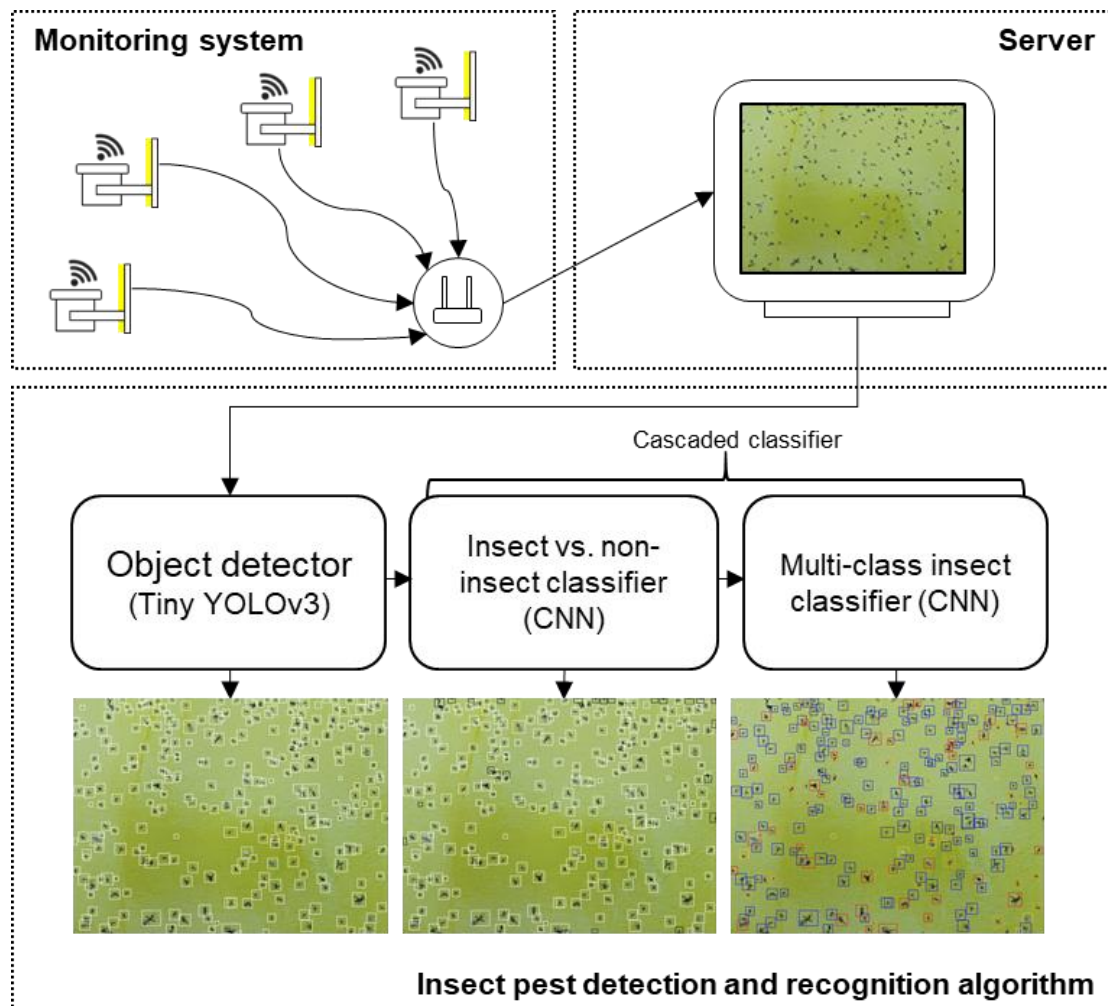
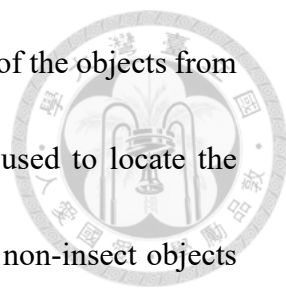


Fig. 3-25. Insect pest detection and recognition algorithm flowchart.



First, the object detector obtains the bounding box coordinates of the objects from the sticky paper trap image. The bounding box coordinates were used to locate the objects to be later classified as insect or non-insect. The identified non-insect objects were ignored and the insects were classified into their corresponding type. The main advantage of the presented multi-stage approach is that additional classes may be easily added to the multi-class insect classifier if necessary. For instance, a next classification stage can be added after classifying the flies by reclassifying the flies to the species level. The presented method is a more convenient and modular approach rather than a one-step detection and classification method; it can even be faster and more accurate since each part of the algorithm can be finely optimized. The algorithm was written using Python 3.5 with the support of OpenCV Deep Neural Network (DNN) library (Bradski, 2000), Tensorflow GPU v1.11 (Abadi et al., 2016), and Keras v2.2.4 deep learning libraries. The next sub-sections describe the technical considerations and the processes involved in each stage.

3.5.1 Object detector

Based on previous studies involving automatic insect pest counting, one source of error was the influence of external factors such as varying lighting condition and the

presence of non-insect objects in the image (Wen & Guyer, 2012; Zhong et al., 2018).

To reduce the errors produced by such factors, a CNN-based object detector was used in this work. Compared to traditional segmentation methods, such as background subtraction and color clustering, CNN-based object detectors are less affected by object scale, color, and size. This was demonstrated in the study by Zhong et al. (2018), in which connected components labeling (CCL) was compared with a You Only Look Once (YOLO) object detection model in detecting flying insects. Their work demonstrated that using YOLO had a better counting accuracy compared to CCL.

The objects from the sticky paper trap images were detected using a series of operations, as shown in Fig. 3-26. First, the original 3280 x 2464 RGB image was resized to 3200 x 2400 using cubic interpolation to make it separable into equal parts. Then, the resized image was tiled into 12 images of 880 x 880 resolution. Tiling is a technique in object detection that optimizes the detection accuracy of the object detector by dividing a high-resolution image into several parts to reduce the size of the object detector input image. This is often done especially for detecting small objects (Huang, 2019; Unel et al., 2019). The image tiling resolution of 880 x 880 was set so that the resizing ratio is only halved before each image was used as an input to the object detector, avoiding excessive image distortion. After tiling, the 880 x 880 images were resized again into 416 x 416 images as individual inputs to the object detector model.

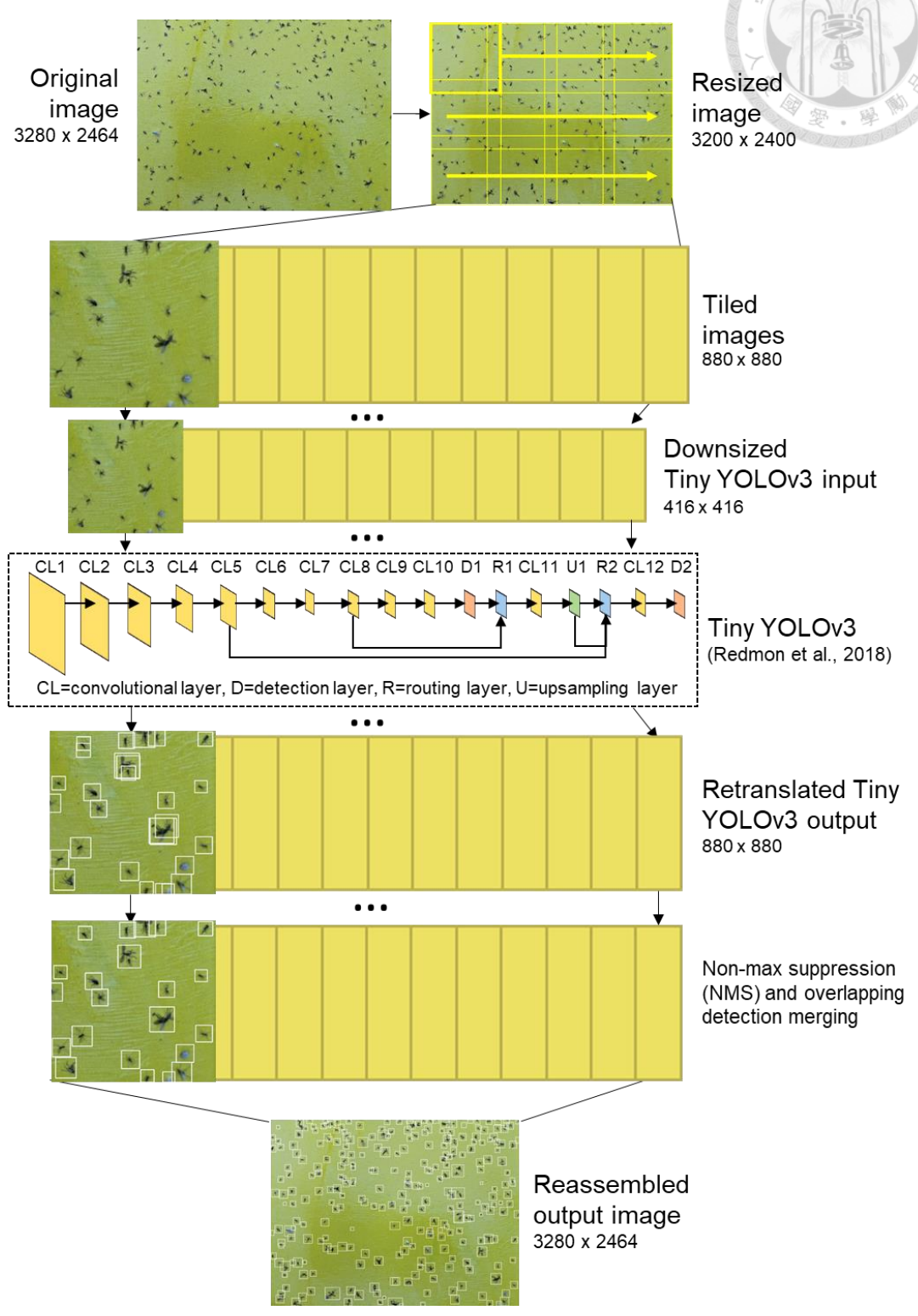
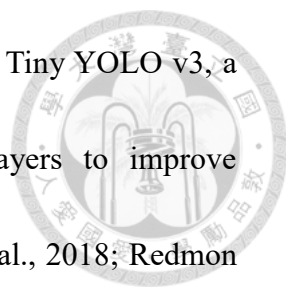


Fig. 3-26. Object detector model flowchart.



The object detector model used in this research was based on Tiny YOLO v3, a smaller version of YOLO v3, that uses less convolutional layers to improve computation speed and is adapted for less complex images (Ma et al., 2018; Redmon & Farhadi, 2018; Yi et al., 2019). Tiny YOLO v3 was chosen instead of YOLO v3 since it was found from prior testing that both models did not have too different performances in detecting the objects in the sticky paper trap images. The possible reason for this was the sticky paper trap images had a simple and fixed background of yellow color; thus a very deep or complex object detector was found unnecessary. It detects the objects from the image by splitting the image into a grid and generating bounding boxes with each object's corresponding class probabilities. The output bounding boxes were retranslated to fit the scale of the original 880 x 880 tiled images. Non-max suppression (NMS) was applied on each tiled image to remove the excess candidate bounding boxes using NMS thresholding with values ranging from 0 to 1.0, where values close to 1.0 lead to more retained bounding boxes and fewer boxes were retained with values closer to 0. Similar overlapping objects were merged into single objects using Intersection over Union IoU (Ding & Taylor, 2016), using Eq. 3-1:

$$Intersection\ over\ union\ (IoU) = \frac{area(B_1 \cap B_2 \cap \dots B_o)}{area(B_1 \cup B_2 \cup \dots B_o)} \quad (3-1)$$

where B_o is the bounding box coordinates of the object with o as the object index. Each B_o contains four coordinates: x_1, y_1, x_2, y_2 of the object where x_1 and y_1 belong to the x y vertex coordinates of the box and x_2 and y_2 belong to the vertex opposite to x_1 and y_1 .

IoU measures the area of overlap over the area of union of several objects based on their bounding box coordinates. *IoU* is measured from 0 to 1.0, where values close to 1.0 have highly matched box coordinates. Thus, if the overlapping objects have *IoU* values greater than or equal to 0.95, only a single set of coordinates was retained among the objects. In this work, only a single class was defined as the target for detection, the object class. The object class includes all insect objects found on the sticky paper traps. However, some of the objects may be non-insect objects. This problem was solved later on in the insect vs. non-insect object classifier stage.

3.5.2 Cascaded multi-class insect classifiers

Using the box coordinates obtained from the object detector, the images were cropped out from the original image for classification. The images were resized into 128 x 128 by cubic interpolation, which was found to be the average size of the insects found from the sticky paper trap images; minimizing image distortion. Objects with cropping coordinates of $m \times n$ sizes were made equal by following the larger distance

between the box length and width such as cropping the images with $n \times n$ sizes. The cropped images were used as input to the CNN image classifier, as shown in Fig. 3-27.

The CNN image classifier architecture shown in Fig. 3-27 is a sequential neural network used for general image classification and fitted for low-resolution images. Three convolutional layers (CL) with a single max pooling layer on each end are used to extract low (CL1), medium (CL2), and high (CL3) level features from the raw RGB pixel values of the 128 x 128 input image. Deep features were obtained from the previously extracted features by the fully connected layer (FCL) using rectified linear units (ReLU) as activation function. The softmax layer (SL) outputs the prediction probability for each defined class. The prediction probabilities were used to determine the class of the image based on a pre-defined classification threshold.

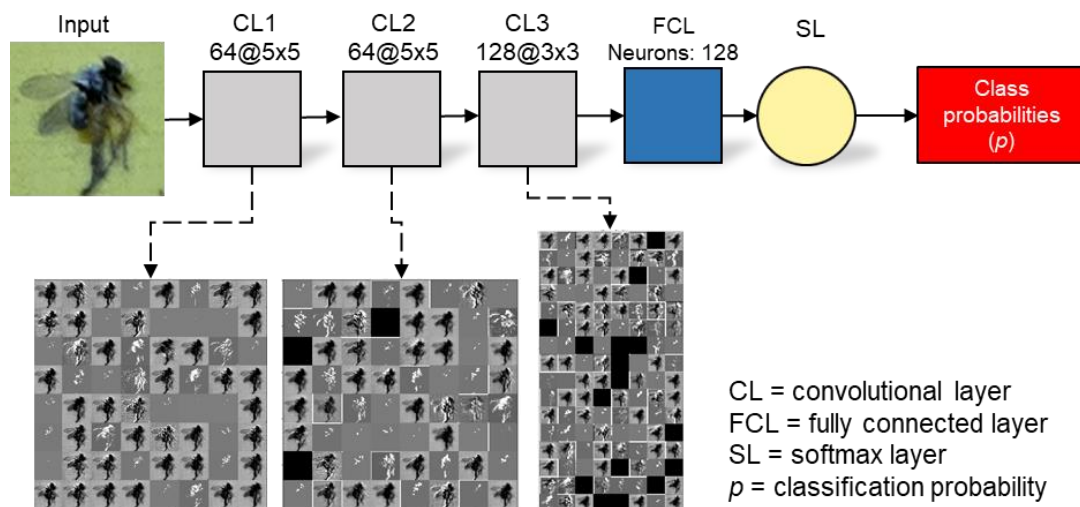
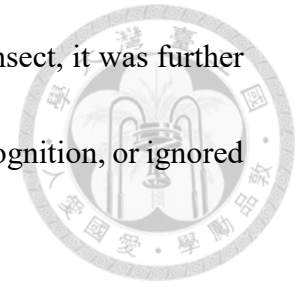


Fig. 3-27. CNN image classifier architecture.

Prior to this study, several network architectures, such as MobileNet and Inception v3, were tested and it was found that the CNN classifier in Fig. 3-27 exceeds the performance of the mentioned network architectures (Rustia et al., 2018). Other works also show that low to medium resolution images do not require a very complex image classifier (Cai et al., 2019a; Chevalier et al., 2016). Thus, the presented classifier structure was adequate for the application, most especially in consideration of complexity and computational cost.

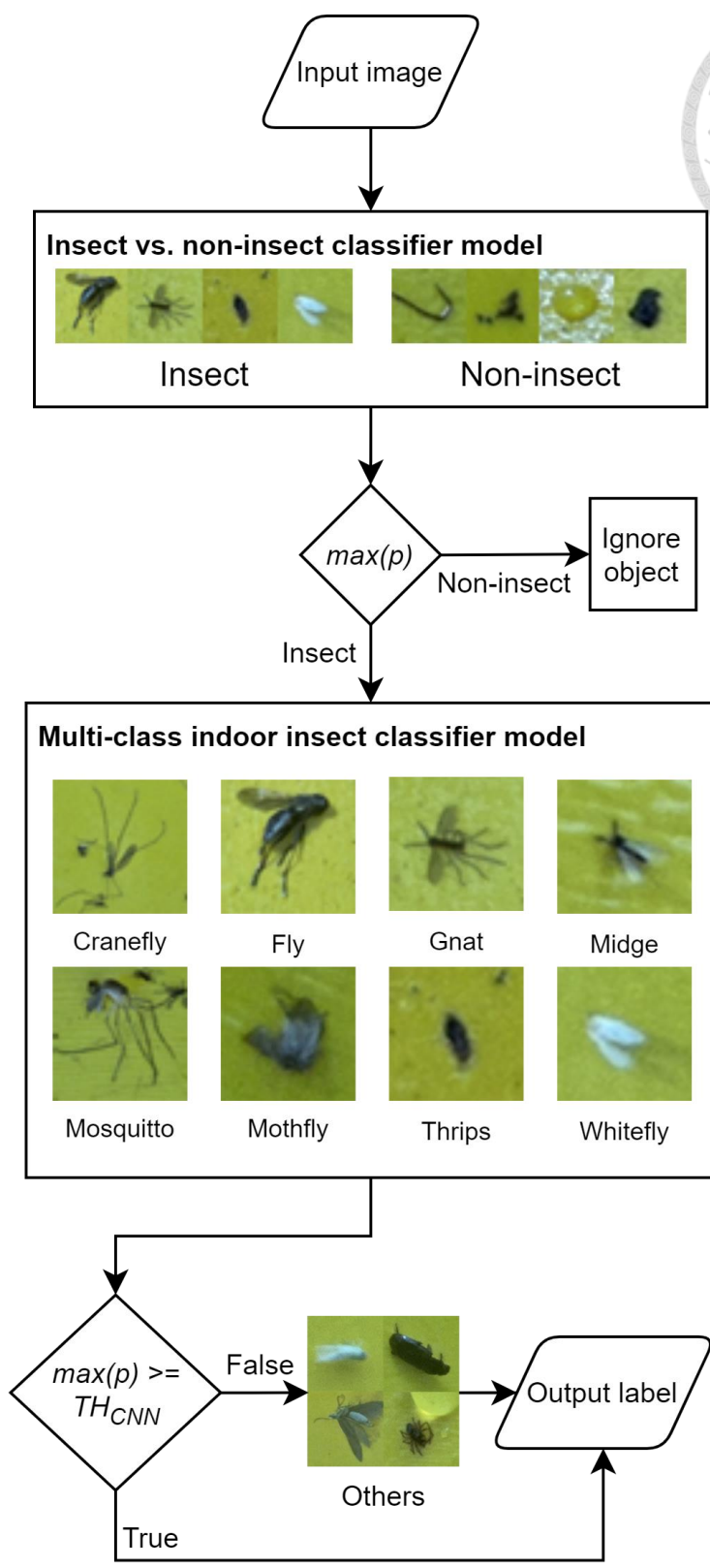
A cascaded approach in classification was implemented in this research. The classification output of each CNN classifier shown in Fig. 3-27 was sequentially connected to recognize an image from a general class to a specific class. This approach was inspired by how insects are taxonomically classified. The first model of the cascaded insect classifier is the insect vs. non-insect model. The insect vs. non-insect model classifies the objects detected by the object detector into two classes: insect or non-insect. The non-insect objects include dirt, water droplets, or glare. The non-insect objects were occasionally stuck on the sticky paper traps due to operations conducted inside the greenhouse or to the weather. Prior to this work, it was found that most misclassified cases are due to the non-insect objects and thus should be filtered out to optimize the algorithm efficiency (Rustia et al., 2020a; Rustia et al., 2020b). The output class of the insect vs. non-insect model was based on the highest classification

probability p . Thus, if there was a higher p that the object was an insect, it was further classified later on with the multi-class insect classifier for insect recognition, or ignored otherwise.



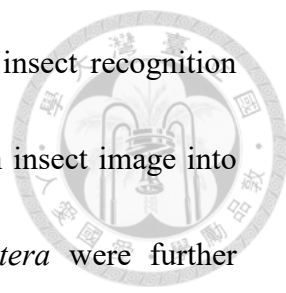
Two different combinations of multi-class insect classifiers were developed for insects found in indoor sites and outdoor sites. Using different classifiers optimizes the classification performance for the two type of sites since it was found that the insects found indoors and outdoors were different. Combining the samples from the two types of sites only caused more false positive errors in classification.

The multi-class indoor insect classifier consists of a single insect recognition stage with 8 types of insects: crane fly, fly, gnat, midge, mosquito, moth fly, thrips, and whitefly, as shown in Fig. 3-28. The insects shown in Fig. 3-28 were the most commonly found insects in the sticky paper images of the indoor sites. The output class of the multi-class insect classifier model was also based on the highest output p . A classification probability threshold TH_{CNN} was used to screen the classified insect. Insect objects classified with the highest p lower than the set TH_{CNN} were classified into others (Geng et al., 2018; Jain et al., 2014). Some of the other insects found in the greenhouses were beetles and spiders. However, such insects cannot fly, rarely seen, and do not inflict considerable damage to the crops.



p = classification probability
 TH_{CNN} = classification probability threshold

Fig. 3-28. Multi-class indoor insect classifier flowchart.



Meanwhile, the multi-class outdoor insect classifier has two insect recognition stages as shown in Fig. 3-29. The first stage, Stage 1, classifies an insect image into *Diptera*, *Cicadellidae*, or thrips. The images classified as *Diptera* were further classified by the Stage 2A model into fly, gnat, or mothfly, which were under the insect order *Diptera*. Similarly, images classified as *Cicadellidae* were re-classified by the Stage 2B model into mango leafhopper or leafhopper. Lastly, images labelled as fly were re-classified into fruitfly or oriental fruitfly in the Stage 3 model. This shows the advantage of the presented cascaded classifier approach as it can further classify insects into more specific insect types. This approach also enhances the performance of the entire cascaded classifier since it initially groups the insect images based on their taxonomic order, which also matches their physical appearances.

Both the indoor and outdoor insect classifiers face the problem of open set recognition. Open set recognition refers to the classification of images that are outside the scope of the training images of a classifier model. One solution in open set recognition is to train a classifier model with a negative class, such as an others class, to avoid misclassification (Bendale & Boult, 2016; Zhong et al., 2018). Unfortunately, it only caused confusion to the classifier in this case. In particular, adding the others class reduces the performance of the classifier in identifying the target insect types that are more relevant than the other insects. This occurs since some of the other insect types

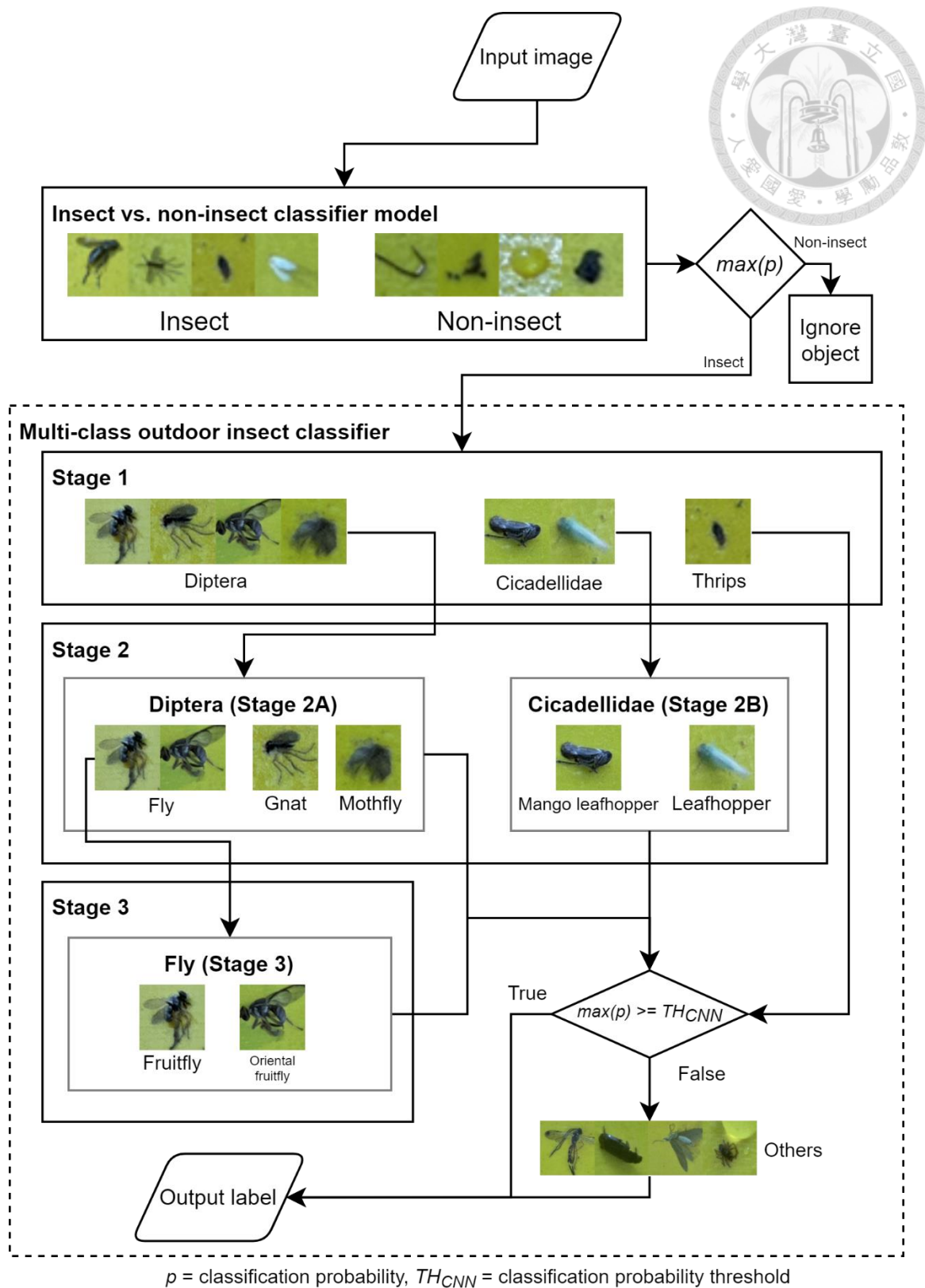
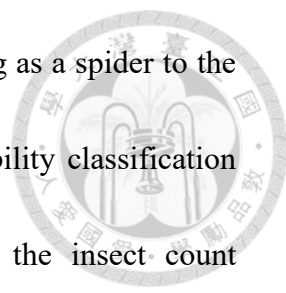


Fig. 3-29. Multi-class outdoor insect classifier flowchart.



may resemble the target insect types such as a gnat or fly appearing as a spider to the classifier model. Another workaround is to ignore the low probability classification results (Bendale & Boulton, 2016); however, this method loses the insect count information since the objects are already classified as insect objects in the first place. Thus, the classification strategy of applying a classification probability threshold in classifying the others class was considered a plausible solution for this open set recognition problem; thus, not losing the count information.

3.5.3 Spatio-temporal voting method

The performance of the insect pest detection and recognition algorithm was enhanced using a spatio-temporal voting method (STVM). STVM utilizes the previous detection and recognition results obtained from the images of the similar sticky paper trap of a sensor node. The theory of STVM is based on similar techniques for object tracking (Feichtenhofer et al., 2017). In image recognition, it improves classification performance using the spatio-temporal information of images, if available. It has been applied by Pourdarbani et al. (2019) in detecting plum fruits in gardens. Majority voting (M-voting) was used to determine the final classification label from different classifiers. Sample output of the STVM as applied in this work is shown in Fig. 3-30.

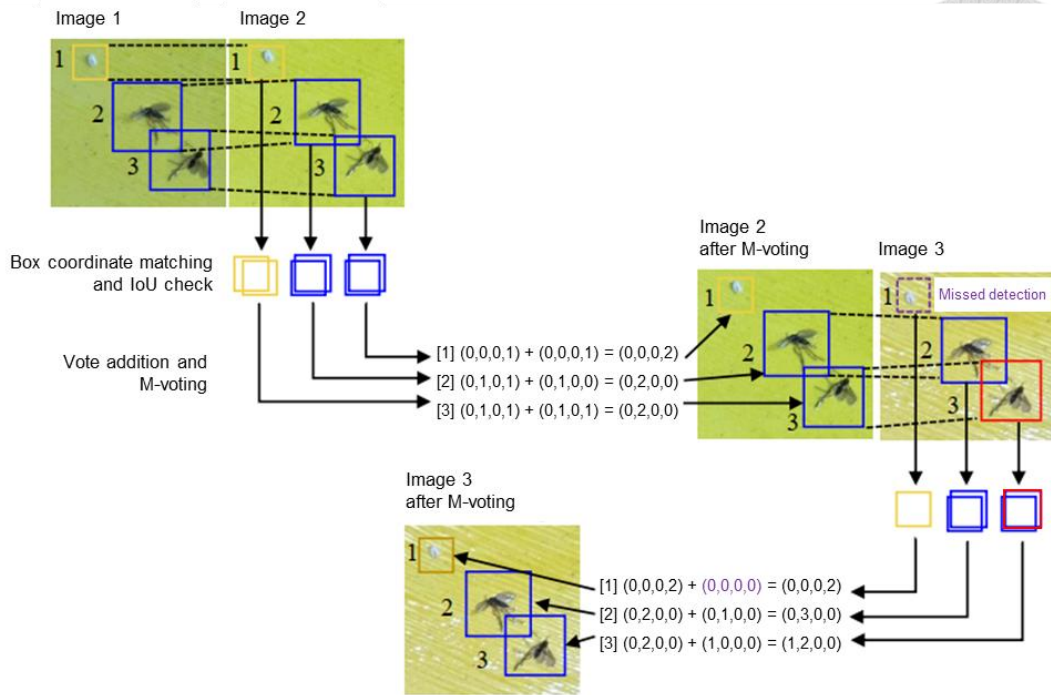


Fig. 3-30. Spatio-temporal voting method (STVM) sample flowchart.

First, the box coordinates of the detected objects of images of time $t-N_{images}$ were matched to the image of time t by checking their IoU , using Eq. 3-1, where N_{images} is the present number of collected images. The index of the image of which the object was detected and classified was recorded. The image index was used as a reference for retaining the classification labels of that object or not. The classification labels of the matched box coordinates were added and considered as votes. Majority voting, was performed on the votes to get the final classification label, using Eq. 3-2:

$$Majority\ voting\ class\ label = \operatorname{argmax}_{I_f - I_i} \frac{1}{I_f - I_i} \sum_{I_f - I_i} (CL_1, CL_2, \dots, CL_c) \quad (3-2)$$

where CL_c is the number of classification label votes of class c , I_i is the initial image index the object was found, and I_f is the final image index the object was found. In the case of missed detections, no votes were added but the classification labels were retained. The classification results were discarded if the difference of I_i and I_f was below the sum of the number of classification label votes the object has. The retention of classification labels prevents missed detections due to sudden changes in lighting condition. This technique is usually used for solving occlusion problems in multiple object tracking applications (Israni & Mewada, 2018).

3.6 Image data collection and preparation

The collected 3280 x 2464 sticky paper trap images were separated into two datasets according to location type: indoor and outdoor. Each dataset was partitioned into training set, validation set, and testing set. The summary of the data partition is shown in Table 3-4. The indoor image dataset consists of sticky paper trap images collected from the 10 indoor installation sites. Each sticky paper image contained at least more than 50 insect pests. The images were separated randomly into 80% and 20% for model training and validation, respectively. Testing sets 1 and 2 were prepared from the most recent sticky paper trap images of all installation sites, 6 months ago. Testing

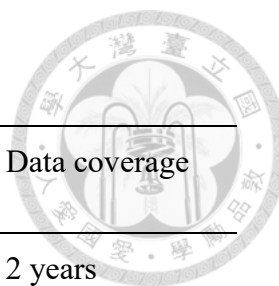


Table 3-4. Sticky paper trap image dataset partition information.

Location type	Image dataset	Source	Total # of 3280 x 2464 images	Data coverage
Indoor	Base model training and validation	All sites except M1	617	2 years
	Semi-supervised learning		105	Recent year
	Testing 1		40	Recent 6 months
Outdoor	Training and validation 2	M1 only	41	1 year and 6 months
	Testing 2		18	Recent 6 months

set 1 was also used as the classification performance reference of the semi-supervised learning applied on the multi-class insect classifier. The prepared testing sets were used to prove whether the insect pest detection and recognition algorithm worked over time.

Each image was prepared by annotating the box coordinates of the target objects using Labellmg, a graphic image annotation tool. The images were resized into 12 images of 880 x 880 resolution for training the object detector. The trained object detector model was used to crop out the insect and non-insect objects programmatically, based on the detection results. The insect vs. non-insect objects were sorted manually for training the insect vs. non-insect models. The insect objects were then sorted according to the defined insect classes, with the assistance of experts and entomologists.

A summary of the images collected for each stage is shown in Table 3-5 and Table 3-6.



Table 3-5. Indoor insect pest detection and recognition model training, validation, and testing image dataset statistics.

Model name	Training method	Class	Total	Training	Validation	Testing set 1
Indoor object detector	Supervised	Object	1377	80% of total	20% of total	312
Insect vs. non-insect (Stage 0)	Supervised	Insect	5536			1987
		Non-insect	2158			1522
Indoor insect classifier (Stage 1)*	Semi-supervised	Cranefly	25			10
		Fly	359			78
		Gnat	1824			232
		Midge	356			122
		Mosquito	47			11
		Mothfly	440			183
		Thrips	668			211
		Whitefly	1817			331

*Samples of this model are the samples of the semi-supervised learning base model (BM) and are not of the final model.



Table 3-6. Outdoor insect pest detection and recognition model training, validation, and testing image dataset statistics.

Model name	Training method	Class	Total	Training	Validation	Testing set 2
Outdoor object detector	Supervised	Object	147	80% of total	20% of total	53
Insect vs. non-insect (Stage 0)		Insect	4078			1348
		Non-insect	1404			621
Outdoor insect classifier (Stage 1)		Diptera	2705			920
		Cicadellidae	251			91
		Thrips	1122			337
Diptera classifier (Stage 2A)		Fly	514			155
		Gnat	1069			428
		Mothfly	1122			337
Cicadellidae classifier (Stage 2B)		Mango leafhopper	95			29
		Leafhopper	156			62
Fly classifier (Stage 3)		Fruitfly	432			130
		Oriental fruitfly	82			25

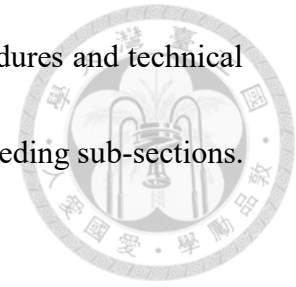
3.7 Semi-supervised learning method



From previous experiences in running the proposed system, it was found that the image samples collected over time from different locations were not always similar to each other. This caused the performance of the insect pest detection and recognition algorithm vary due to several reasons such as varying lighting condition, greenhouse operations, and presence of rare insects in the sticky paper traps. Due to this, an online semi-supervised learning method was developed as an adaptive solution. Semi-supervised learning is a method for training deep learning models by applying feature extraction, image collection, model training, model selection, and other related learning tasks with partial human supervision to continuously improve a pre-existing model (Baucum et al., 2017). This is an ideal solution for biological image monitoring systems since few images can be usually obtained from open source datasets (Amorim et al., 2019). Although semi-supervised learning is a very challenging task, it can be an efficient way in designing image monitoring systems to avoid laborious labeling effort required in supervised learning.

The pipeline of the proposed method is shown in Fig. 3-31. Throughout this text, a semi-supervised learning cycle is referred to as performing online image data collection down to model update and selection at each time t , where $t \in \{1, 2, 3, \dots, n\}$

can be defined in units of days, weeks, months or years. The procedures and technical considerations of each part of the method are presented in the succeeding sub-sections.



3.7.1 Base model building

Building the base model (BM) is a supervised step that involves training of an image classifier model with manually labelled image samples. The BM has a structure similar to as shown previously in Fig. 3-27 and was trained with insect images that were cropped out from the 3280 x 2464 sticky paper trap images and resized into 128 x 128 pixels by cubic interpolation. The BM serves two purposes: image pseudo-labelling and image feature extraction. It is an important component of the semi-supervised learning method since it contains the relevant representation knowledge, including the number of target classes and basic features of the images to be pseudo-labelled later on.

3.7.2 Online image data collection

On each semi-supervised learning cycle, online image data collection was performed. In each time t , the last sticky paper trap image was selected from the images collected of each node from $t-1$ to t . An object detector model, as mentioned in Section

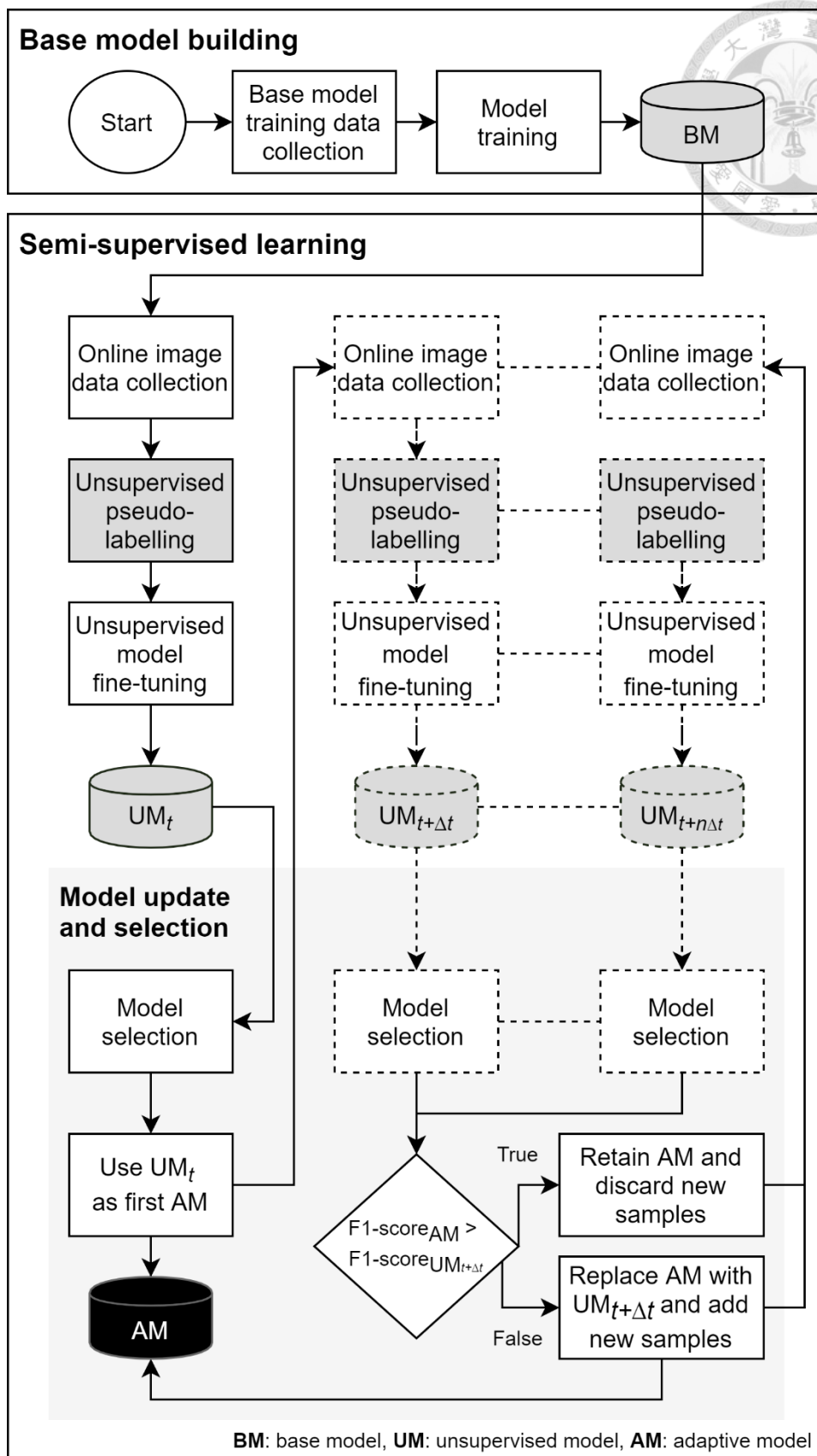
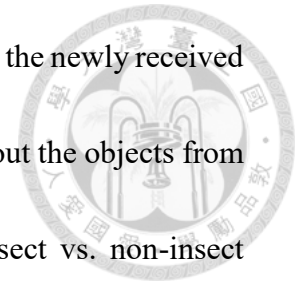


Fig. 3-31. Semi-supervised learning method flowchart.

3.5.1, was trained to obtain the coordinates of the objects found from the newly received sticky paper trap images. The object coordinates were used to crop out the objects from the images. The cropped-out objects were classified using an insect vs. non-insect classifier model; with the non-insect objects ignored while the insect objects undergo unsupervised pseudo-labelling.



3.7.3 Unsupervised pseudo-labelling algorithm

The most important component of semi-supervised learning is pseudo-labelling. Pseudo-labelling is an automatic method for labelling, sorting, or grouping sets of unlabeled image data through machine learning methods (Kwasnicka & Paradowski, 2010; Lee, 2013). The critical part of pseudo-labelling is the proper selection of image data (Veit et al., 2017). In a given set of unlabeled image data, there is always a possibility that a classifier model will falsely classify an image with high confidence, while it actually belongs to a different or undefined class. These labels are often called noisy labels. In testing by Karimi et al. (2020), adding noisy labels in a set of training images may degrade the performance of a model to a certain extent. The proposed pseudo-labelling algorithm resolves this problem by following a sequential strategy inspired by how the samples are sorted and classified with human supervision. The

pseudo-labelling algorithm is made up of three steps: image labelling, label reconfirmation, and sample cleaning, as shown in Fig. 3-32.



After the insect images were cropped out in the online image data collection step, the insect images were labelled through image classification using the BM. The initial pseudo-label L_I of each insect image was determined based on the maximum softmax probability output p of the BM. Each labelled image was grouped into its corresponding class.

Next, label reconfirmation was done by initially extracting the features of each insect image using the last layer of the BM, and then encoding the extracted features into two dimensions: PC1 and PC2, using principal component analysis (PCA). PCA is an unsupervised, non-parametric technique for dimensionality reduction and is often used in image compression, noise filtering, and data visualization since it can compress data with minimal information loss (García-Fernández et al., 2013). The projection of the PCs corresponding to the insect image features was plotted to obtain the cluster centroid of each class. The Euclidean distance d between each encoded point and the class centroid in the PCA projection plane was calculated using Eq. 3-3:

$$\text{Euclidean distance } (d) = \sqrt{(x - \bar{x}_c)^2 + (y - \bar{y}_c)^2} \quad (3-3)$$

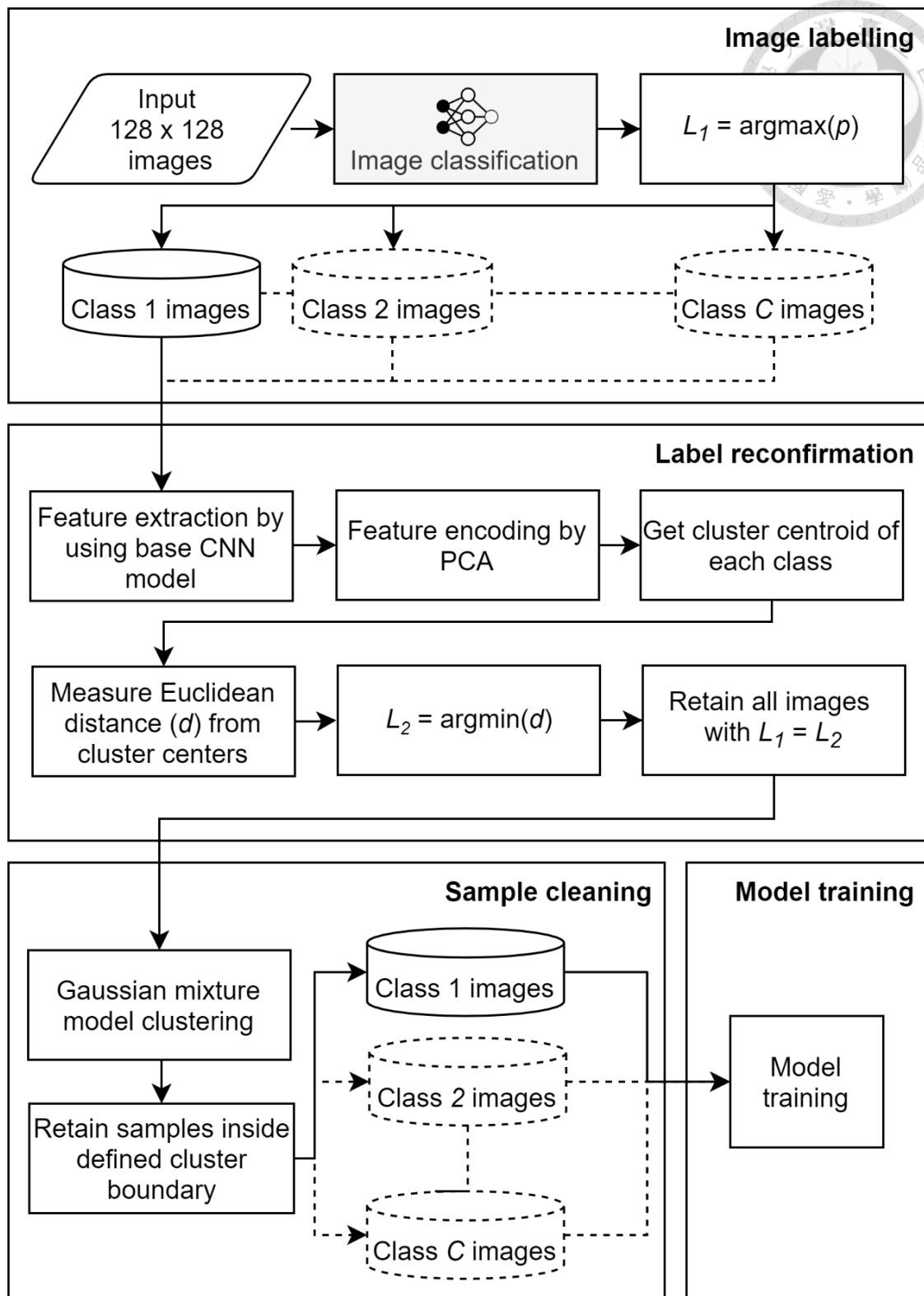


Fig. 3-32. Unsupervised pseudo-labelling method flowchart

where x and y are the encoded PC1 and PC2 value of the image, respectively, while \bar{x}_c and \bar{y}_c are the average of the encoded PC1 and PC2 values of class c , respectively. The nearest class centroid of each encoded point in the PCA projection plane was used to determine its second label L_2 . All the images with the same labelling results L_1 and L_2 were retained as new model training samples; otherwise, the images were discarded.

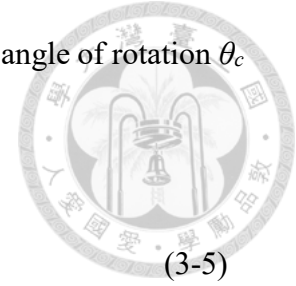
Lastly, confidence cluster ellipses were fitted onto each class of the BM training samples using Gaussian Mixture Model (GMM) clustering. GMM clustering is an unsupervised algorithm that utilizes an expectation-maximization algorithm to fit a Gaussian model that represents the cluster ellipse of a group of points. In many applications, GMM is used as a technique for detecting outliers from fitted cluster ellipses (Zong, 2018). Through GMM clustering, a covariance matrix A was computed from the training samples of each class. Singular value decomposition (SVD) was applied to calculate the eigenvalues and normalized eigenvectors of the covariance matrix. The covariance matrix A , after performing SVD, is in the form of Eq. 3-4:

$$\text{Covariance matrix } (A) = \begin{matrix} \begin{bmatrix} v_{1x} & v_{2x} \\ v_{1y} & v_{2y} \end{bmatrix} \\ U \end{matrix} \begin{matrix} \begin{bmatrix} R_{x,c} \\ R_{y,c} \end{bmatrix} \\ D \end{matrix} \quad (3-4)$$

where D includes the singular eigenvalues R_x and R_y , and the x and y radii of the cluster ellipse of class c , while U is a matrix composed of singular vectors v_{1x} , v_{2x} , v_{1y} , and v_{2y} .

The left singular vectors v_{1x} , v_{1y} were used to find the cluster ellipse angle of rotation θ_c by Eq. 3-5:

$$\text{Ellipse angle of rotation } (\theta_c) = \frac{180 \arctan\left(\frac{v_{1y}}{v_{1x}}\right)}{\pi} \quad (3-5)$$



The radii and angle of the cluster ellipse were used to exclude pseudo-labelled samples based on the inequality equation of a point inside an ellipse (Eq. 3-6), modified from Larson and Falvo (2011):

$$\left[\frac{(x - \bar{x}_c) \cos(\theta_c) - (y - \bar{y}_c) \sin(\theta_c)}{(\omega R_{x,c})^2} \right]^2 + \left[\frac{(x - \bar{x}_c) \sin(\theta_c) - (y - \bar{y}_c) \cos(\theta_c)}{(\omega R_{y,c})^2} \right]^2 \leq 1 \quad (3-6)$$

where values of points smaller than, or equal to, 1 are inside the ellipse and values greater than 1 are outside the ellipse. A scaling factor ω was introduced in Eq. 3-6. The scaling factor ω is an arbitrary parameter of the pseudo-labelling algorithm that defines the confidence level of the cluster ellipse for sample cleaning. In theory, r_x and r_y are equal to the x and y standard deviation of the cluster ellipse, respectively (Larson & Falvo, 2011). Through the scaling factor ω , the size of the cluster ellipse can be adjusted. The value of ω was computed from the square root of the probability of a chosen confidence level found on a Chi-Square probability table (Milton & Arnold, 2004).

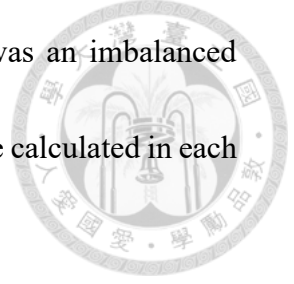


3.7.4 Unsupervised model fine-tuning

Fine-tuning without human supervision was performed on the BM by including the pseudo-labelled images to the current image dataset. Fine-tuning is a transfer learning technique that is done by freezing the first few layers of a neural network model while leaving the last layer, the deep feature extraction layer, for training. Fine-tuning aims to dynamically learn from new samples by partially changing the layer weights of the neural network model. In this work, fine-tuning was used since it can preserve the basic features learned from the first few layers, while it only learns more specific features of new image samples collected at different times and locations.

3.7.5 Model update and selection

Under online system operation, an adaptive model (AM) was selected after each semi-supervised learning cycle. It was called as adaptive model since it continuously learns from the incoming image data of each location. It is the actual model used by the monitoring system. In each semi-supervised learning cycle, the fine-tuned AMs are tested by stratified k -fold cross validation. Stratified k -fold is a validation strategy that splits training images into k number of folds and ensures that the proportion of training



samples per class are identical. It avoids biases whenever there was an imbalanced number of samples. The precision, recall, and F_1 -score per class were calculated in each fold as follows:

$$Precision = \frac{TP}{\text{predicted number of objects (detections or classification)}} \quad (3-7)$$

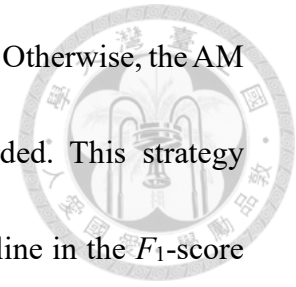
$$Recall = \frac{TP}{\text{true number of objects (detections or classification)}} \quad (3-8)$$

$$F_1\text{-score} = 2 \left(\frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \right) \quad (3-9)$$

where TP are true positive detections and/or true positive classifications later on in the classifier stage. F_1 -score is a classification performance metric that considers both precision and recall (Rustia et al., 2020a). The value of F_1 -score ranges from 0 to 1, where values closer to 1 reflect better classification performance. Since this work involves multi-class classification, the average F_1 -scores shown in the results are the macro-average of the F_1 -scores, the sum of the F_1 -score per class divided by the number of classes.

In selecting the new AM after each semi-supervised learning cycle, the validation F_1 -score of the current AM was compared with the next unsupervised model at $t+\Delta t$ ($UM_{t+\Delta t}$), as shown in Fig. 3-31. If the F_1 -score of the UM was higher than the F_1 -score of the AM, the AM was replaced with the new unsupervised model and the pseudo-

labelled samples were added to the semi-supervised training dataset. Otherwise, the AM was retained and the new pseudo-labelled samples were discarded. This strategy ensures that the noisy pseudo-labels, that can possibly cause a decline in the F_1 -score and contamination of the semi-supervised training dataset, are avoided.



3.8 Algorithm evaluation

The evaluation of the algorithm was broken into several parts. First, the performance of the object detector model was measured by image level. Evaluating by image level refers to the evaluation of the algorithm as tested on each 3280 x 2464 sticky paper trap image, as a whole. A BM, containing images from the indoor sites, was trained by supervised learning and evaluated in an object level, which evaluates the performance of the classifiers regardless of the number of insects found on each sticky paper trap image. As the BM was evaluated from the previous step, it was continuously evaluated using metrics developed for measuring the performance of the semi-supervised learning method. Meanwhile, the supervised classifier model for the outdoor site was also evaluated in an object level. These evaluation methods were designed to impartially assess the performance of each stage of the algorithm. This section explains about the evaluation methods involved in each stage of the algorithm.



3.8.1 Object detector image level evaluation

For measuring the performance of the object detector, several metrics were used for evaluation (Rustia et al., 2020a):

$$\text{Miss rate (MR)} = \frac{\text{missed detections}}{\text{true number of objects (detections)}} \quad (3-10)$$

$$\text{False positive rate (FPR)} = \frac{\text{wrong detections}}{\text{predicted number of objects (detections)}} \quad (3-11)$$

This also includes the metric, F_1 -score (Eq. 3-9). The object detector was tuned based on the validation images and evaluated on the testing sets on an image level. For programmatically checking the detection results, the bounding box coordinates of the annotated objects were matched with the detected objects using IoU . If the IoU value measured from the compared objects was higher than 0.5, the object was considered a correct detection. A wrong detection was determined when the detected object does not match any of the annotated objects, and a missed detection, otherwise.

3.8.2 Image classifier object level evaluation

Both the insect vs. non-insect and multi-class insect classifiers were evaluated by the object level. Based on the F_1 -score definition in Eq. 3-9, the image classifiers were

evaluated using the average of the F_1 -scores per class. The modified equation for F_1 -score is shown in Eq. 3-12:

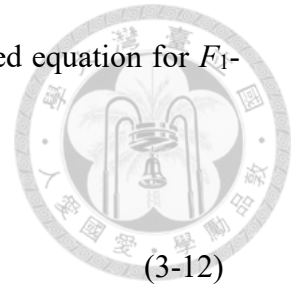
$$\text{Average } F_1\text{-score} = \frac{\sum_{c=1}^C F_1\text{score}_c}{C} \quad (3-12)$$

where c is the class index and C is the number of target classes of the classifier. The average F_1 -score was used as the main metric for image classifier model evaluation as it considers the false positives and false negatives of the classifier, unlike other basic metrics.

3.8.3 Integrated algorithm evaluation

The integrated algorithm was evaluated in both image and object levels. In object level evaluation of the integrated algorithm, the 128×128 images were extracted from the original 3280×2464 images and sorted for testing. Using the sorted images, the F_1 -scores of each class (Eq. 3-9) and the average F_1 -score (Eq. 3-12) were computed based on the confusion matrix results.

In the image level, two metrics were measured: F_1 -score and mean counting accuracy ACC . Image level F_1 -score was computed from the precision and recall similar to Eq. 3-12, but in a per class and per sticky paper trap image basis. For measuring the



correctness of the automatic counts based on its relative difference to the manual counts, average counting accuracy ACC was used and computed using Eq. 3-13:

$$\text{Average counting accuracy (ACC)} = \frac{1 - \sum_{n=1}^{N_{images,c}} \left| \frac{M_n - A_n}{M_n} \right|}{N_{images,c}} \quad (3-13)$$

where n is the image index, $N_{images,c}$ is the total number of images of class c , M is the manual count and A is the automatic count by the algorithm.

3.8.4 Semi-supervised learning evaluation

The semi-supervised learning method was evaluated both visually and numerically. In visual evaluation, the extracted final CNN layer features of each image were encoded using PCA and plotted onto a two-dimensional plane to observe the variation and distribution of the image sample features. PCA was used since it is clear enough to be explained by human observation. Unlike other methods, PCA is non-parametric and non-probabilistic, making it very consistent (García-Fernández et al., 2013).

In numerical evaluation, two sets of metrics were computed after each semi-supervised learning cycle: online and offline. Online metrics are the values that can be monitored while the system was running; they can provide the approximate real-time

performance of the trained models. The calculated online metrics were as follows: validation F_1 -score, cluster density CD , and silhouette score SS . The validation F_1 -score was calculated based on the average F_1 -score of each class from the stratified 5-fold split of the entire pseudo-labelled dataset, similar to Eq. 3-9. To measure the relative amount of data that each cluster contains, the cluster density CD was calculated:

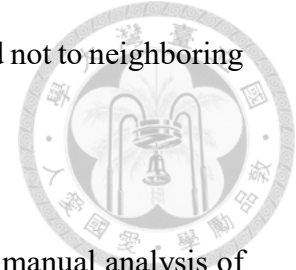
$$\text{Cluster density } (CD) = \frac{N_{pseudo,c}}{2\pi r_x r_y} \quad (3-14)$$

where $N_{pseudo,c}$ is the total number of pseudo-labelled images of class c ; the divisor was derived from the formula for the area of an ellipse (Larson & Falvo, 2011). The value of CD was used to measure the relative probability that a group of points can be clustered. Meanwhile, the silhouette score SS was calculated using its standard formula, as follows (Palacio-Niño & Galiano, 2019):

$$\text{Silhouette score } (SS) = \frac{1}{n} \sum_{i=1}^n \frac{b(i) - a(i)}{\max(a(i), b(i))} \quad (3-15)$$

where $b(i)$ is the minimum average distance between the samples, but not containing the analyzed sample i , $a(i)$ is the average distance of all samples to their own cluster, and n is the number of samples. The terms $b(i)$ and $a(i)$ are also called separation and cohesion, respectively. The value of SS ranges from -1 to 1 where values close to 1

mean that the points are grouped accordingly to their own cluster and not to neighboring clusters.



The offline metrics are values that can only be computed after manual analysis of the data. This includes the pseudo-labelling accuracy $ACC_{pseudo,c}$, where c is the class label, and testing F_1 -score. The pseudo-labelled images were inspected manually and the number of clean and noisy pseudo-labels were counted for computation as follows:

$$\text{Pseudo-labelling accuracy } (ACC_{pseudo,c}) = \frac{N_{clean,c}}{N_{pseudo,c}} \quad (3-16)$$

where $N_{clean,c}$ is the number of clean pseudo-labelled images and N_{pseudo} is the total number of pseudo-labelled images. $ACC_{pseudo,c}$ measures the correctness of the pseudo-labelling algorithm. Meanwhile, testing F_1 -score uses a similar definition as Eq. 3-9.

3.9 System performance indicators

Besides the algorithm performance, the system was evaluated based on two other performance indicators: data throughput and insect pest trapping efficacy. Detailed discussion about the indicators is presented in the next sub-sections.



3.9.1 System data throughput

Data throughput is used to measure the reliability of a data collection system in terms of the data sent by the sensor nodes and data received by the system. It reflects how reliable the system is regardless of the environment it was installed in. A set of data was defined as received if a sensor node was able to send at least 50% of the data it is expected to send and lost, otherwise. In this context, an indoor sensor node should be able to send at least 48 environmental data packets while an outdoor node should send at least 12 environmental data packets. Each sensor node should send at least 7 sticky paper trap images. Using this condition, the data throughput was computed using Eq. 3-17:

$$\text{Data throughput (\%)} = 100 * \frac{\text{Data received}}{\text{Data expected}} \quad (3-17)$$

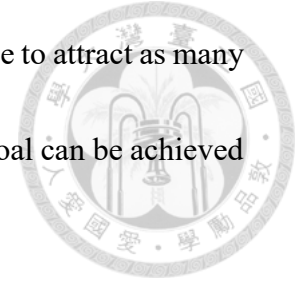
On the other hand, data loss was computed using Eq. 3-18:

$$\text{Data loss (\%)} = 100 * \frac{\text{Data lost}}{\text{Data expected}} \quad (3-18)$$

3.9.2 System insect pest trapping efficacy

One of the key goals of this research is to replace manually set sticky paper traps with an automated device such as the sensor node. An evaluation was necessary to

determine if a sensor node, with a sticky paper trap attached, was able to attract as many insects as a sticky paper trap separated from the sensor node. This goal can be achieved if similar data trends were obtained from both methods.



An experiment was designed with the assistance of TDARES in which a cylindrical separate sticky paper trap was put below each sensor node of Farms TS1, TS2, and TS3. The sticky paper trap was rolled into a cylinder since it was found from previous studies that it is the most effective trap shape that can attract most insects (Ghani et al., 2012). TDARES manually counted the selected insect pests from the sticky paper traps such as whitefly and thrips. Only the two insect pest types were counted since they are relatively more harmful compared to the other insect pest types that the system can detect and recognize. The experimental setup is shown in Fig. 3-33.

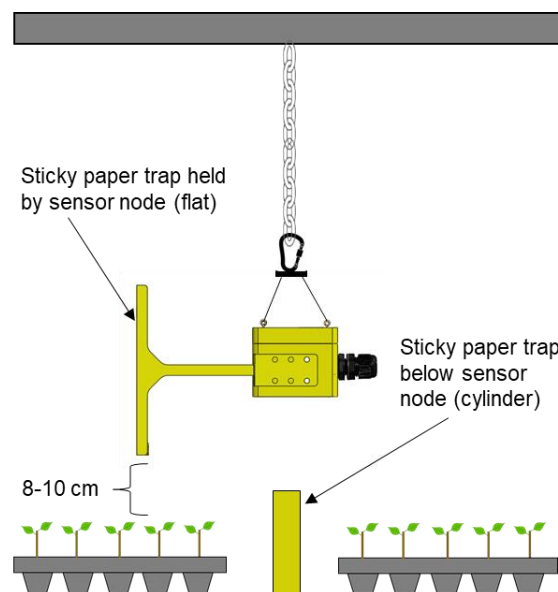
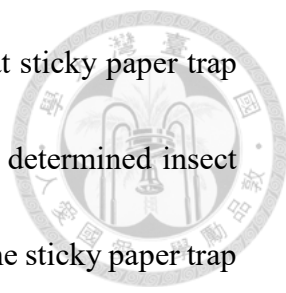


Fig. 3-33. Sensor node trapping efficacy experimental setup.



In this evaluation, the automatic insect pest count IC of the flat sticky paper trap held by the sensor node is referred to as IC_{auto} , while its manually determined insect pest count is referred to as IC_{device} . The manual insect pest count of the sticky paper trap below the sensor node is referred to as IC_{below} . The insect pest counts obtained from all insect trapping setups were normalized into values from 0 to 1 to evaluate the system regardless of scale. Normalization removes the bias in comparing two related variables. Normalization of the insect pest counts was done in the form of Eq. 3-19:

$$IC^* = \frac{IC - \min(IC)}{\max(IC) - \min(IC)} \quad (3-19)$$

3.10 Data analytics

The proposed system is incomplete without providing data analytics that can concisely explain the insect pest condition in their farms. Unlike scholars or researchers, not all users of the system are familiar with the meaning of the numerical data shown to them. The only way to make the system useful to them is to interpret and transform data into information. This section discusses about the theory of the different data analytics techniques applied in the proposed system.

3.10.1 Insect pest count alarm model development



Based from the feedback of the users during the first few months of running the system, it was found that the insect pest counts data was not adequate to assist the farmers in decision-making. The users had difficulty in interpreting the data since they had no idea of how many insect pests detected can be considered critical in their farm. The proposed solution to this problem was to develop an alarm model. An alarm model was used to automatically compute for threshold values that can be used to determine which decisions can be recommended under certain values of the input data. The alarm model in this work aims to adaptively convert the increase in insect pest count ΔIC per day into an equivalent alarm level using alarm thresholds in the form of $TH_{\Delta IC a}$, where a is the alarm level number, with 1 and 5 having the lowest and highest priority, respectively. The flowchart of the alarm model is shown in Fig. 3-34.

First, it was checked whether there were crops in the farm or not. If there were, the historical ΔIC data was retrieved from the server and the number of unique values (u) was counted. In this research, u was set to 20, a value found after tuning on different datasets. If the number of unique values was less than the value of u , only the insect counts were shown to the users; such as not to cause confusion by delivering invalid alarms to the users. Otherwise, the SEVERE alarm threshold $TH_{\Delta IC 5}$ was computed

using isolation forest. Isolation forest is an unsupervised learning technique that is used to detect anomalies or outliers from a dataset. Unlike any other techniques, isolation forest attempts to isolate the fewer and different points instead of finding the normal points in a dataset. It iteratively builds isolation trees, with a similar structure as binary search trees, to form decision trees that split the dataset into normal and anomalous points. Each data point was fed to each tree of the trained isolation forest model and a so-called anomaly score S was computed using Eq. 3-20 (Liu et al., 2008):

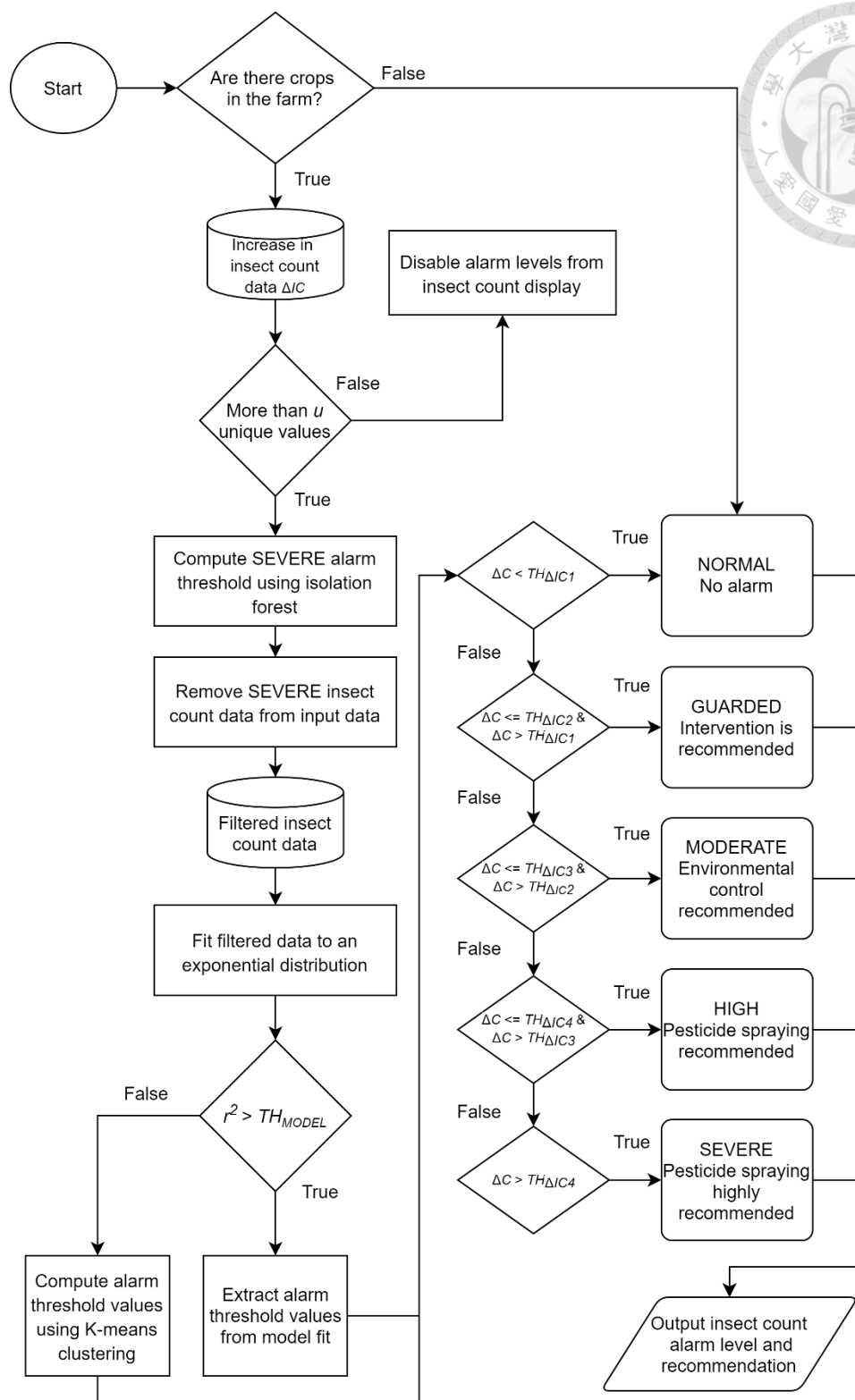
$$\text{Anomaly score } (S) = 2^{-\frac{E}{c(n)}} \quad (3-20)$$

where E is the average path length $h(k, m, N_{trees})$, computed using Eq. 3-21:

$$\text{Isolation forest average path length } (E) = \frac{\sum_{i=1}^{N_{trees}} \begin{cases} \sum_{j=1}^M 1, & k = 1 \\ \sum_{j=1}^M 1 + c(k), & \text{otherwise} \end{cases}}{N_{trees}} \quad (3-21)$$

where N_{trees} is the total number of isolation trees, M is the total number of binary splits, and k is the total number of data points in the exit node. Meanwhile, $c(n)$ is the unsuccessful search average path length, defined using Eq. 3-22:

$$\text{Unsuccessful search average path length } c(n) = 2(\ln(n - 1) + \varepsilon) - 2\left(\frac{n - 1}{n}\right) \quad (3-22)$$



ΔIC = increase in insect count
 $TH_{\Delta IC a}$ = ΔIC threshold of alarm level a
 TH_{MODEL} = model fit r^2 threshold
 u = number of unique ΔIC values

Fig. 3-34. Insect count alarm model flowchart.

where n is the number of data points in the dataset and ε is the Euler constant with a value 0.5772156649. Anomalies were determined from the values of S , where a lower value of S means higher tendency of being an anomaly. An anomaly threshold TH_{IF} was set to determine at which values of S , as matched to each value of ΔIC , were considered as anomalies. In this research, the S value of a corresponding value of ΔIC lower than $TH_{IF} = -0.1$ was considered as an anomaly. The said value also represents the hypothetical percentage of anomaly values in relation to the input dataset.

Afterwards, the ΔIC data lower than $TH_{\Delta IC5}$ were fit to an exponential distribution function using Eq. 3-23:


$$y = \lambda e^{-\lambda \Delta IC} \quad (3-23)$$

where λ is the scale parameter of an exponential distribution. The optimal value of λ was estimated using Levenberg-Marquadt (LM) method (Haefner, 2005). The estimated value of λ was used to compute for the ΔIC threshold value of alarm level a $TH_{\Delta ICa}$ using Eq. 3-24:

$$\text{Change in insect count alarm level threshold } (TH_{\Delta ICa}) = -\frac{\log(CI)}{\lambda} \quad (3-24)$$

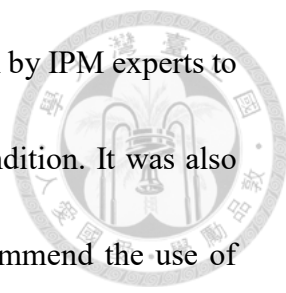
where CI is the defined confidence interval or probability of the specified alarm level. It was used to obtain the thresholds of alarm levels 2 (GUARDED), 3 (MODERATE),

and 4 (HIGH), using *CI* values of 0.50, 0.20, 0.01, respectively.



The exponential distribution function was chosen for model fitting since it was observed from prior analysis that the frequency distribution of the historical ΔIC data resembles an exponential distribution. It was reasonable since there were very few insect pests found on the sticky paper traps most of the time since the farm managers continuously control the insect pest population. On the other hand, the frequency of experiencing insect pest outbreaks was very low, which was found to be due to the times that the farm managers sometimes do not immediately respond to the insect pest condition. Based on these observations, it was assumed that if the computed coefficient of determination r^2 after fitting to an exponential distribution was lower than a set model fit threshold TH_{MODEL} , then the trend in the insect pest counts was abnormal. In this work, the value of TH_{MODEL} was set to 0.7. Alternatively, K-means clustering was applied. K-means clustering is an unsupervised learning technique that iteratively groups data points into K number of clusters or centroids. The centroids are used to train a k-nearest neighbor (kNN) classifier which classifies data points based on their distance to the centroids found. K-means was used as an alternative method since it does not depend on the frequency distribution of the data. However, it was found that using K-means equally cuts the dataset, which leads to too many SEVERE alarms.

Finally, recommendations were displayed in the user interfaces, together with the



alarm levels. The recommendations were based on the advices given by IPM experts to the farm managers depending on the severity of the insect pest condition. It was also specifically defined that only the HIGH and SEVERE levels recommend the use of pesticides. This was meant to help the farm managers reduce their pesticide usage and use alternative methods such as isolation of crops, environmental control, and other natural methods. Notifications were shown in the user interfaces whenever the alarm level was HIGH to SEVERE based on their defined time of day.

3.10.2 Insect pest hotspot detection

One way to optimize the integrated pest management is to selectively control the insect pest population in specific hotspots. Based on the collected insect pest counts per sensor node, hotspots were detected by ranking the insect pest count data in ascending order. The insect pest count data per sensor node was displayed using gradient colors from white (lowest) to black (highest). The insect hotspot information can be used by the farm managers to reduce their pesticide usage since other locations may not have any insect pests around. The farm managers may also simply isolate the crops located in the hotspots and move them away from the other crops. This information was given simultaneously in the user interfaces with the insect pest count alarm.



3.10.3 Insect pest flight rate model development

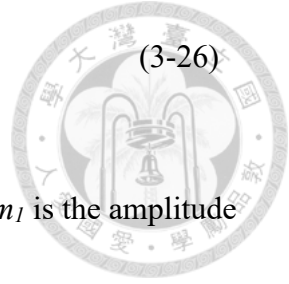
The temperature data and insect pest count data obtained by the system was used to develop an insect flight model. The insect flight model was used to estimate the probability that an insect will fly into the sticky paper trap. The model can help the farm managers to control the temperature condition in their farm such that to diminish the activity of the insect pests and stop them from emerging.

To prepare the data for model fitting, a series of pre-processing steps was performed. First, the moving average of the increase in insect pests counts from T_i to T_f was obtained using a window size of 1.5°C from 0°C to 50°C was obtained. This step converts the increase in insect pest counts into insect flight rate F under different temperature levels. The outliers from the F data were determined using Eq. 3-25:

$$F_T > (\bar{F} - 3\sigma_F), \quad F_T < (\bar{F} + 3\sigma_F) \quad (3-25)$$

where F_T is the insect flight rate at temperature T and σ_F is the standard deviation of the insect flight rate. Eq. 3-25 shows that all values three standard deviations below or above the average of F were considered as outliers; following the definition of an outlier in a Gaussian distribution. All the outliers detected were converted to the average of F . The filtered F data was fit to a five-parameter double Weibull function (Haefner, 2005):

$$\text{Insect flight rate } (F) = m_1 \left(1 - e^{-\left(\frac{T}{m_2}\right)^{m_3}} \right) e^{-\left(\frac{T}{m_4}\right)^{m_5}} \quad (3-26)$$



where F is the insect flight rate (time^{-1}), T is the temperature in $^{\circ}\text{C}$, m_1 is the amplitude of F , and m_2 , m_3 , m_4 , and m_5 , are the shape and scale parameters of the Weibull distribution and its complement, respectively. The five-parameter double Weibull function is the product of a two-parameter Weibull function and its complement. It is a widely used distribution function for modeling biological phenomena (Haefner, 2005; Rustia & Lin, 2019). The five parameters of Eq. 3-26 were estimated using LM method. Finally, the values of F from the fitted model was normalized from 0 to 1 to generalize the model.

Three important data were obtained from the fitted model: lower niche temperature $T_{niche,min1}$, peak niche temperature $T_{niche,max}$, and upper niche temperature $T_{niche,min2}$. Niche temperature is the temperature level in which there is non-zero probability that the insects will take flight. The lower and upper niche temperature are the temperature levels in which there is a minimum probability of insect flight, while the peak niche temperature represents the temperature level with maximum flight probability. This information was provided in the user interfaces so that the farm managers know the possible environmental control conditions they can apply to control the insect activity.



3.10.4 Crop growth information

To assist the farmers in optimizing the yield of their crops, two crop growth information were extracted from the sensor data: daily light integral *DLI* and growing degree days *GDD*. Both crop growth information are indicators used by the farm managers to estimate how much enough energy the crops received from sunlight and temperature (Miller et al., 2018; Rustia & Chung, 2016).

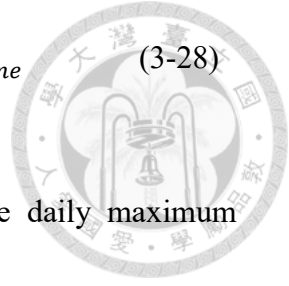
DLI was computed using the measured light intensity data of the monitoring system, using Eq. 3-27:

$$\text{Daily light integral (DLI)} = \frac{CF * LI_{24} * 3600 * 24}{1000000} \quad (3-27)$$

where LI_{24} is the average light intensity over a 24-hour period in lux, CF is the calibration factor to convert lux to $\text{mol/m}^2/\text{s}$ (Thimijan & Heins, 1983), 3600 is computed from the number of seconds in a 60 minute period, and 24 is the number of hours in a day. *DLI* is measured in $\mu\text{mol/m}^2/\text{day}$ or simply as photosynthetic photon flux density (ppfd).

On the other hand, *GDD* was computed from the measured temperature data of the monitoring system using Eq. 3-28:

$$\text{Growing degree days (GDD)} = \frac{(T_{min} + T_{max})}{2} - T_{baseline} \quad (3-28)$$



where T_{min} is the daily minimum temperature in °C, T_{max} is the daily maximum temperature in °C, and $T_{baseline}$ is the growing degree days baseline temperature in °C.

In this research, a general $T_{baseline}$ of 18°C was used, which is the most common lower niche temperature for crop growth.

3.11 System front-end

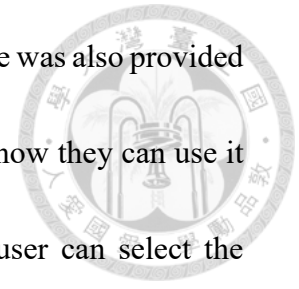
To distribute the system to the users of the system, two major front-end user interfaces were designed: website and mobile APP. The user interfaces contain all the data and information that the users may need to perform efficient and data-driven IPM. All the data displayed in the website and mobile APP were pre-processed and converted to locally stored .csv files in the server using background programs written in Python 3.5, supported by Pandas data analysis library (McKinney, 2010) and Scikit-learn machine learning library (Pedregosa et al., 2011). Unlike loading all the data from a database, pre-processing the data into .csv files greatly decreases the loading time and makes the user interfaces more fluid. This section discusses about the design and development of the user interfaces.

3.11.1 Website



The system website was written using PHP, Javascript, and HTML programming languages. The display data were obtained from the MySQL database and shown through charts and graphs using Highcharts, an interactive Javascript library for generating charts. In the website home page, as shown in Fig. 3-35, the users can view a summary of their data such as the current insect pest count data and alarm, percentage of insect pest counts, cumulative insect pest counts, environmental data, and more. For more in-depth data analysis, the users can access the temporal analysis page (Fig. 3-36) in which users can select the data they need to analyze, such as insect pest count and environmental data, and pick the starting date and ending date of the data. In the same page, users can also pick the data period frequency such as daily, monthly or per DB. In this context, a DB is referred to as period in which a set of sticky paper traps was not replaced with clean sticky paper traps. The users can also check the spatial analysis page (Fig. 3-37) which shows the insect pest count and environmental data in a per node display. The insect pest hotspots were also shown in the same page. Finally, users can also print out their data by exporting them to a .pdf file through the report page, as illustrated in Fig. 3-38. The report was also sent to the users depending on pre-defined subscription settings. The report contains data analyses for all the data obtained by the

system such as insect pest count and environmental data. A user guide was also provided in the same .pdf file explaining about the meaning of the data and how they can use it in their IPM program. In the case of having multiple sites, the user can select the location name in each page. The website can be viewed in any browser using a desktop computer or mobile device such as smartphone or tablet.



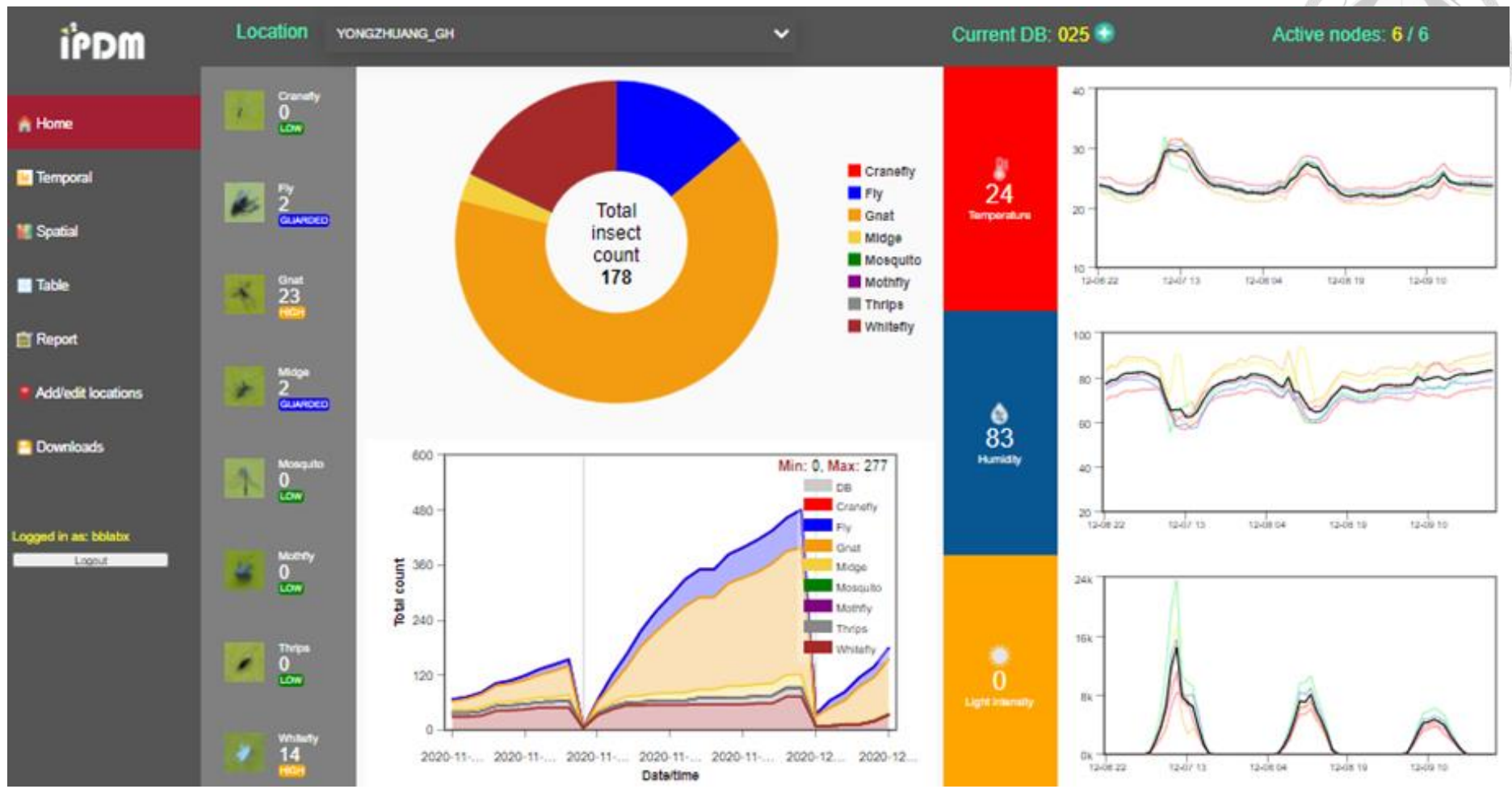
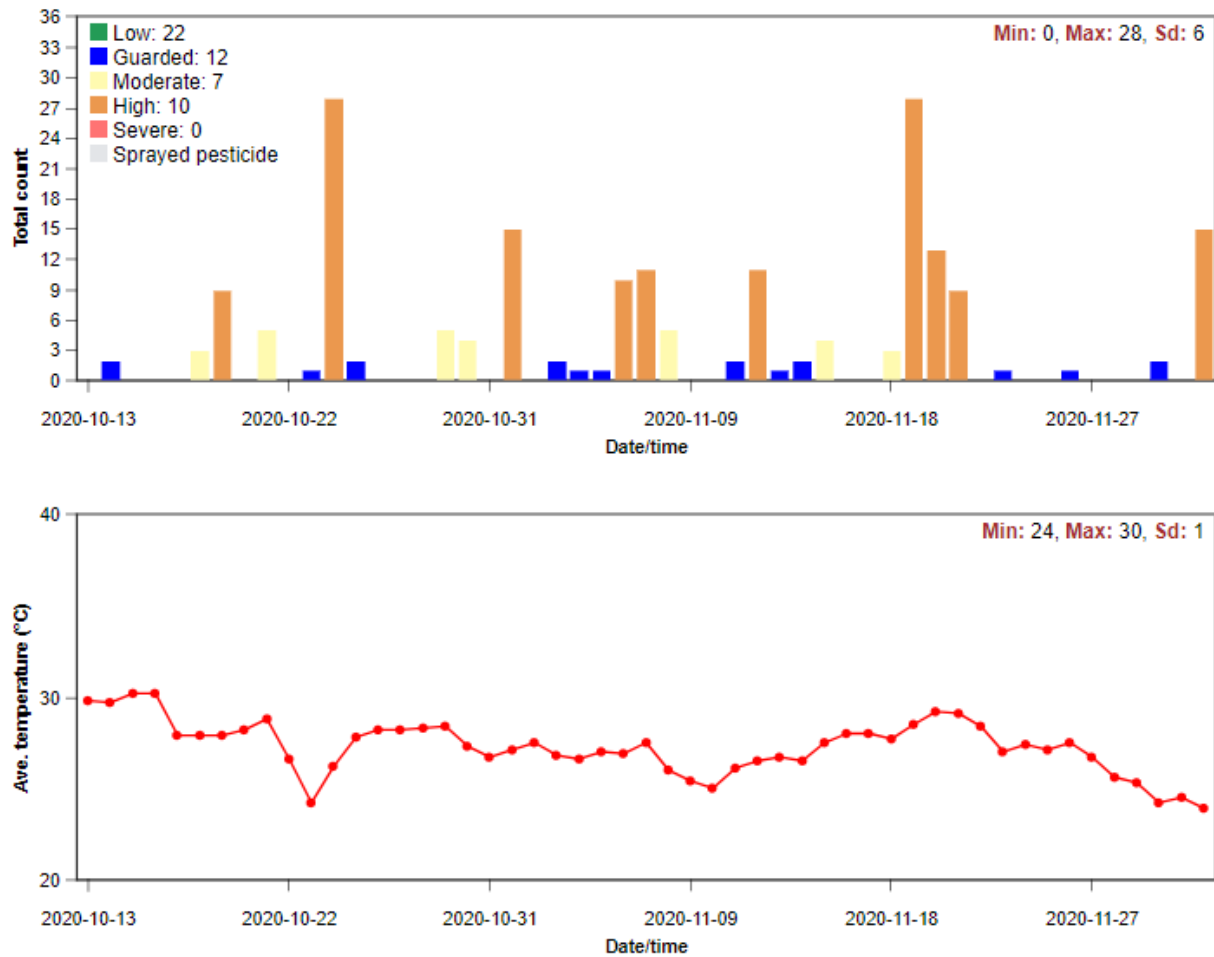


Fig. 3-35. I²PDM website home page.

i²PDM

- Home
- Temporal**
- Spatial
- Table
- Report
- Add/edit locations
- Downloads

Logged in as: bblabx
Logout



Location 1
YONGZHUANG_GH

Data type 1
Insect count

Data 1
Whitefly

Location 2
YONGZHUANG_GH

Data type 2
Sensor data

Data 2
Temperature

Frequency
Daily

Start: 2020-10-13

End: 2020-12-02

Submit query

Fig. 3-36. I²PDM website temporal analysis page.

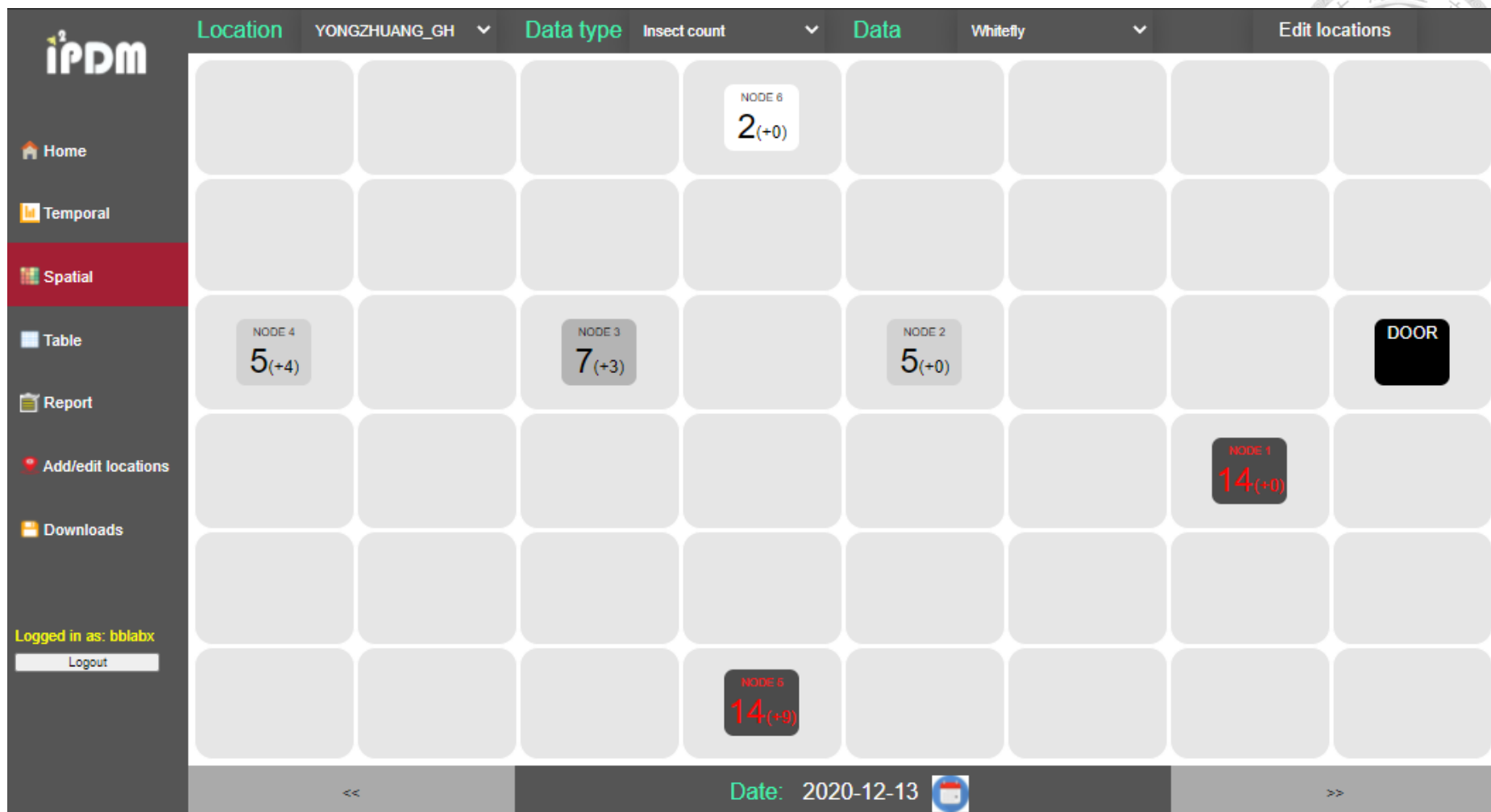


Fig. 3-37. IPDM website spatial analysis page.



i²PDM Location YUNLIN_GH

Report generator

Query by: **Date** Language: **English**

Start: **2018-10-16** End: **2020-12-20**

GENERATE REPORT **DOWNLOAD**

E-mail report subscription settings

Enable subscription: E-mail address: **sample_email@hotmail.com**

Number of days: **5** Time: **8:00**

Days of the week: M T W TH F S SUN **SAVE SETTINGS**

Data analytics report

Date/s: 2018-10-16 - 2020-12-20
Location name: YUNLIN_GH (嘉義)
Crops: Tomato

Daily differential insect count (all)

Daily insect count distribution

Fig. 3-38. i²PDM website report page.

3.11.2 Mobile APP



Users of the system may also view their data from the mobile APP. The system mobile APP Android version was written using Java programming language and the iOS version was written using Swift programming language. The mobile APP is downloadable in the Google Play Store and APP Store for mobile devices running in Android and iOS, respectively. The mobile APP displays a more concise presentation of the users' data.

In the APP's home page (Fig. 3-39), users can pick their location, view the current environmental data, insect pest count data and alarm, local weather condition, and pesticide calendar. The pesticide calendar can be used to tick the days in which they have sprayed pesticides in their farm. It can be used to track whether their pesticide application was effective or not. This can convince the farmers to decrease their pesticide usage and realize the benefits of spraying pesticides only whenever the insect pest condition was out of hand.

Users can also access the mobile APP's analysis page, as shown in Fig. 3-40. Similar to the website, the users may also see the map view (Fig. 3-40a) which contains the spatial presentation of their insect pest count data and environmental data. It also shows the insect pest hotspots wherein users can focus on controlling the insect pest

population. The table view (Fig. 3-40b) shows the data of each sensor node and the processed sticky paper trap images. The insect pest count analysis page (Fig. 3-40c) shows the temporal change in insect pest counts while the environmental page (Fig. 3-40d) displays the environmental data. The users of the mobile APP also receive notifications whenever there was a HIGH or SEVERE insect pest count alarm, as shown in Fig. 3-41. Upon clicking the notification, users may see the current insect pest count data and pesticide recommendations for the specific insect pest. This guides the users to use appropriate pesticides and effectively reduce the insect pest population.

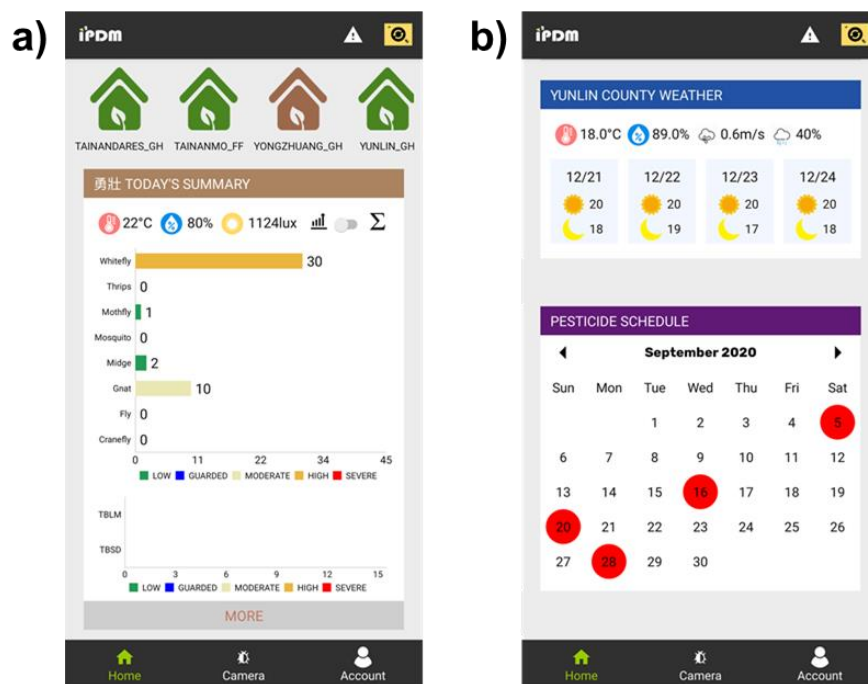


Fig. 3-39. I²PDM mobile APP home page. a) Insect pest count; b) Weather, and pesticide calendar.

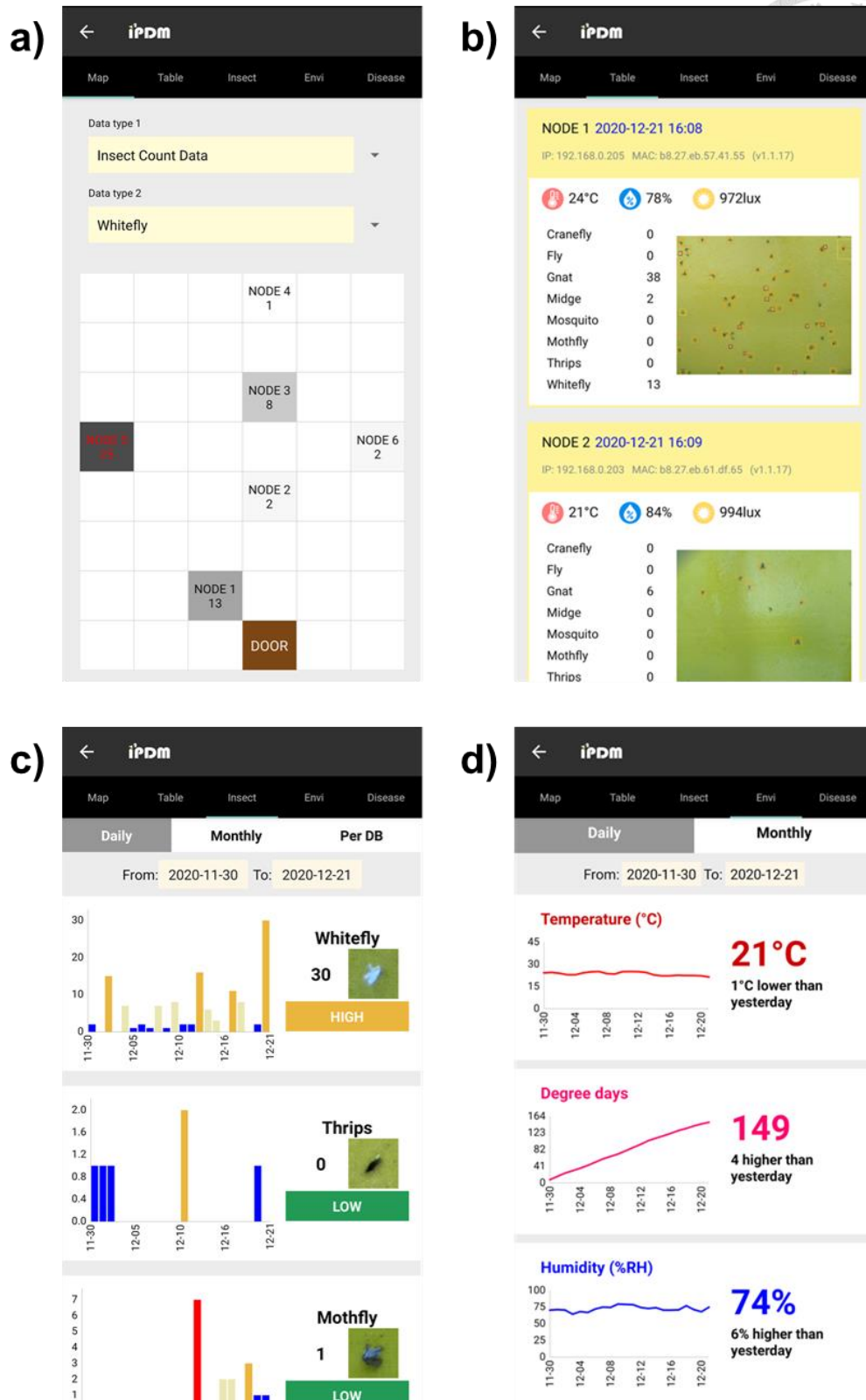


Fig. 3-40. iPDM mobile APP analysis page. a) Map view; b) Table view; c) Insect pest count analysis; and d) Environmental data analysis.

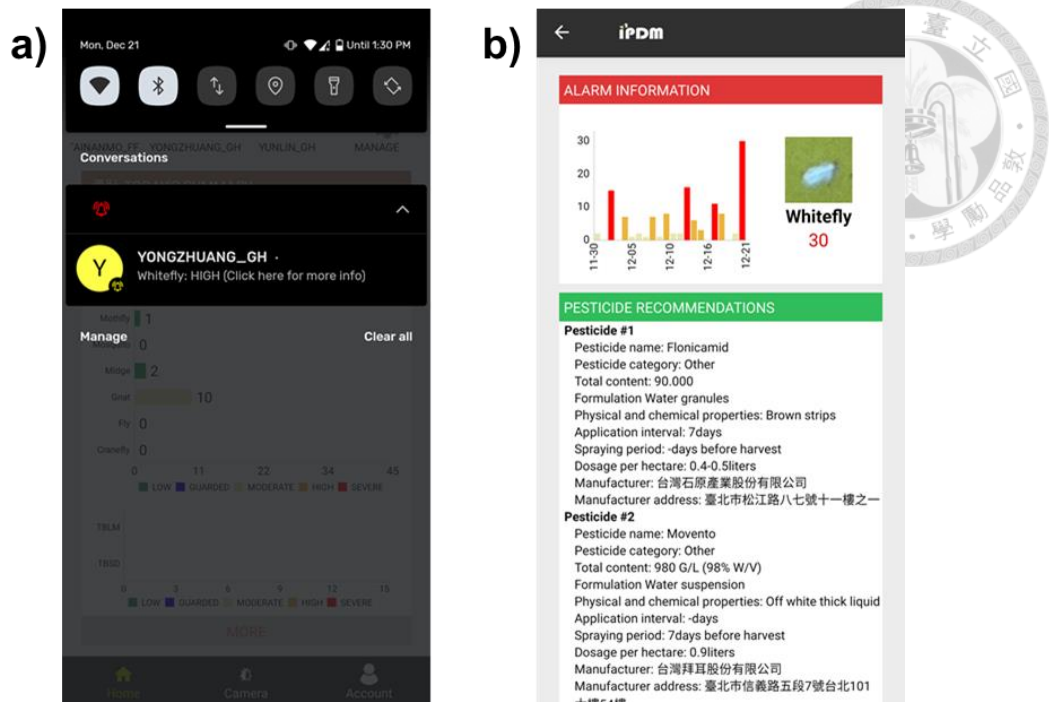


Fig. 3-41. I²PDM mobile APP alarm notification sample. a) Notification; b) Insect pest count alarm and pesticide information.

Chapter 4 Results and Discussion



4.1 Insect pest detection and recognition algorithm model

training, validation and testing

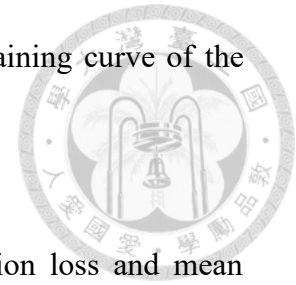
This section covers the processes involved in developing the insect pest detection and recognition algorithm. It includes the model training, tuning, and testing results of each part of the algorithm. All the related processes were carried out using a desktop computer, different from the server computer, running under Windows 10 operating system, with an Intel Core i7-6700 CPU @ 3.40 GHz, 40 GB RAM and Nvidia GTX1060 GPU.

4.1.1 Supervised object detector and insect vs. non-insect model

4.1.1.1 Indoor object detector and insect vs. non-insect model

The indoor object detector was trained with the samples of the indoor sites according to the number of prepared samples in Table 3-5. Training was done with 300 epochs, with 1000 training steps per epoch, and the model was optimized by stochastic

gradient descent using Adam optimizer as loss minimizer. The training curve of the indoor object detector is shown in Fig. 4-1.



Two metrics were used in selecting the best model: validation loss and mean average precision mAP . Validation loss is the error computed after testing on the validation set, while mAP is a measure of the correctness of the detections by the object detector model. The training curve shows that the trained model was able to quickly learn from the training images as it converged at around 200 epochs and the mAP began to saturate at 0.96. This shows that the model did not learn any new relevant features after 200 epochs. Therefore, the model trained with 200 epochs was chosen as the object detector model.

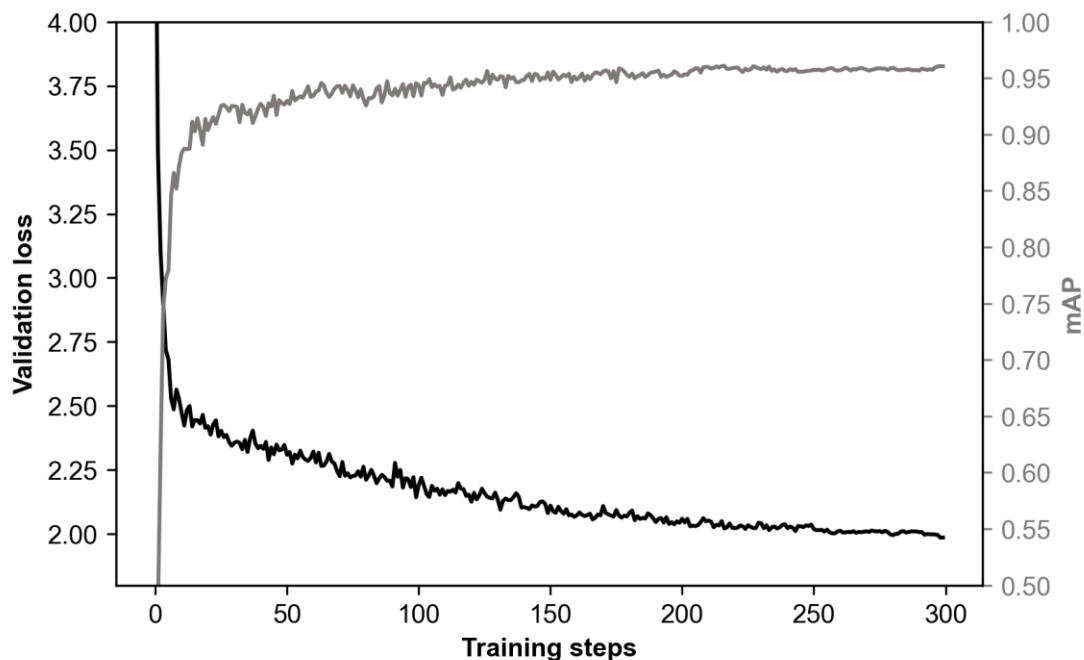
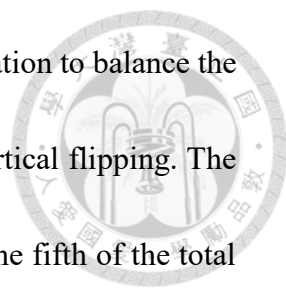


Fig. 4-1. Indoor object detector model training curve.



The insect vs. non-insect model was trained with data augmentation to balance the number of training samples by rotation, horizontal flipping, and vertical flipping. The number of training samples was balanced by increments of 2400, one fifth of the total augmented number of non-insect training samples, until the best model was found. Adam optimizer was used to minimize loss and grid search learning values of 0.01, 0.001, and 0.0001 was set, with batch sizes of 8, 16, and 32. Training of the models was done with 200 epochs, with 20 training steps each. Model validation was carried out every 20 epochs, using the manually prepared 128 x 128 images of the validation set. The model validation results are shown in Fig. 4-2.

It can be seen that the trained insect vs. non-insect model with the raw training samples had a high average validation F_1 -score, even as different classification thresholds were used. Applying data augmentation improved the validation F_1 -score to around 0.97, with 7200 balanced augmented training samples. It also showed that augmenting the data up to 9600 and 12000 did not improve the classification performance, probably caused by overfitting. Based on the results, the trained model with 7200 augmented training samples was selected, setting its classification threshold TH_{CNN} to 0.55. The results of testing the insect vs. non-insect model trained without and with data augmentation by object level is also shown in Fig. 4-2b and Fig. 4-2c, respectively. It shows that both models had very close classification performance. This

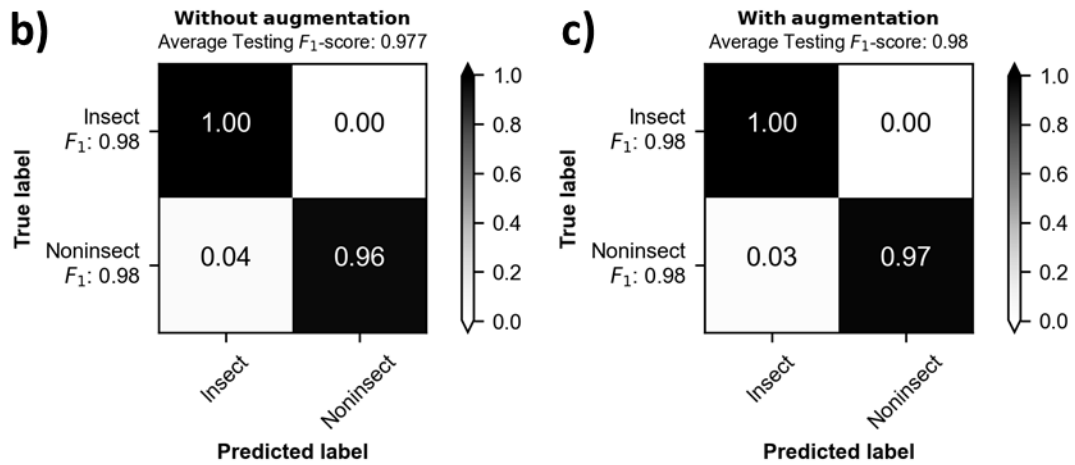
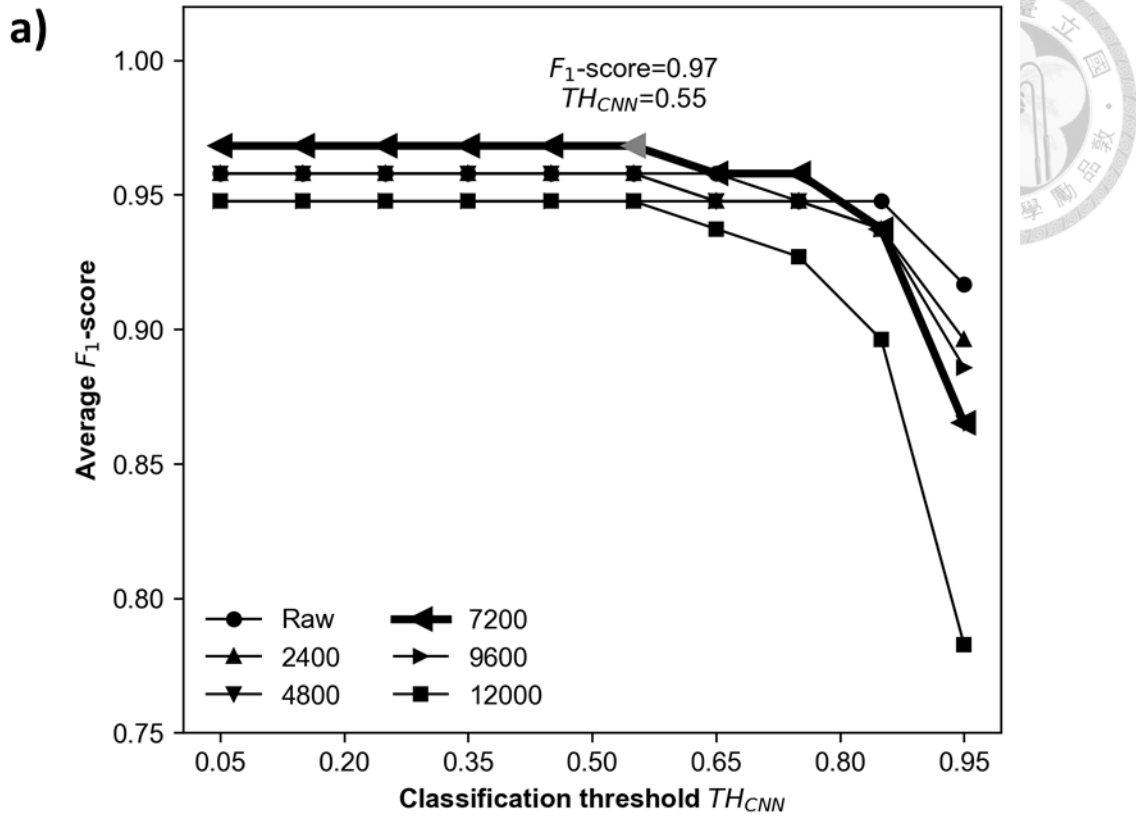


Fig. 4-2. Indoor insect vs. non-insect model validation and testing results: a) Classification threshold tuning; b) Object-level testing confusion matrix without augmentation; and c) Object-level testing confusion matrix with augmentation (7200 samples).

shows that the models were robust enough to filter out the non-insect objects. Having a high insect vs. non-insect model F_1 -score was very important since it prevents false positive classification of unknown detected objects.



The NMS threshold of the object detector model was tuned as it was cascaded with the insect vs. non-insect model. It was selected as the main tuning parameter since it controls the number of objects detected by the object detector model. NMS threshold values of 0.05, with increments of 0.10, were selected for tuning. The tuning results are shown in Fig. 4-3. The effect of filtering out the non-insect objects from the object detector model detections by the insect vs. non-insect model was quite noticeable. In Fig. 4-3a, the false positive rate of the object detector model was quite high without the insect vs. non-insect model. This was the issue to be solved since the object detector model was too sensitive and there were too many excess detections. The excess detections usually included overlapping detections of single objects and objects found at the edges of the image. With the insect vs. non-insect model, the false positives were almost completely avoided. This caused an increase in the miss rate which was found to be negligible since the reliability of detecting the insect objects increased with an F_1 -score of 0.93. The results showed that using a high NMS threshold led to more false positive detections. Based on the tuning curve, an NMS threshold of 0.15 was set since it yields the highest F_1 -score with very low false positive rate and minimal miss rate.

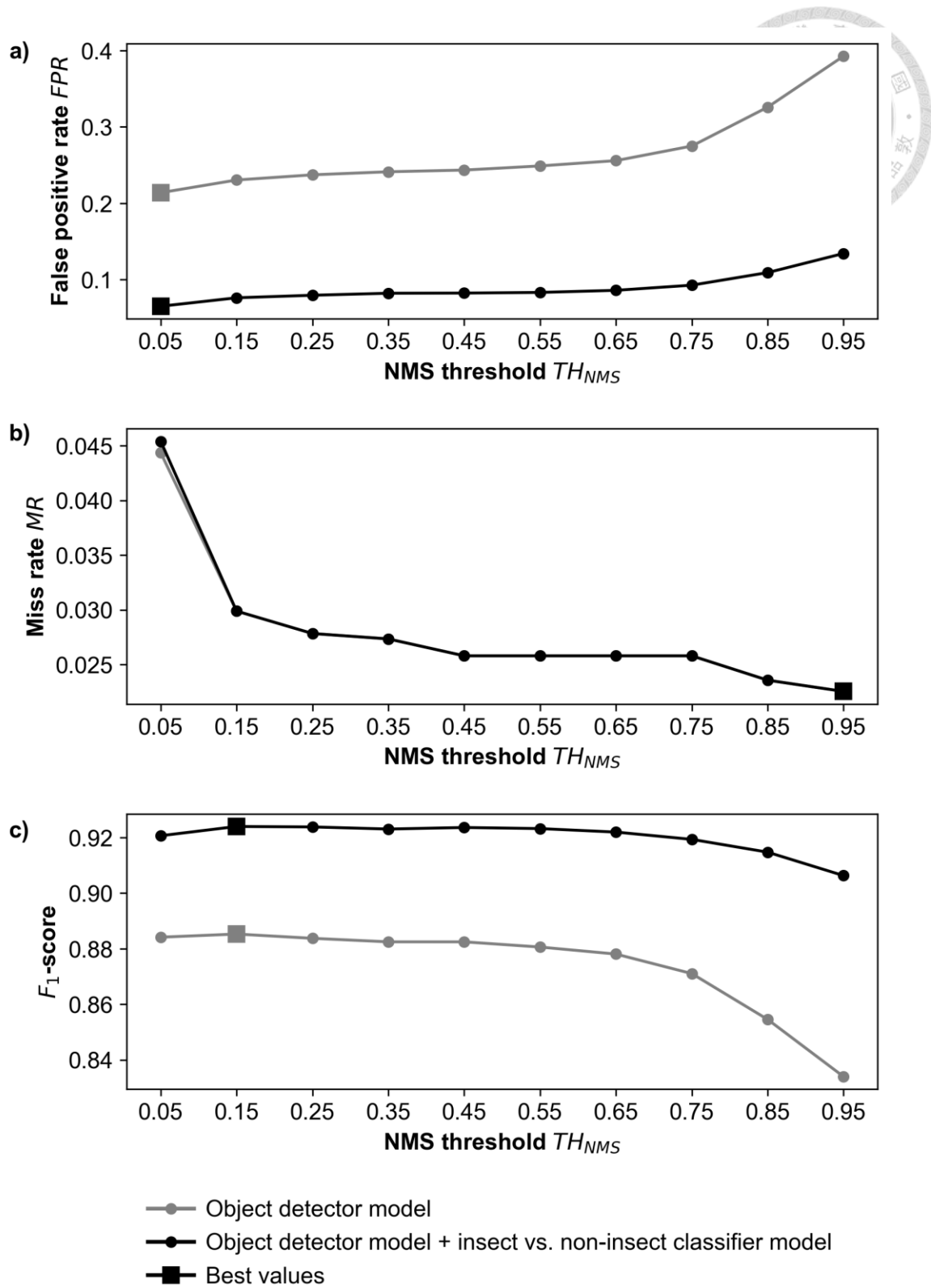


Fig. 4-3. NMS threshold tuning curves of the indoor insect object detector: a) False positive rate; b) Miss rate; c) F_1 -score.



4.1.1.2 Outdoor object detector and insect vs. non-insect model

The outdoor object detector was trained with the samples of Farm M1. Training was done with similar training parameters as the indoor object detector. The training results are shown in Fig. 4-4. It shows that the outdoor object detector had relatively lower *mAP* of around 0.88 compared to the indoor object detector. This was due to the foreign objects stuck in the sticky paper trap images that caused false positive detection of non-insect objects. Based on the training curve, both validation loss and *mAP* saturated quickly at 100 epochs. This means that there were not too many training images to train on. Finally, the object detector model trained with 100 epochs was selected as the final object detector model.

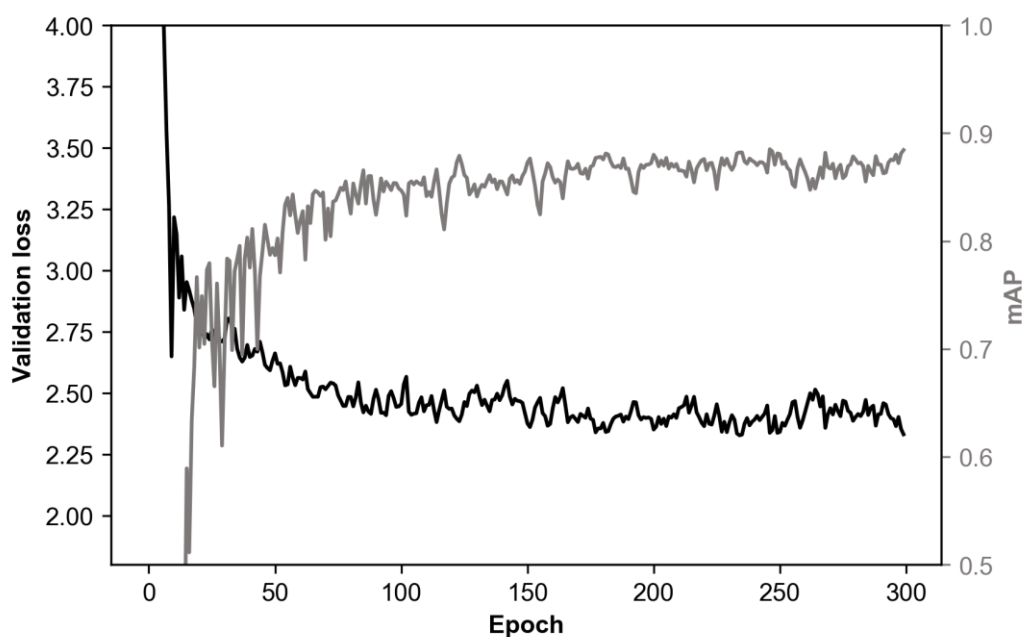
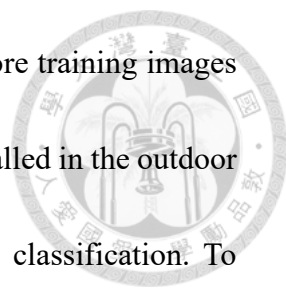


Fig. 4-4. Outdoor object detector model training curve.



The object detector model can be improved in the future, if more training images can be collected. But due to the limited number of sensor nodes installed in the outdoor sites, it was necessary to filter out the non-insect objects before classification. To resolve this issue, an insect vs. non-insect model was trained. Since there was a limited number of non-insect images collected, data augmentation by rotation, horizontal flipping and vertical flipping was applied. The number of training samples of each class was balanced by increments of 1600. Training was carried out for 200 epochs, with 20 steps per epoch, and with a learning rate of 0.005, using Adam optimizer for minimizing the validation loss. The model with the lowest validation loss and highest validation accuracy was selected as the final model. Each trained model was validated through stratified k -fold validation, with k set as 5. The classification threshold TH_{CNN} was also tuned with values from 0.05 to 0.95, with increments of 0.05. The model validation results are shown in Fig. 4-5a.

It can be seen from the results the trained model was able to achieve an average validation F_1 -score of 0.95. It also shows that data augmentation did not contribute much in improving the classification performance of the trained model. But as the object level testing confusion matrix of the trained model without augmentation was obtained, it shows that it was overfit to the insect class (Fig. 4-5b). This was due to the imbalanced number of samples of each class; with the insect class having more than

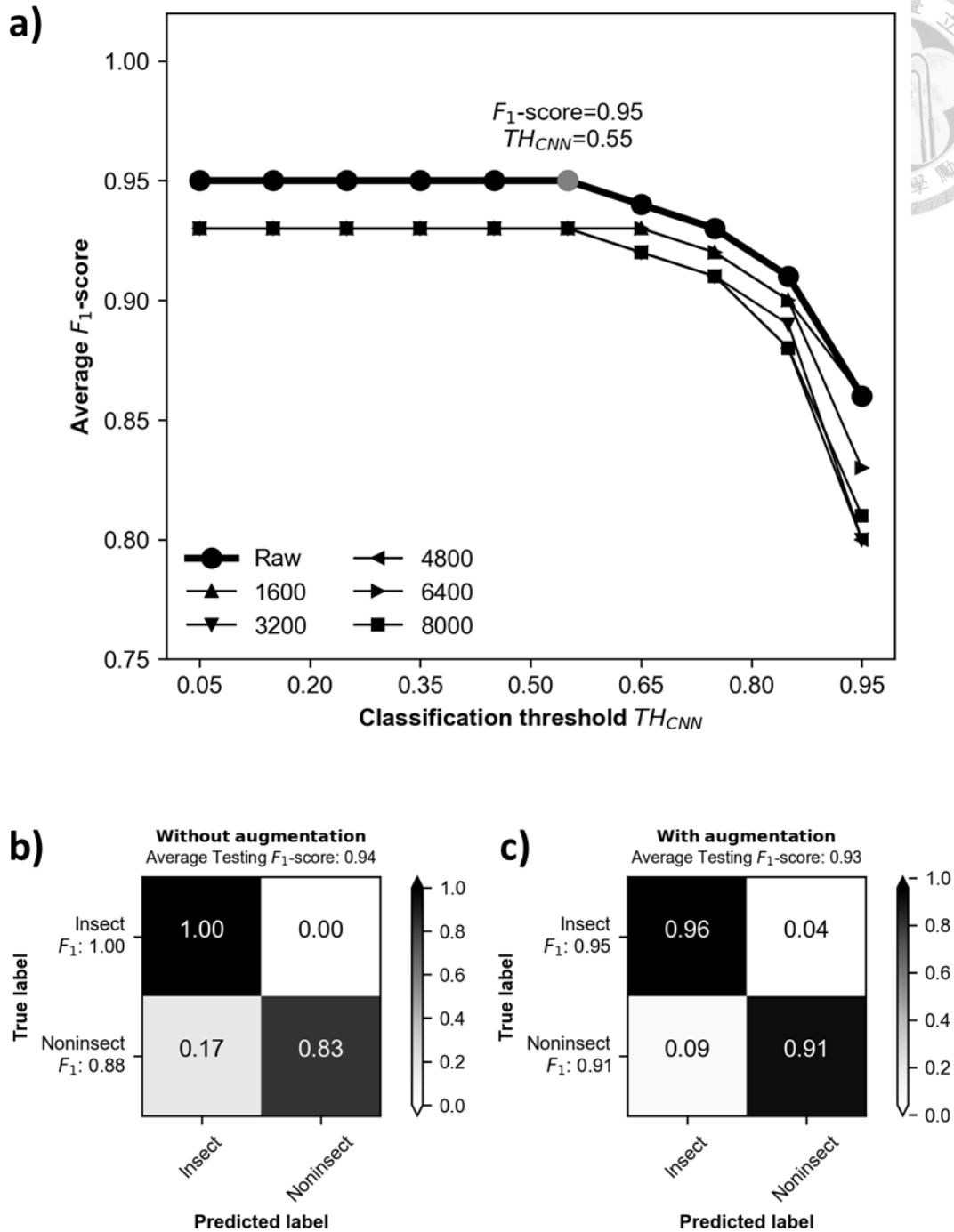


Fig. 4-5. Outdoor insect vs. non-insect model validation and testing results: a) Classification threshold tuning; b) Object-level testing confusion matrix without augmentation; and c) Object-level testing confusion matrix with augmentation (8000 samples).

twice the number of samples of the non-insect class. On the other hand, the trained model with 8000 augmented samples showed that the classification performance for each class was balanced (Fig. 4-5c). This shows that data augmentation was able to resolve potential overfitting issues caused by imbalanced number of samples. Therefore, the trained model with 8000 augmented samples was selected as the final model, with its classification threshold TH_{CNN} set to 0.55.

The NMS threshold of the outdoor object detector model was tuned in a similar manner as the indoor object detector model. The tuning results are shown in Fig. 4-6. It can be seen from the results that the insect vs. non-insect model was able to greatly reduce the false positive rate of the insect object detector to around 0.05 (Fig. 4-6a). This had a minimal effect to the miss rate but it was negligible and the F_1 -score also increased up to around 0.92. The tuning curve also shows that a very low NMS threshold value can lead to more missed detections. But considering the goal of the insect object detector model, an NMS threshold of 0.15 was selected since it yields a balance of all the evaluation metrics. The validation and tuning results show the importance of the insect vs. non-insect model in improving the over-all performance of the algorithm, especially for the outdoor site that had a lot of foreign objects stuck in the sticky paper traps.

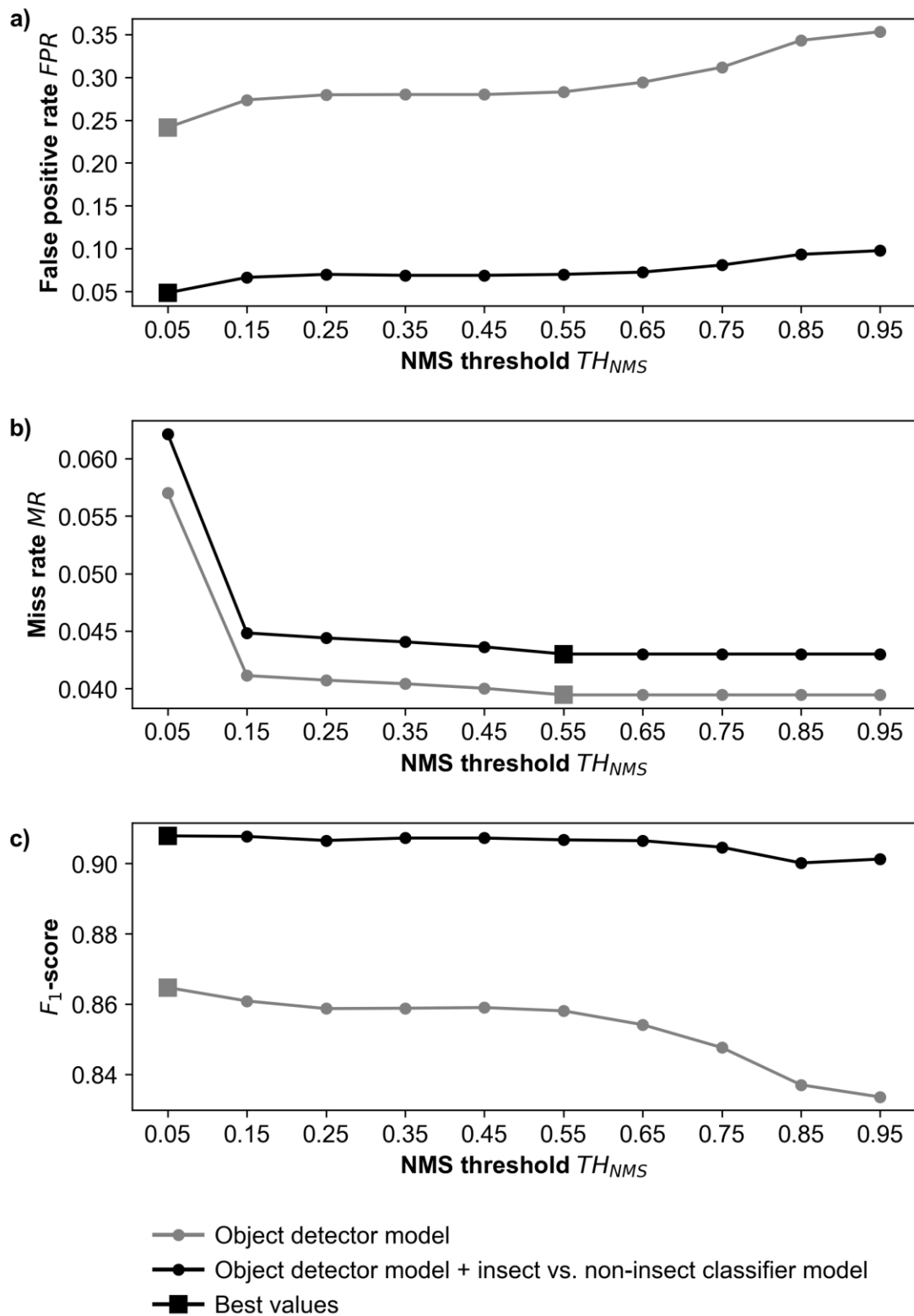


Fig. 4-6. NMS threshold tuning curves of the outdoor insect object detector: a) False positive rate; b) Miss rate; c) F_1 -score.

4.1.2 Multi-class insect classifier model



The multi-class insect classifier models for indoor insects and outdoor insects were trained in two different ways. The indoor insect multi-class insect classifier model was trained using the presented semi-supervised learning method. The primary goal in applying the semi-supervised learning method was to be able to continuously collect new image samples of classifier target classes with few samples. The multi-class insect classifier model should improve over time. One pre-requisite of the semi-supervised learning method is that the number of image sources for pseudo-labelling is adequate to improve the model after each semi-supervised learning cycle. Unfortunately, there was currently just one outdoor installation site to collect the images from. After performing preliminary tests, the semi-supervised learning method barely improved the outdoor multi-class insect classifier base model after each semi-supervised learning cycle. Moreover, the outdoor multi-class insect classifier consisted of more than one cascaded classifier. However, the current semi-supervised learning method was not yet optimized for this kind of setup. The stated reasons justify the differentiation between the learning methods for the outdoor and indoor insect classifiers. This section discusses about the training, validation, testing, and optimization steps performed to develop the multi-class insect classifiers.

4.1.2.1 Semi-supervised indoor multi-class insect classifier model



A BM was trained using the training images, 2 years since the system was running.

The semi-supervised model learning method was started during this time since there was already an adequate number of samples per insect class to train the model. Training of the BM was done for 200 epochs, with 20 steps per epoch, and with a learning rate of 0.0001, using Adam optimizer to minimize the validation loss. The BM was validated every epoch and the model with the lowest validation loss and highest validation accuracy was selected as the final model. The validation results are shown in Fig. 4-7.

The model validation results show that the BM had a good classification performance with a best validation F_1 -score of 0.85. This was expected since some of the classes defined for training did not have too many training samples that will balance the performance of the model. This justifies that the semi-supervised learning method can be applied to continuously retrain the model as new samples of the other insect classes were received. A classification threshold of 0.45 was set based on the tuning curve.

The trained BM was also tested in an object level, as shown in Fig. 4-7b. The results clearly show that the base model had difficulty recognizing craneflies,

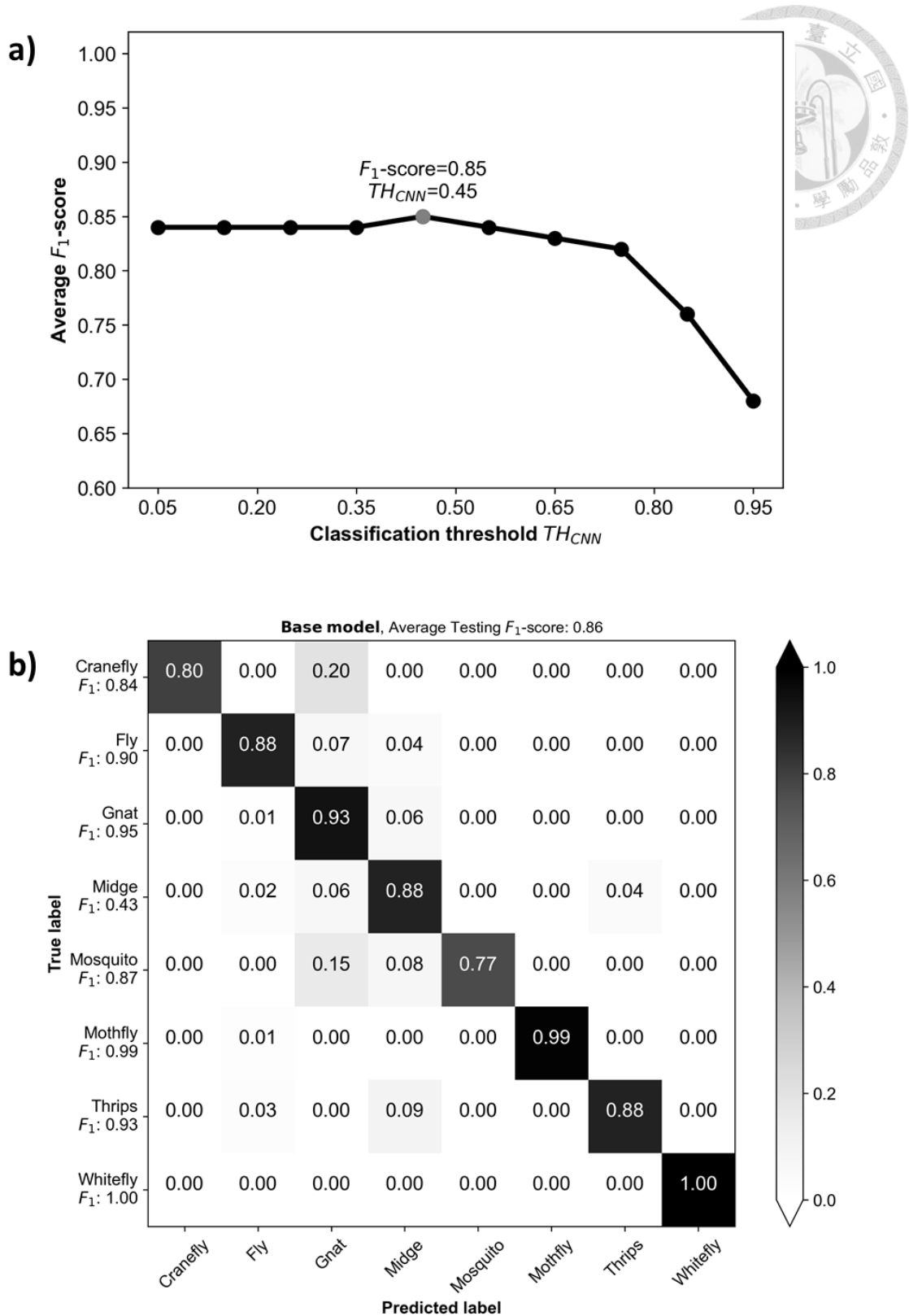
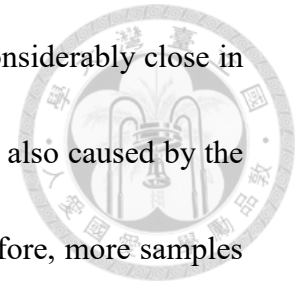


Fig. 4-7. Indoor multi-class insect classifier base model performance: a) Model

validation results; b) Object-level testing results using the best model.

flies, midges, mosquitoes, and thrips. The four insect pests were considerably close in appearance; which caused the confusion in classification. This was also caused by the lack of image samples of the crane fly and mosquito classes. Therefore, more samples should be collected to improve the classification performance.



To prepare for applying the semi-supervised learning technique, the feature projection of the samples was analyzed. The base model was used to extract the features of the training images and were encoded using PCA into two principal components. The results are shown in Fig. 4-8. The feature projection shows that there were at least two easily distinguishable classes: mothfly and whitefly. This also explains why the F_1 -score of the two classes was very high. It can also be seen that the features of the other classes were overlapping with each other. It shows that the model should be trained to know the difference between the five classes.

The semi-supervised learning technique was applied to automatically collect more samples of the weaker classes to improve the over-all algorithm performance. In so doing, each semi-supervised learning cycle was set to every month. On each cycle, the sticky paper trap image of each sensor node in each location with the most insects was selected automatically with the assistance of the object detector model and insect vs. non-insect model. The sticky paper trap images were filtered out by ignoring the images with less than 20 insects. A classification threshold TH_{CNN} of 0.8 was also set for the

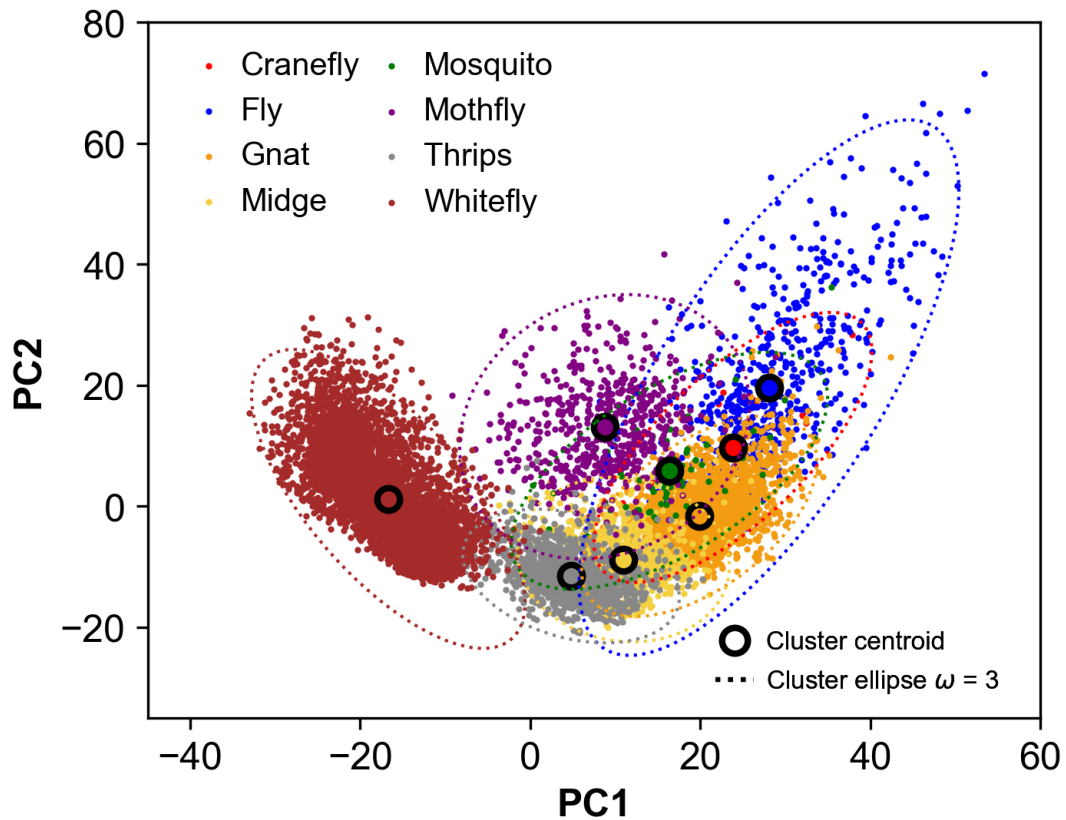


Fig. 4-8. Two-dimensional PCA feature projection of the indoor multi-class insect classifier base model.

insect vs. non-insect model to prevent collection of non-insect objects images.

The collected insect images were then sorted using the proposed unsupervised pseudo-labelling algorithm. The scaling factor of the sample cleaning step was set to $\omega = 3$, to remove outliers. It was tested from previous trials that the ω value of 3 achieved the most reliable and efficient results. Sample results of the pseudo-labelling algorithm are shown in Fig. 4-9.

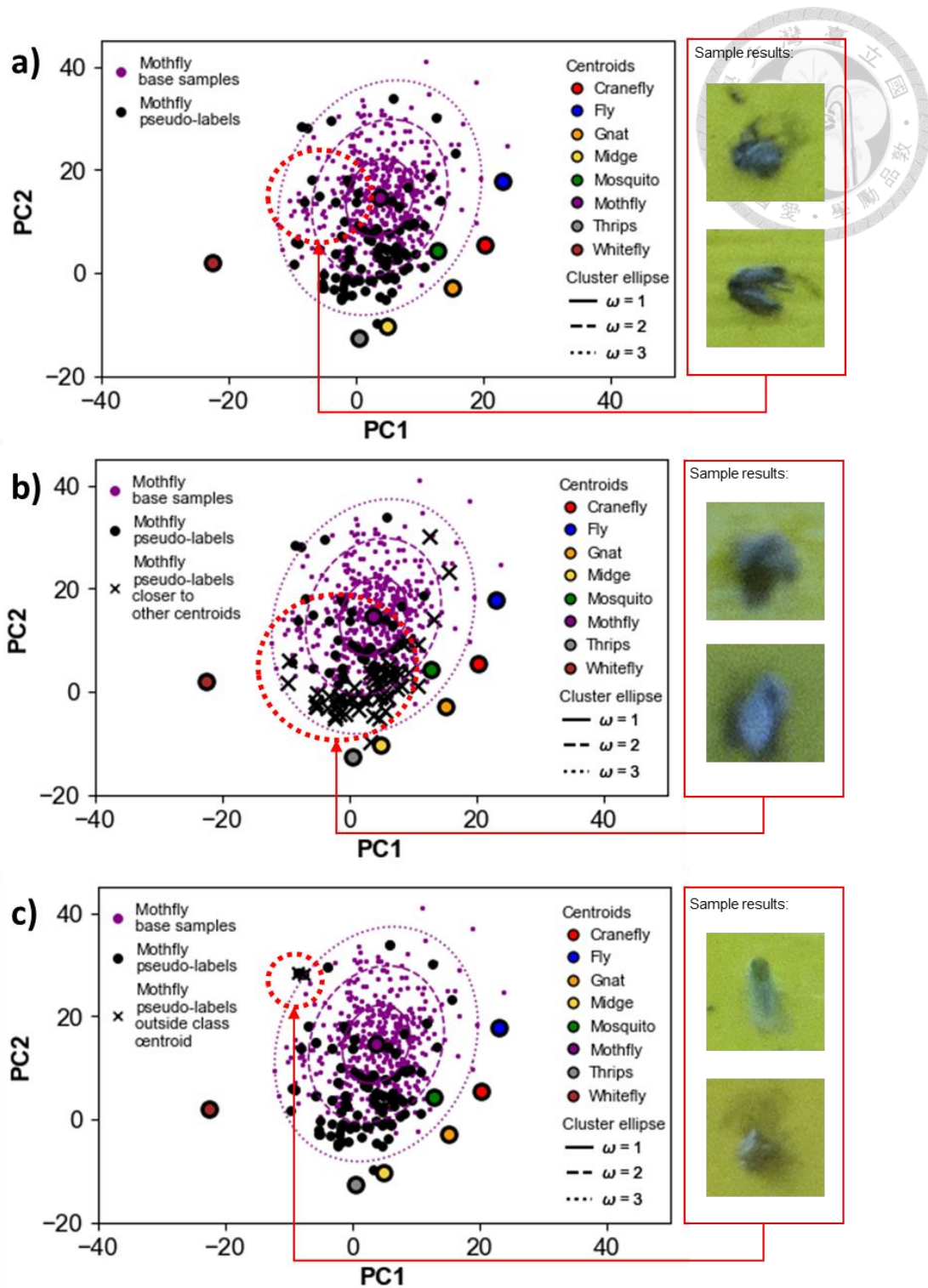
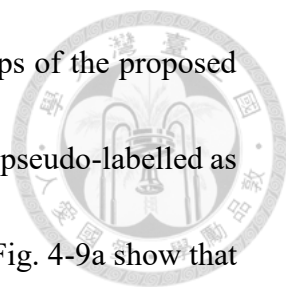


Fig. 4-9. Sample results of pseudo-labelling mothfly image samples from images of Farm TS1: a) Pseudo-labelled images via base model CNN classification; b) Images closer to another centroid; and c) Images outside cluster ellipse $\omega = 3$.



The results shown in Fig. 4-9 demonstrates the three major steps of the proposed pseudo-labelling algorithm. First, the incoming image samples were pseudo-labelled as mothfly using the base model. The sample image results shown in Fig. 4-9a show that the samples were correctly pseudo-labelled as mothfly. Actually, the sample image results show typical mothfly images. But in Fig. 4-9b, some pseudo-labelled samples were found closer to the other class centroids. It appears that such samples were occasionally blurred samples that may only cause confusion to the classifier. Finally, the sample cleaning step was able to find outlier mothfly images, as shown in Fig 4-9c. It can be noticed that the images were not an image of a mothfly but were of a non-insect or of an unknown class. This shows that the pseudo-labelling algorithm effectively selected clean samples that can possibly be added to the current training set and improve the base model.

In each semi-supervised cycle, the cluster density and testing F_1 -score of each class were computed to evaluate the effectiveness of the semi-supervised learning method. This can also to which class was the adaptive model learning after each cycle. The results are shown in Fig. 4-10. Looking closely at the weaker classes found previously in Fig. 4-7b, it can be observed in Fig. 4-10 that there were very few additional crane-fly samples collected by pseudo-labelling. This was because crane-flies were rarely found in each farm. However, it can be seen that new samples of crane-fly were collected

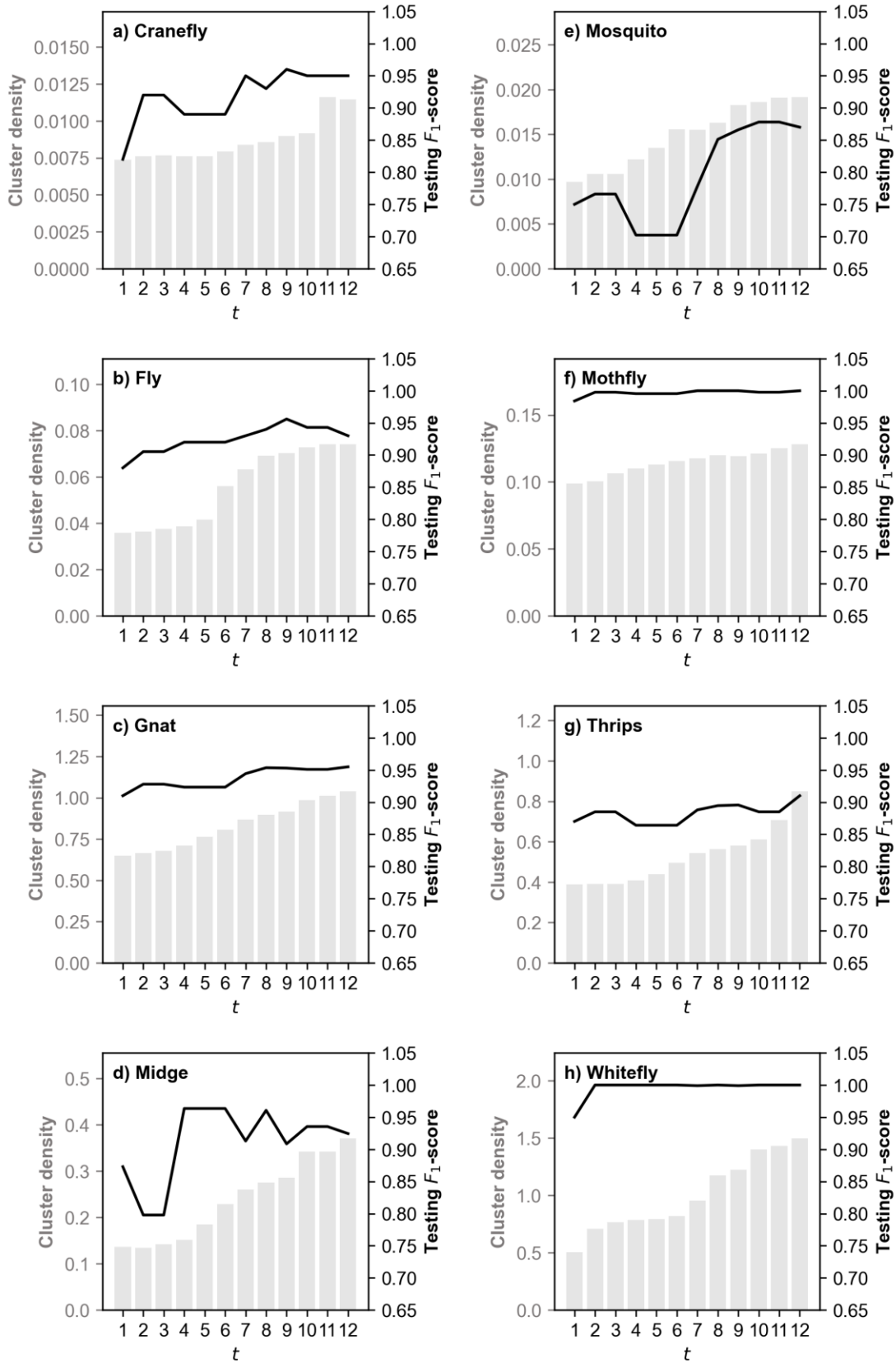
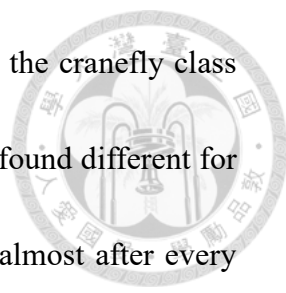


Fig. 4-10. Temporal semi-supervised learning results per class.



starting at $t = 6$. The F_1 -score of the adaptive model in classifying the crane fly class slowly increased until it stopped at around 0.95 at $t = 12$. This was found different for the mosquito class; new mosquito training images were collected almost after every time t . Although, it can be noticed that there was a sudden drop at $t = 4$. It was found that a few misclassified mosquito samples were added to the training dataset. But as soon as more samples were collected, it slowly increased again until the final testing F_1 -score was 0.86. Even though the F_1 -score was still low, it was still an improvement from 0.77 of the base model. The performance of the adaptive model also progressed slowly for the rest of the classes. It can also be seen that there was almost no improvement in performance of the stronger classes, moth fly and white fly. Despite this, it means that the other classes were unaffected by the new samples collected.

The temporal improvement of the adaptive model, based on the semi-supervised learning online and offline metrics, was analyzed, as shown in Fig. 4-11. Supervised learning was also carried out for comparison. It can be observed that the total number of samples collected at each semi-supervised learning cycle varied (Fig. 4-11a). This was because the availability of new samples depended on the number of good quality sticky paper trap images collected. The pseudo-labelling accuracy was also measured by inspecting the clean and noisy pseudo-labels obtained at each cycle. It shows that the pseudo-labelling accuracy ranged from 0.85 to 0.95 (Fig. 4-11b), with an average

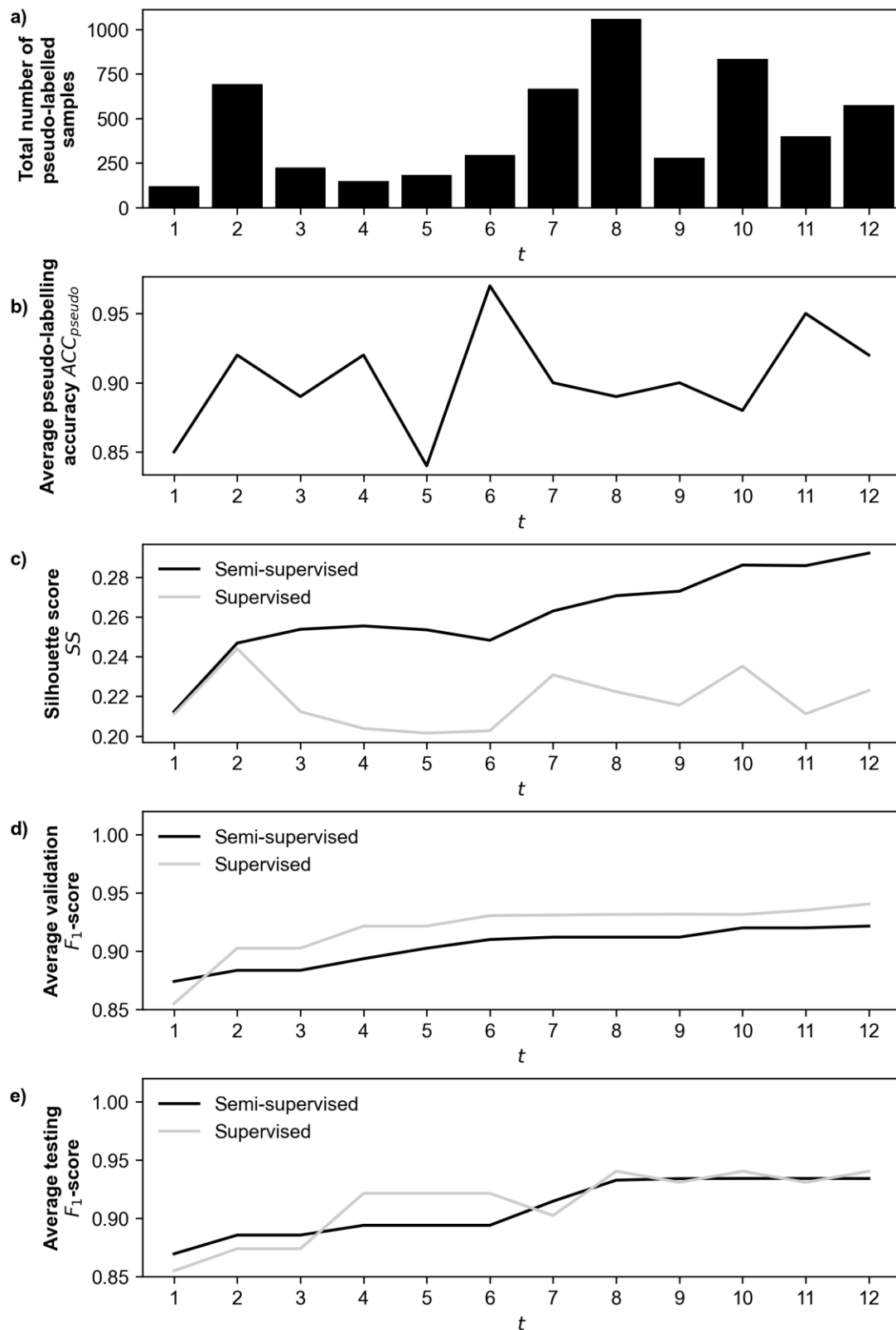
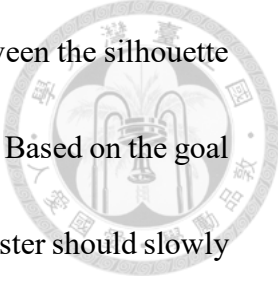


Fig. 4-11. Temporal evaluation of the adaptive model based on the semi-supervised learning online and offline metrics.



of 0.91, based on 12 cycles. Interestingly, there was a difference between the silhouette score values of semi-supervised and supervised learning (Fig. 4-11c). Based on the goal of the pseudo-labelling algorithm, the distance between each class cluster should slowly separate since new uncommon clean training samples were collected at each cycle. But in supervised learning, all the correct samples were added to the training dataset. Apparently, this caused the clusters to draw closer since the model was slowly overfit due to redundant samples collected. This showed that the pseudo-labelling algorithm was effective to reduce the collection of redundant samples to prevent model overfitting.

The average validation and testing F_1 -scores at each cycle was also recorded for both semi-supervised learning and supervised learning. It shows that the validation F_1 -score of the model trained by supervised learning was higher than the model trained by semi-supervised learning. However, it can be observed in Fig. 4-11e that the high validation F_1 -score did not always reflect a high testing F_1 -score. This may indicate that the model trained by supervised learning was slowly overfitting and did not perform well on images outside the training dataset. This has proven that the semi-supervised learning was able to avoid the issue of overfitting. Moreover, it also shows that the semi-supervised learning method was a feasible alternative to supervised learning since the model F_1 -scores of both methods at each cycle were not too different.

It can also be noticed that the F_1 -score of the semi-supervised adaptive model never

dropped. This was due to the model selection criterion that prevented contamination of the current base model. As soon as an unsupervised model with lower F_1 -score was validated, the model was discarded and the pseudo-labelled training samples obtained during that cycle was not added to the current training dataset. This shows that the model selection method was useful in avoiding system issues due to training of a corrupted adaptive model.

The object level testing results of the trained semi-supervised and supervised models are shown in Fig. 4-12 and Fig. 4-13, respectively. One main difference can be seen from both confusion matrices, the supervised model indicates overfitting in several classes. The supervised model had F_1 -scores very close to 1.00 in classifying the fly class and gnat class. Unfortunately, this has caused the classifier model to incur more errors in classifying the crane fly and mosquito class. On the other hand, the semi-supervised model showed balanced F_1 -scores for most of the classes. However, it can be seen that the semi-supervised model can still be improved further as soon as additional samples were collected. It was still slightly inaccurate in recognizing the fly, mosquito, and thrips class. This also shows that an adaptive model was necessary to continuously learn from the images collected by the system.

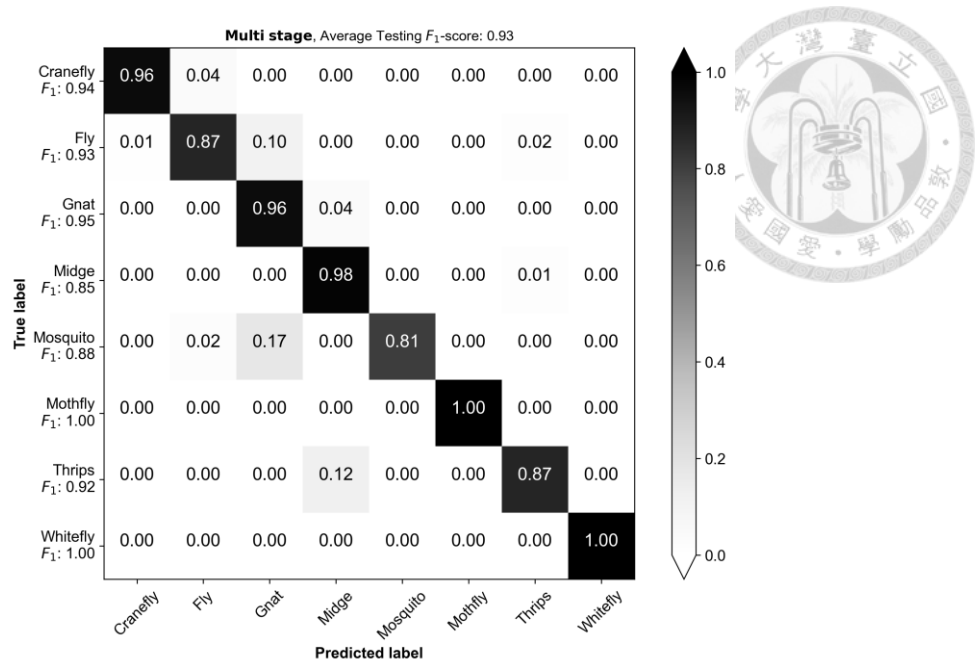


Fig. 4-12. Semi-supervised indoor multi-class insect classifier model object-level testing confusion matrix.

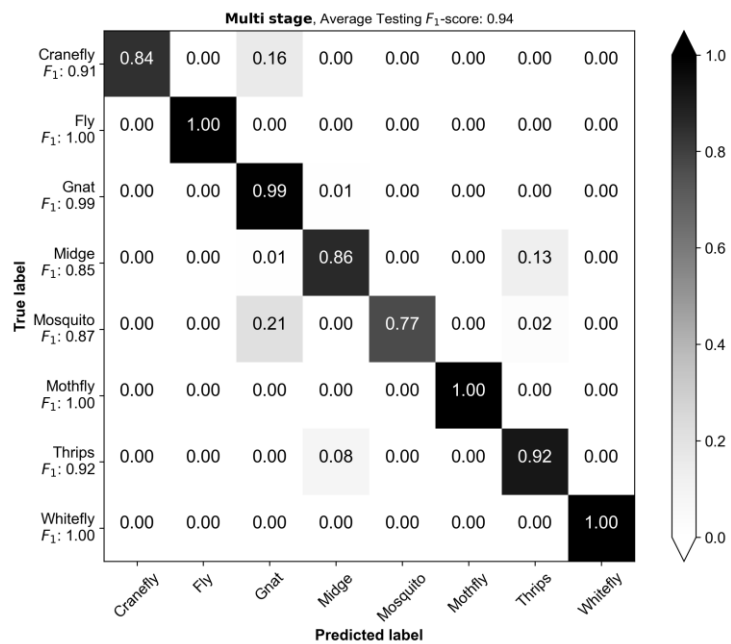
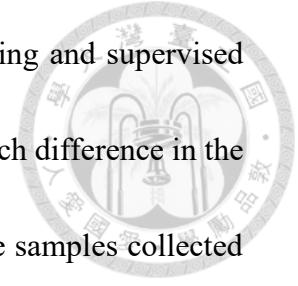


Fig. 4-13. Supervised indoor multi-class insect classifier model object-level testing confusion matrix.



The final total training images through semi-supervised learning and supervised learning are shown in Table 4-1. It was found that there was not much difference in the number of samples collected by both approaches. There were more samples collected by semi-supervised learning since there might be noisy samples also included. However, it was found that the noisy samples did not have detrimental effect to the performance of the trained adaptive model. In conclusion, the semi-supervised model was proven useful in automating the image data collection and model training routines in building the image classifier models used by the system.

Table 4-1. Final total number of training images of the indoor multi-class insect pest classifier model after applying semi-supervised learning.

Class	Base model	Final adaptive model (supervised)	Final adaptive model (semi-supervised)
Cranefly	25	53	47
Fly	359	773	619
Gnat	1824	1900	2366
Midge	356	656	711
Mosquito	47	58	69
Mothfly	440	550	580
Thrips	668	907	1199
Whitefly	1817	1959	2199
Total	5536	6856	7790

4.1.2.2 Supervised outdoor multi-class insect classifier models



Two classification strategies were tested for training the outdoor multi-class insect classifier model: single stage and multi-stage. The single stage strategy uses a single model to classify the insect pests while the multi-stage strategy applies the hierarchical classification scheme shown in Fig. 3-29. The purpose of this comparison was to show the benefits of applying the hierarchical classification scheme for multi-class classification of insect pests.

Since the number of training samples of each class was imbalanced, data augmentation by rotation, horizontal flipping and vertical flipping were applied. After augmentation, the number of training samples of each class was balanced by increments of 100. Training of the single stage model was carried out for 200 epochs, with 20 steps per epoch, and with a learning rate of 0.005, using Adam optimizer as validation loss minimizing function. After training each model, the model with the lowest validation loss and highest validation accuracy was selected as the final model. Each trained model was validated through stratified k -fold validation, with k set as 5. Model validation also included tuning of the classification threshold TH_{CNN} from 0.05 to 0.95, with increments of 0.05. The single stage model validation results are shown in Fig. 4-14a.

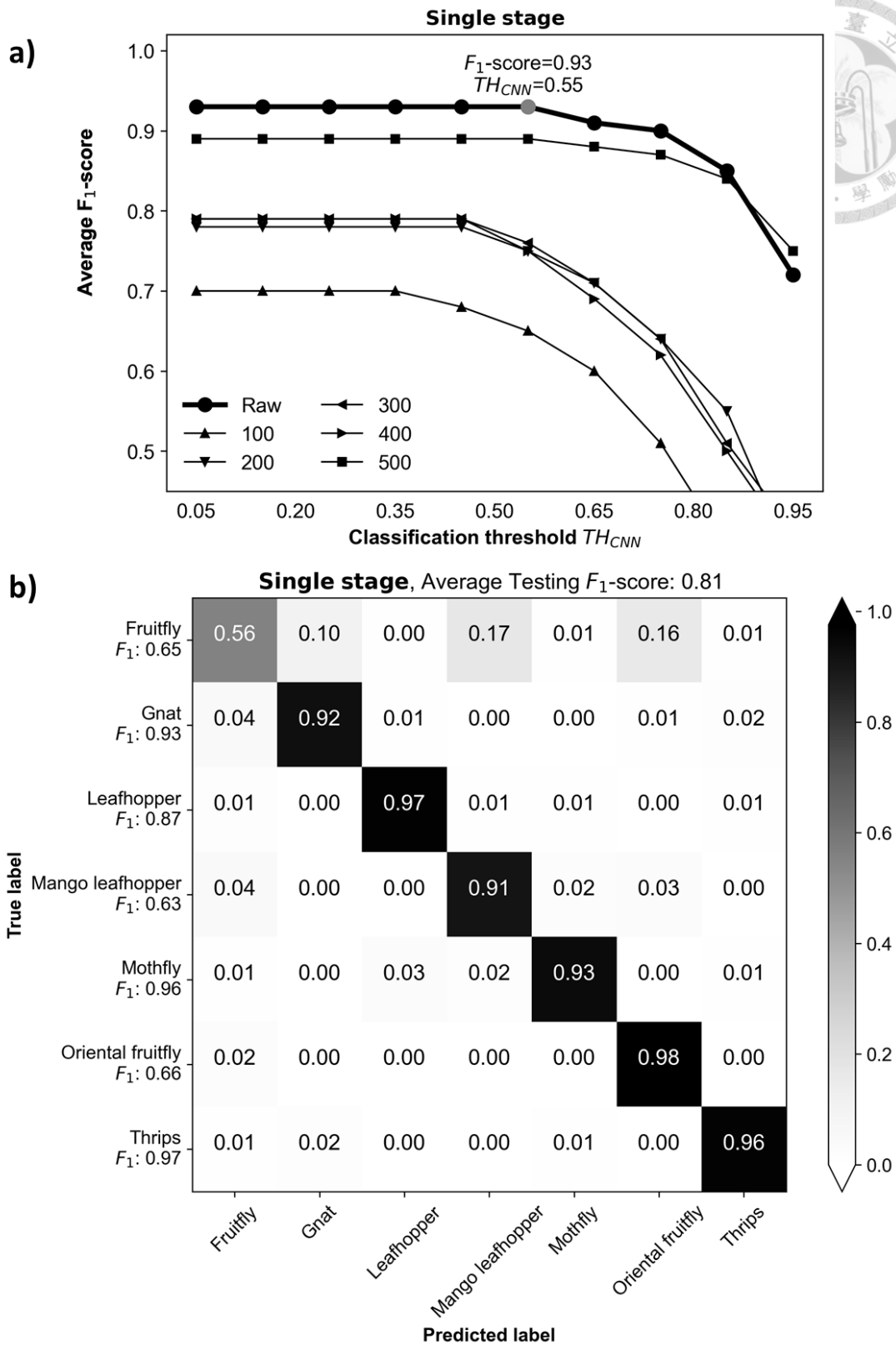
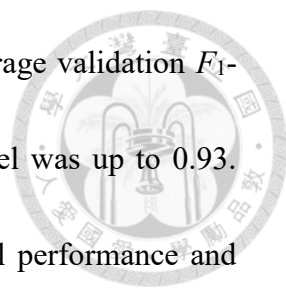


Fig. 4-14. Single stage outdoor multi-class insect classifier model performance: a)

Model validation results; b) Object-level testing results using the best model.



It can be seen from the results in Fig. 4-14a that the best average validation F_1 -score of the single stage outdoor multi-class insect classifier model was up to 0.93. Despite applying data augmentation, it did not improve the model performance and failed to exceed the performance of the model with the training set that consisted of non-augmented samples. Therefore, the trained model with raw samples was selected for testing. Based on the tuning curve in Fig. 4-14a, the final classification threshold TH_{CNN} was set to 0.55, such as to avoid an oversensitivity classification issue if the threshold value was too low. The testing results are shown in Fig. 4-14b. Unfortunately, it can be seen that the trained model performed poorly in classifying the testing set 2 images with an average testing F_1 -score of only 0.81. The confusion matrix shown clearly indicate that the trained single stage classifier model was confused in recognizing classes: fruitfly, gnat, mango leafhopper, and oriental fruitfly. The confusion was reasonable and expected since the four classes had similar appearances; black, medium to large in size, and had wings. Meanwhile, it was able to recognize the other insect classes.

Each model for multi-stage classification was trained with close training parameters as the single stage model. The steps per epoch was set to 10, and learning rate was set to 0.0001, adjusting to the few training samples. The validation results of the Stages 1, 2A, 2B, and 3 models are shown in Figs. 4-15 to 4-18, respectively.

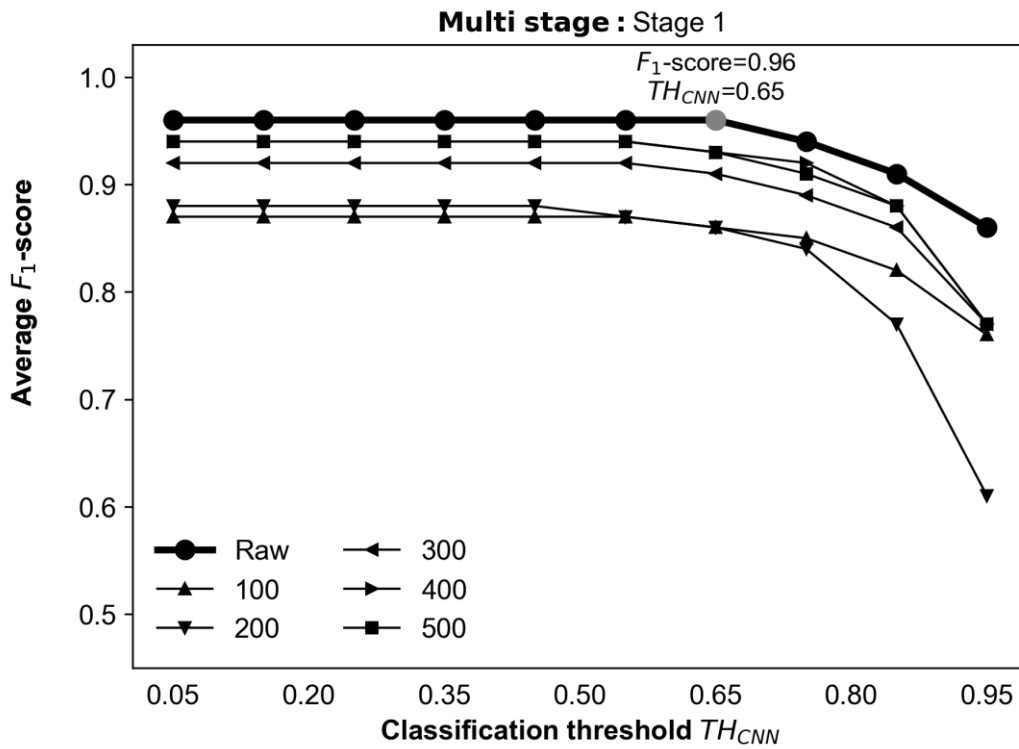


Fig. 4-15. Multi-stage outdoor insect classifier Stage 1 model validation results.

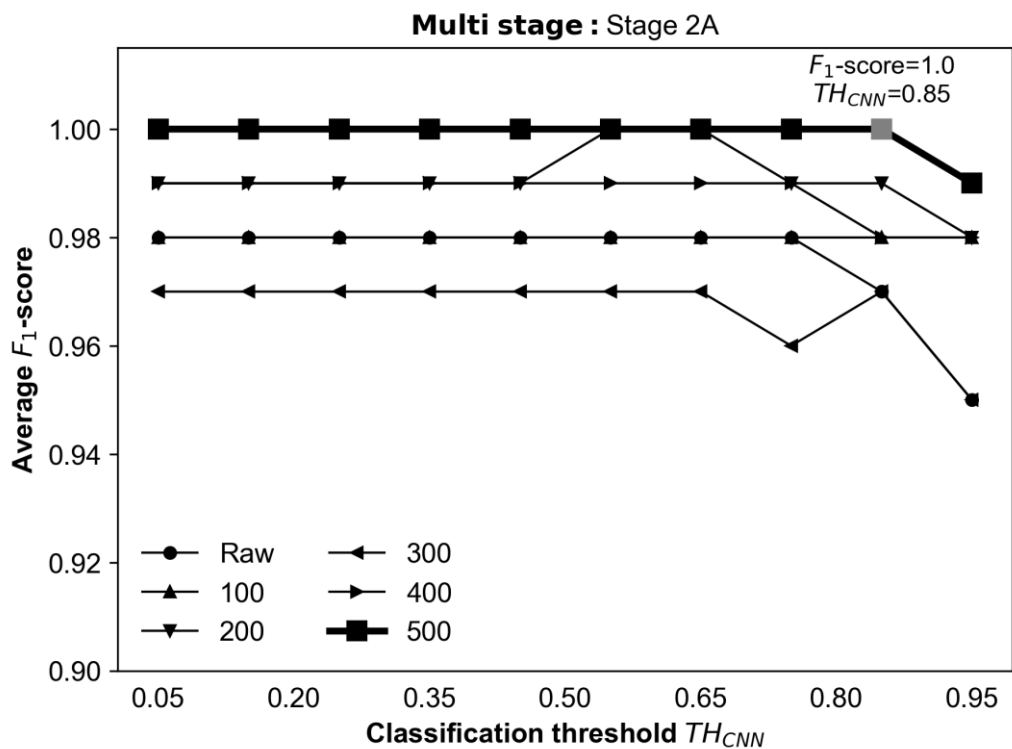


Fig. 4-16. Multi-stage outdoor insect classifier Stage 2A model validation results.

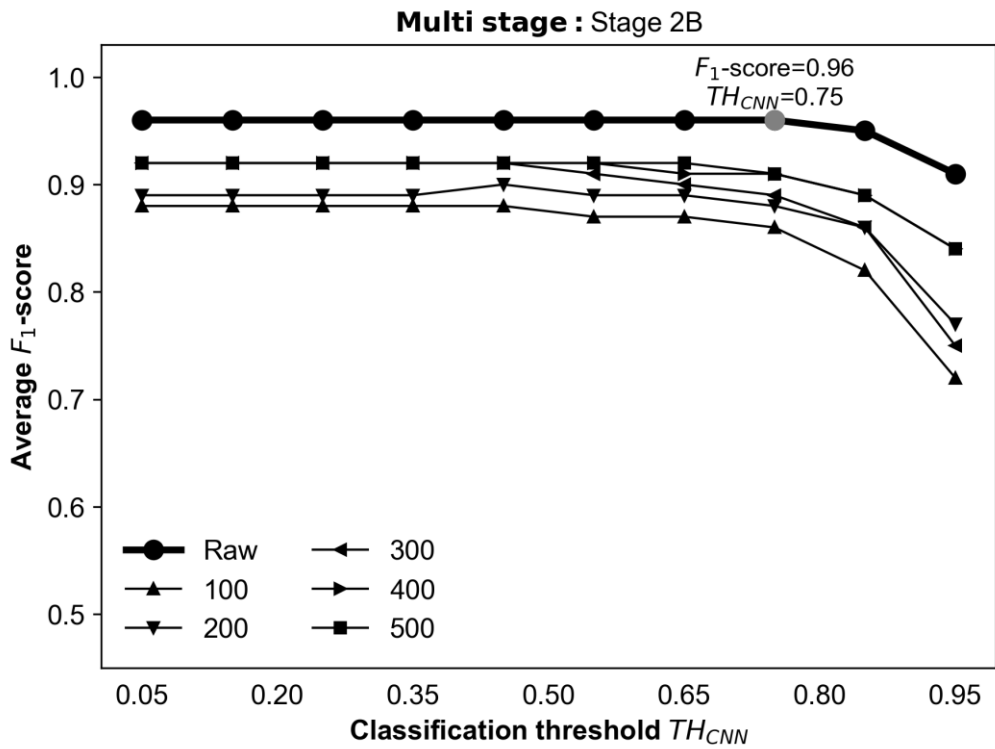


Fig. 4-17. Multi-stage outdoor insect classifier Stage 2B model validation results.

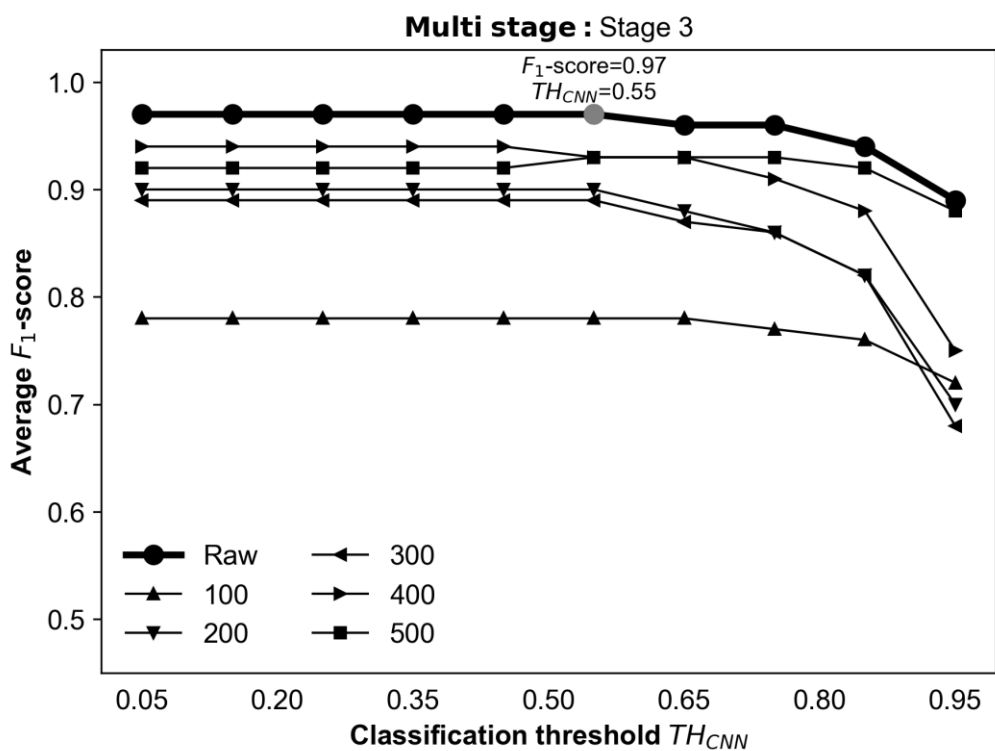
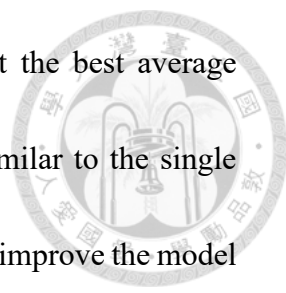
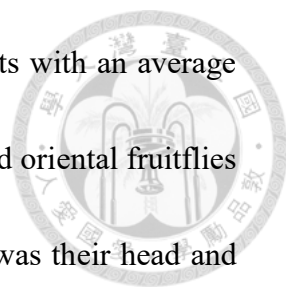


Fig. 4-18. Multi-stage outdoor insect classifier Stage 3 model validation results.



The model validation results of the Stage 1 model show that the best average validation F_1 -score of the Stage 1 multi-stage model was 0.96. Similar to the single stage model, applying data augmentation was still not able to further improve the model classification performance. This was likely caused by the few raw training samples of mango leafhopper and oriental fruitfly. From the current number of samples, not too many augmented samples can be generated. Still, the Stage 1 model classification performance was still high. A classification threshold TH_{CNN} of 0.65 was selected based from the tuning results.

The model validation results of the Stage 2A model in Fig. 4-16 show that classifying between mango leafhopper and leafhopper had no issues, as expected. The main reason for this was that mango leafhoppers were black while leafhoppers were green. Carrying out data augmentation also improved the classification performance. Therefore, the trained model with 500 balanced augmented samples was selected as the final Stage 2A model, with a classification threshold TH_{CNN} of 0.85. Meanwhile, the Stage 2B model, as shown in Fig. 4-17, also had good validation results with an average F_1 -score of 0.96. Applying data augmentation did not improve the classification performance of the model. However, the trained model with raw training samples was already enough; therefore, it was selected as the final model with a classification threshold TH_{CNN} of 0.75.



The final model, the Stage 3 model, had good validation results with an average F_1 -score of 0.97 (Fig. 4-18). This was unexpected since fruitflies and oriental fruitflies had very close appearances. The only difference between the two was their head and wing pattern. This shows that the classifier model was still able to detect unique features from the two classes. The trained model with raw training samples was selected as the final model, with a classification threshold TH_{CNN} of 0.55.

In average, the validation F_1 -score of the entire multi-class outdoor insect classifier was 0.97, which was already higher than the validation F_1 -score of the single stage classifier model. To further evaluate the performance of the multi-stage classification strategy, its confusion matrix was obtained as shown in Fig. 4-19. The results show that the Stage 1 model can still be improved since there were few misclassifications of mango leafhoppers into fruitfly. It also shows that there were issues in misclassification of fruitfly into gnat and vice-versa. This was expected since the images of the fruitfly class and gnat class were somehow similar in appearance. In summary, the classification performance of the multi-stage classification strategy was still satisfactory with an average testing F_1 -score of 0.91, a huge improvement from the average testing F_1 -score of the single stage model of 0.81. The results show that the multi-stage classification strategy was able to successfully boost the classification performance of the algorithm.

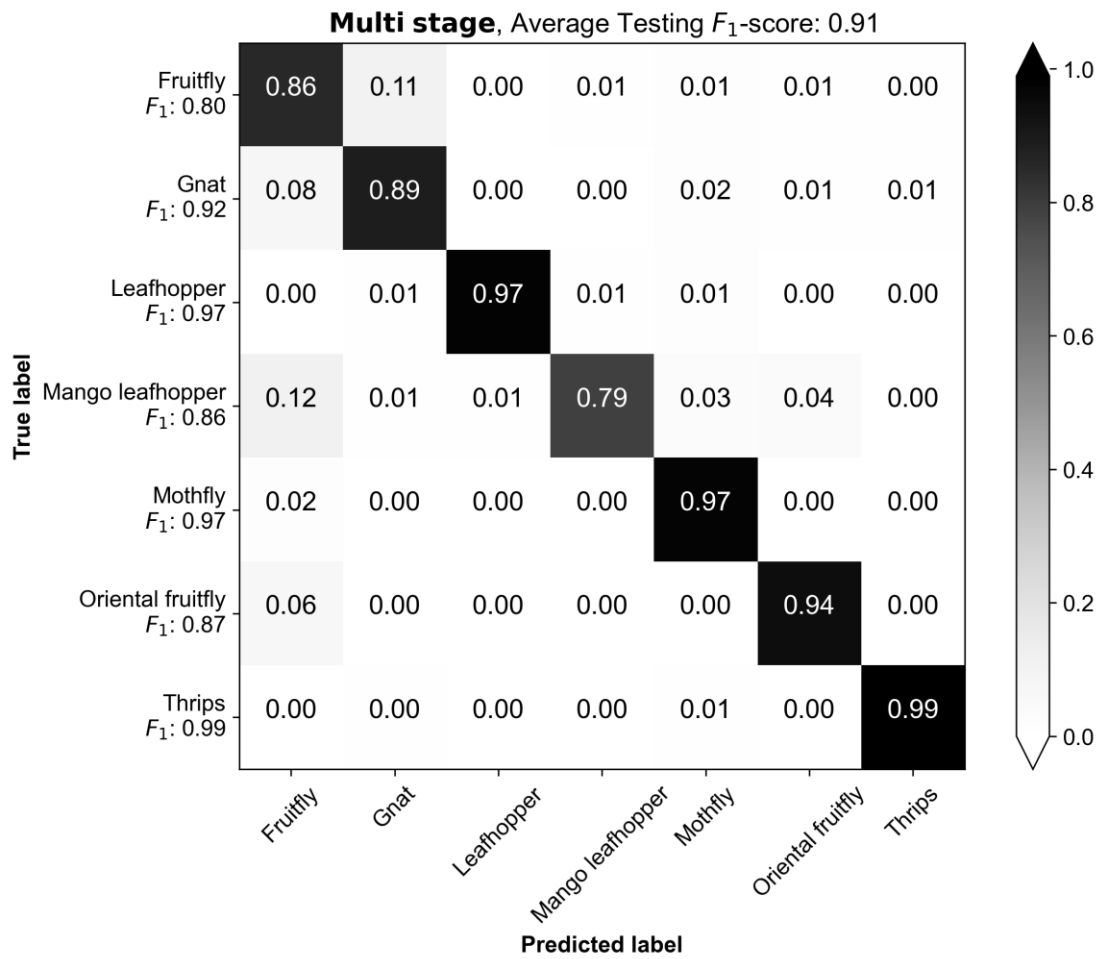


Fig. 4-19. Multi-stage multi-class outdoor insect classifier object-level testing confusion matrix results.

4.1.3 Image-level testing results



4.1.3.1 Indoor insect pest detection and recognition testing results

The sample processed sticky paper trap images of detecting and recognizing the indoor insect pests are shown in Fig. 4-20. The results of the object detector model (Fig. 4-20a) show that most of the insect pests were detected except for large insects that the model may have not seen yet. Unlike the small-sized insects, the large insect pests such as crane fly and mosquito do not appear often; causing missed detections. However, this can be easily solved in the future as the object detector model is retrained. The zoomed in image in Fig. 4-20a also shows that some insects were detected twice. This was seldom encountered since some insect objects appear as two objects if their body parts resemble other small insect pests such as thrips. The sample results in Fig. 4-20b shows this example in which the head of the gnat was wrongly detected as another insect object. The insect vs. non-insect model was able to correct this error as shown in Fig. 4-20b. In other cases, droplets were wrongly detected as insects and classified as whiteflies, dirt were classified as mothflies or thrips, and more. Through the insect vs. non-insect model, all of these issues were resolved since it was purposely trained to examine the objects detected by the object detector model.

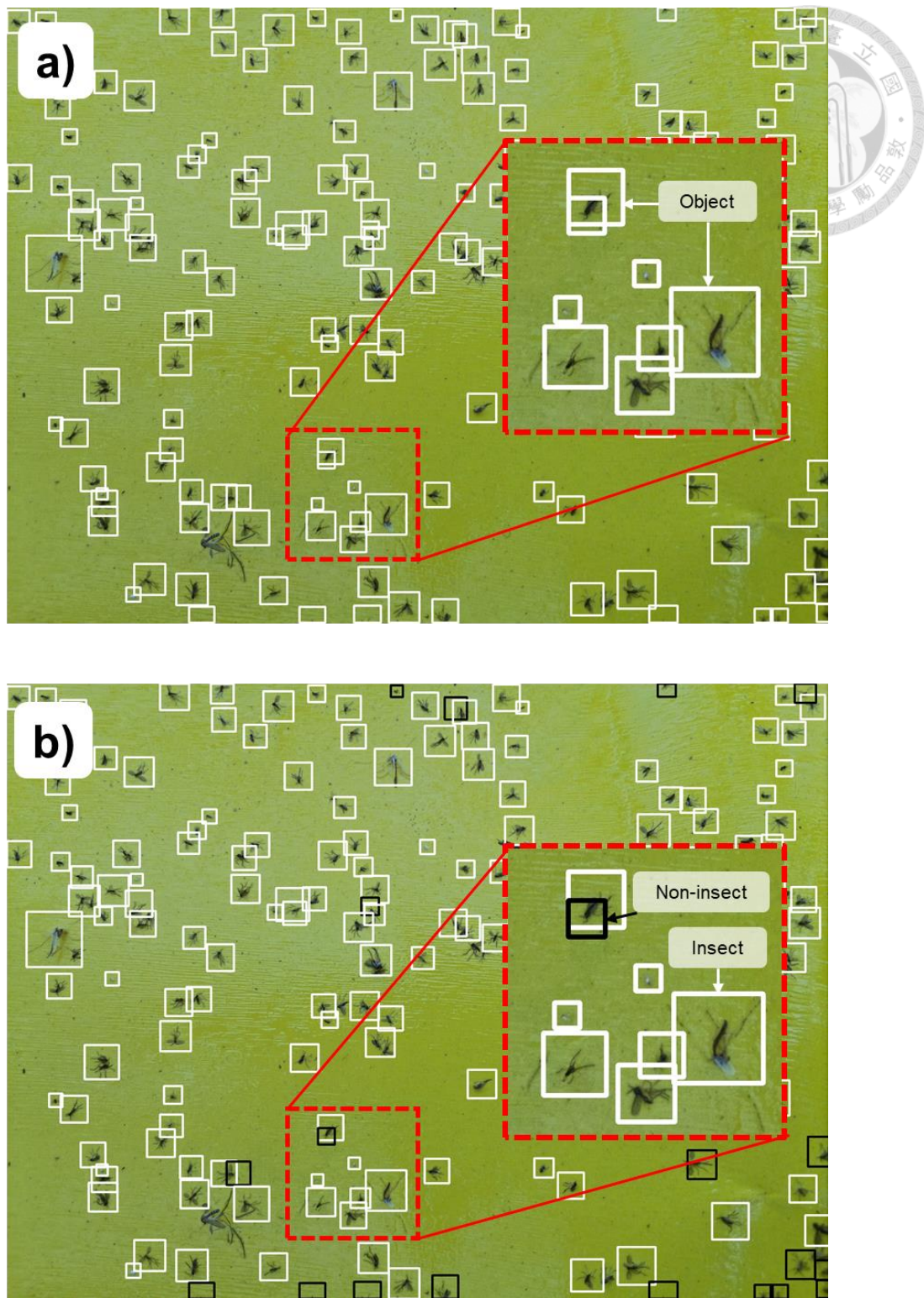
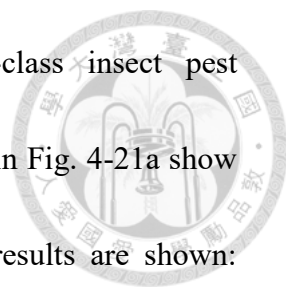


Fig. 4-20. Indoor object detector and insect vs. non-insect classifier model sample processed images: a) Image with detected images; b) Image with insect and non-insect images.



Following insect vs. non-insect classification was multi-class insect pest classification, as shown in Fig. 4-21. The sample processed image in Fig. 4-21a show the reclassified results of Fig. 4-21. Five sample classification results are shown: mosquito, gnat, mothfly, fly, and whitefly. One potential error in classification is shown in Fig. 4-21a, a mosquito classified as gnat. This was caused by the similar appearance of the two types of insect pests. However, this can still be solved as the classifier model improves through semi-supervised learning and additional training samples are collected. Meanwhile, Fig. 4-21b shows a sample processed image that contains cranefly, midge, and thrips. The sample processed image show that the algorithm was effective for any size of insect pests. It also shows that the multi-class insect pest classifier can properly distinguish between midges and thrips, which were very similar in appearance at times. One sample error is shown in Fig. 4-21b, classification of dirt to thrips. The thrips class was the most prone to error dealt by misclassification of non-insect objects. Hence, it is still recommended to maintain the cleanliness of the sticky paper traps to assure the performance of the algorithm. This means that the farm managers still have the responsibility in maintaining the condition of the sticky paper traps. The sample results illustrated show that the insect pest detection and recognition algorithm was able to successfully count the insect pests from the sticky paper traps, regardless of image complexity.

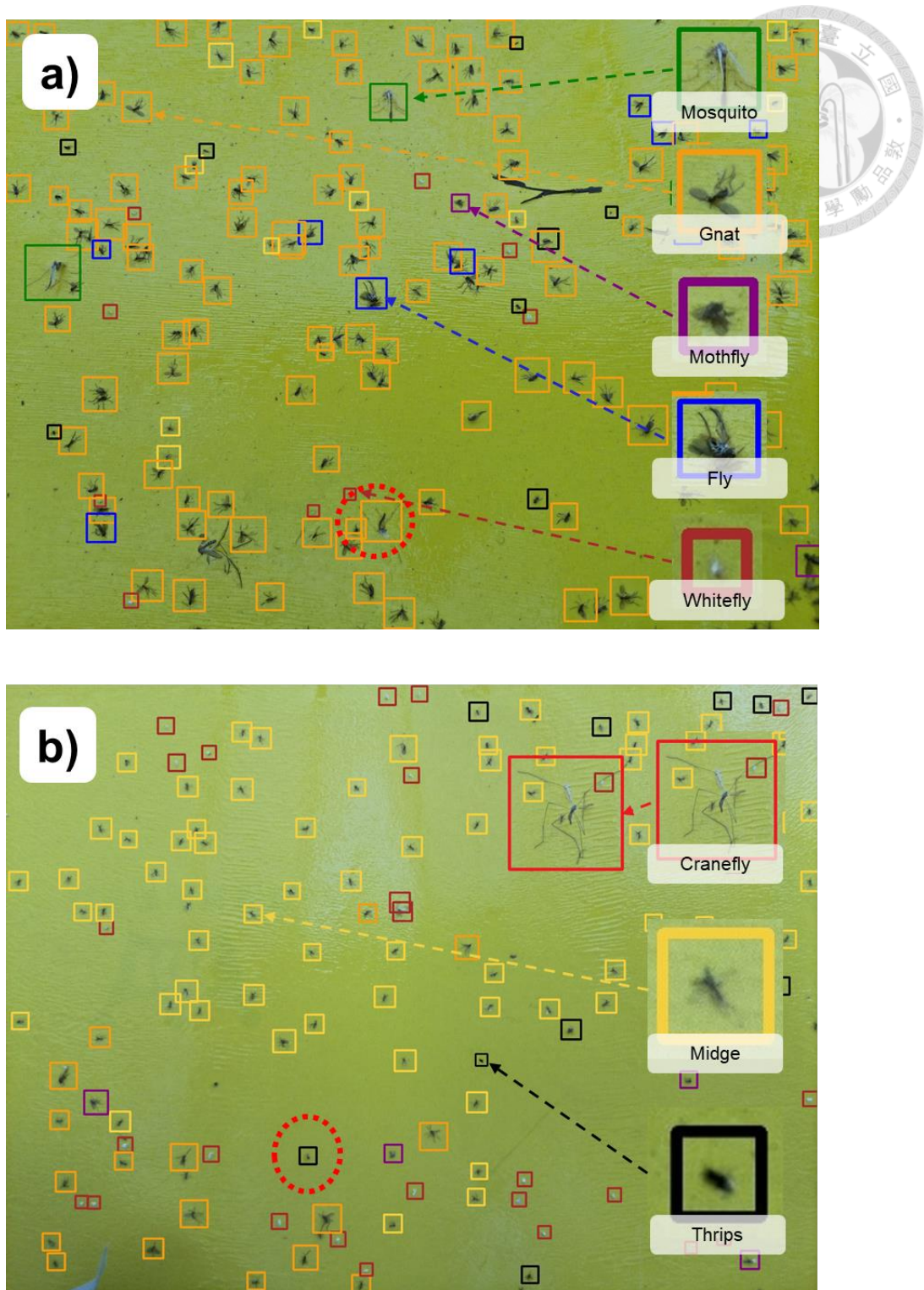



Fig. 4-21. Indoor multi-class insect pest classifier model sample processed images: a) Image with mosquito, gnat, mothfly, fly, and whitefly images; b) Image with cranefly, midge, and thrips images.



The insect pest detection and recognition algorithm, using the final best adaptive model trained by semi-supervised learning, was tested by image level using the images included in testing set 1. The scatter plots are shown in Fig. 4-22. It can be seen that the algorithm had an acceptable performance in counting craneflies and mosquitoes. But for the rest of the classes, the algorithm performed well with accuracies higher than 0.87, except for the whitefly class with only 0.81. The inaccuracy in counting whiteflies was caused by the presence of water droplets in the testing images. This has caused wrong detection of water droplets which were eventually classified as whiteflies. However, it still shows the algorithm was able to properly count even high number of whiteflies, more than 250 (Fig. 4-22h). The algorithm had slight overestimation of gnats (Fig. 4-22c) and thrips (Fig. 4-22g) at higher counts but were still found to be satisfactory. Gnats were usually counted twice due to the disintegrated bodies of the gnats after getting stuck on the sticky paper traps. On the other hand, the overestimation of thrips was found to be caused by non-insect objects such as dirt that were occasionally classified as thrips. It can also be noticed that there was overestimation of other insect pests, since there are still remaining insect pests that the classifier model has not been trained on. In general, most of the insect pests were detected and recognized with an average accuracy of 0.87 and r^2 of 0.94.

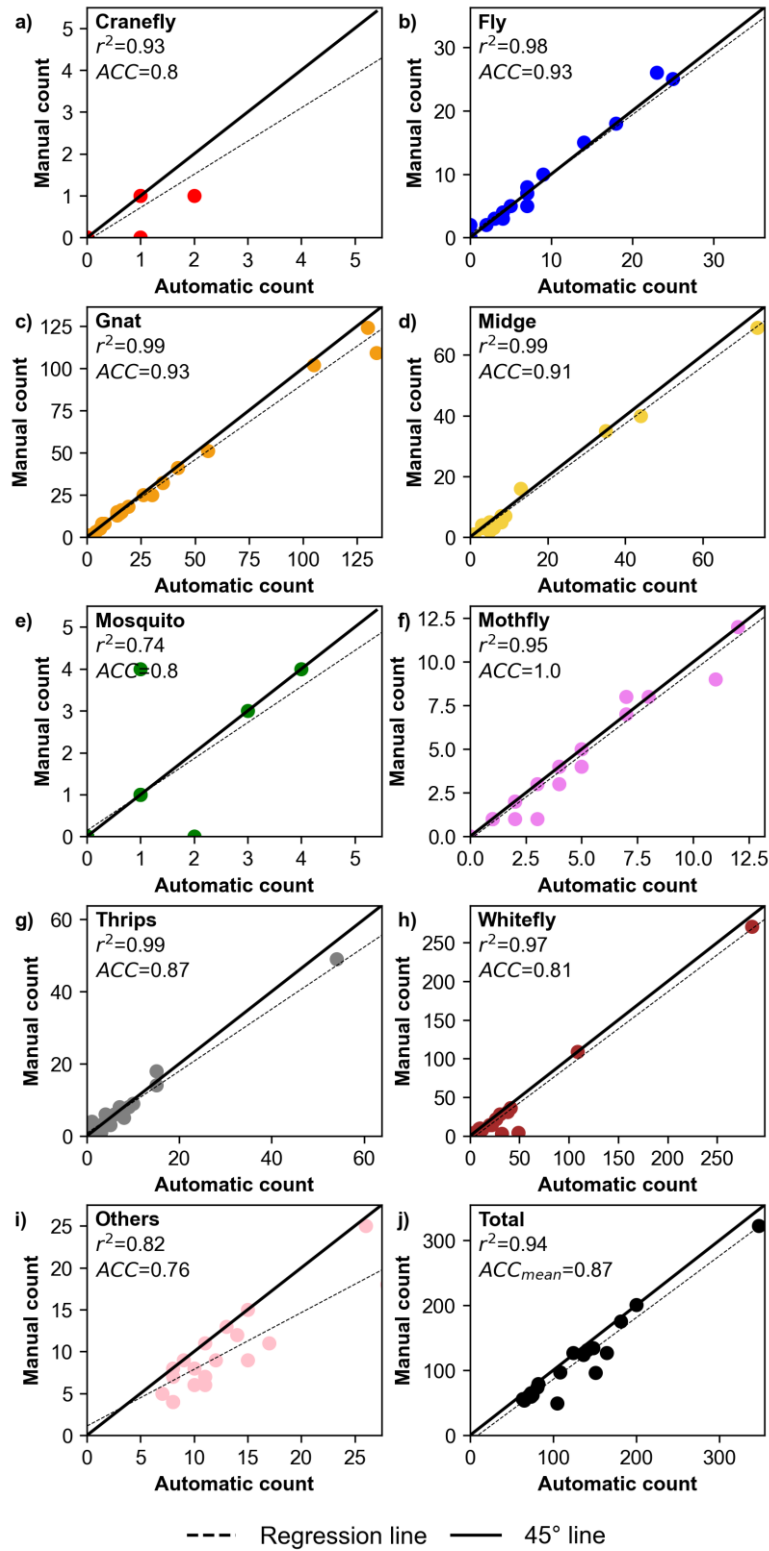


Fig. 4-22. Scatter plots of the manual and automatic insect counts of the sticky paper

images collected from the indoor installation sites, from the recent year.

The results of image level testing of the algorithm on indoor insects are also presented in boxplots in Fig. 4-23. As expected, it can be seen that the algorithm had high F_1 -scores in terms of detecting and recognizing gnats and whiteflies. It can be noticed that it performed satisfactorily in counting the rest of the classes. In particular, it only had an F_1 -score of 0.84 in counting craneflies. However, this was still acceptable since there were very few craneflies in the testing images. A relatively low F_1 -score was obtained in detecting and recognizing thrips due to the similar issue of having non-insect objects in the images. In general, the algorithm still performed well with an average F_1 -score of 0.89 based on the testing images of testing set 1. The results prove that the algorithm was reliable in both object level and image level.

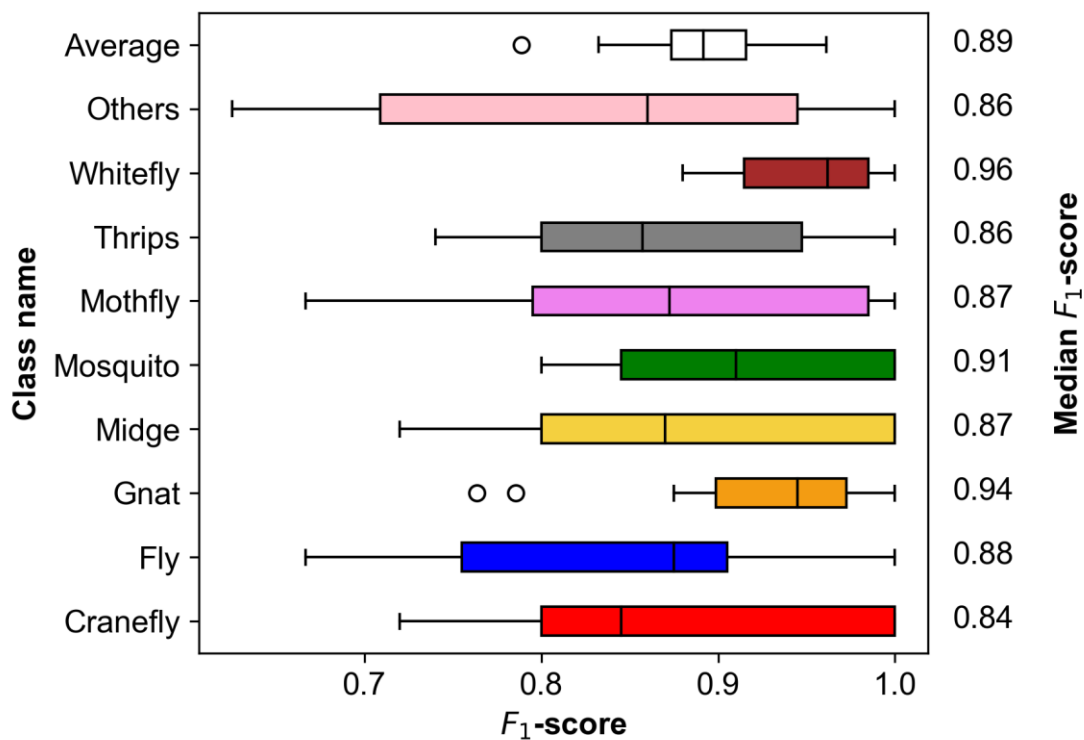
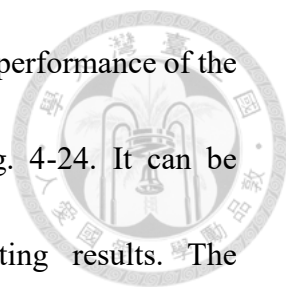


Fig. 4-23. Algorithm image level F_1 -score boxplots per class as tested on testing set 1.



The spatio-temporal voting method was applied to improve the performance of the algorithm. The re-validated scatter plot results are shown in Fig. 4-24. It can be immediately seen that there was improvement in the counting results. The overestimation issues encountered a while ago was almost solved completely through the proposed post-processing method. Interestingly, accuracies as high as 0.97 were achieved for the fly and gnat class, and 0.96 for the whitefly class. The counting of whiteflies was improved since the misclassification issue due to water droplets was solved. In other cases, the spatio-temporal voting method was also helpful in resolving the issue of glares that leads to missed detection of whiteflies. Through the retention criteria of the proposed method, the whiteflies are still detected even they appear as glares. The accuracy in counting thrips also improved from 0.87 to 0.89 (Fig. 24g). this was because the dirt objects classified into thrips were reclassified in every sticky paper trap image. By continuously checking the objects' class, it was eliminated since it will be classified into non-insect. The misclassification of other insect pests was also resolved. As a whole, the average increased to 0.91, with an r^2 of 0.99.

Finally, the image-level F_1 -score boxplots after applying the spatio-temporal voting method on the images of testing set 1 are shown in Fig. 4-25. The difference between the results in Fig. 4-23 and Fig. 4-25 can easily be noticed as the boxes drew closer to F_1 -scores of 1.0. Similarly, the algorithm still performed well in detecting and

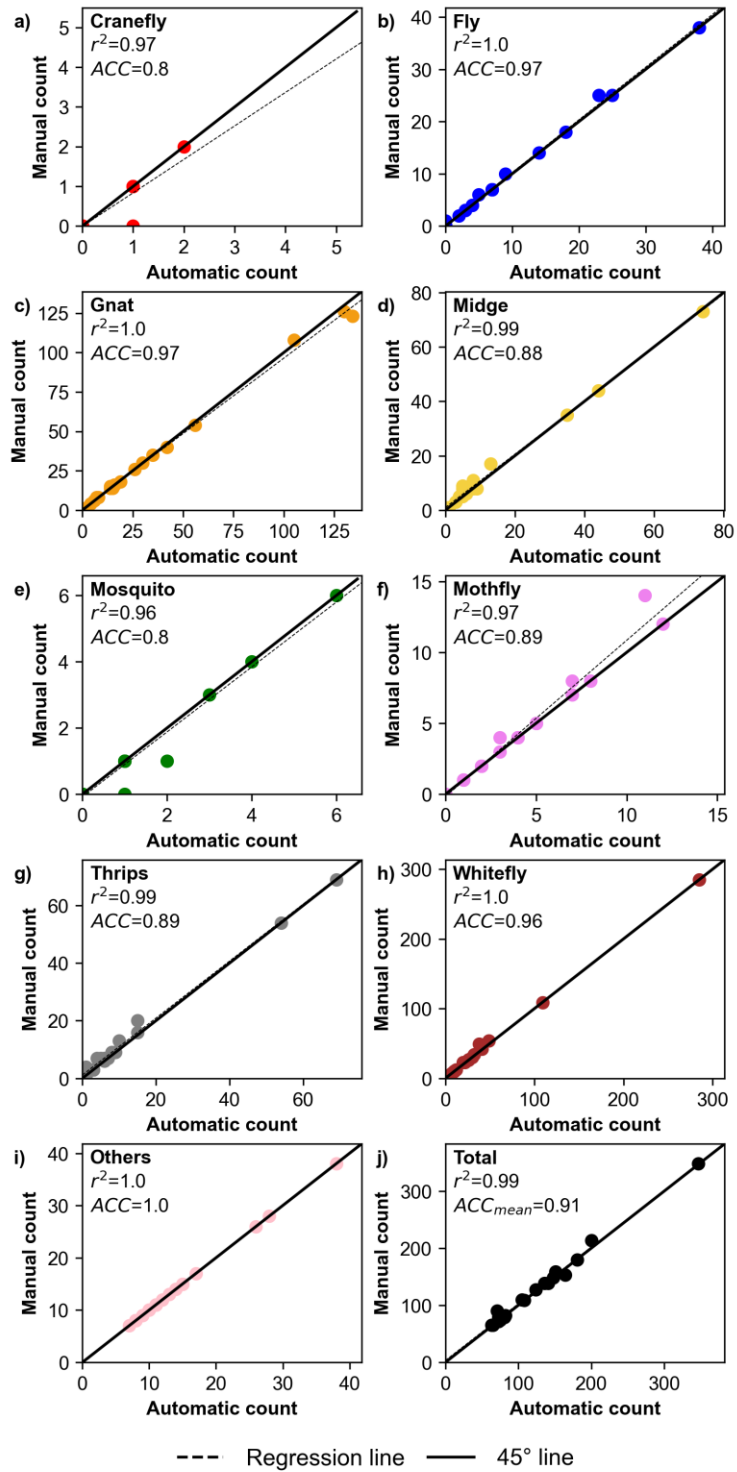


Fig. 4-24. Scatter plots of the manual and automatic insect counts of the sticky paper images collected from the indoor installation sites, from the recent year, after applying the spatio-temporal voting method

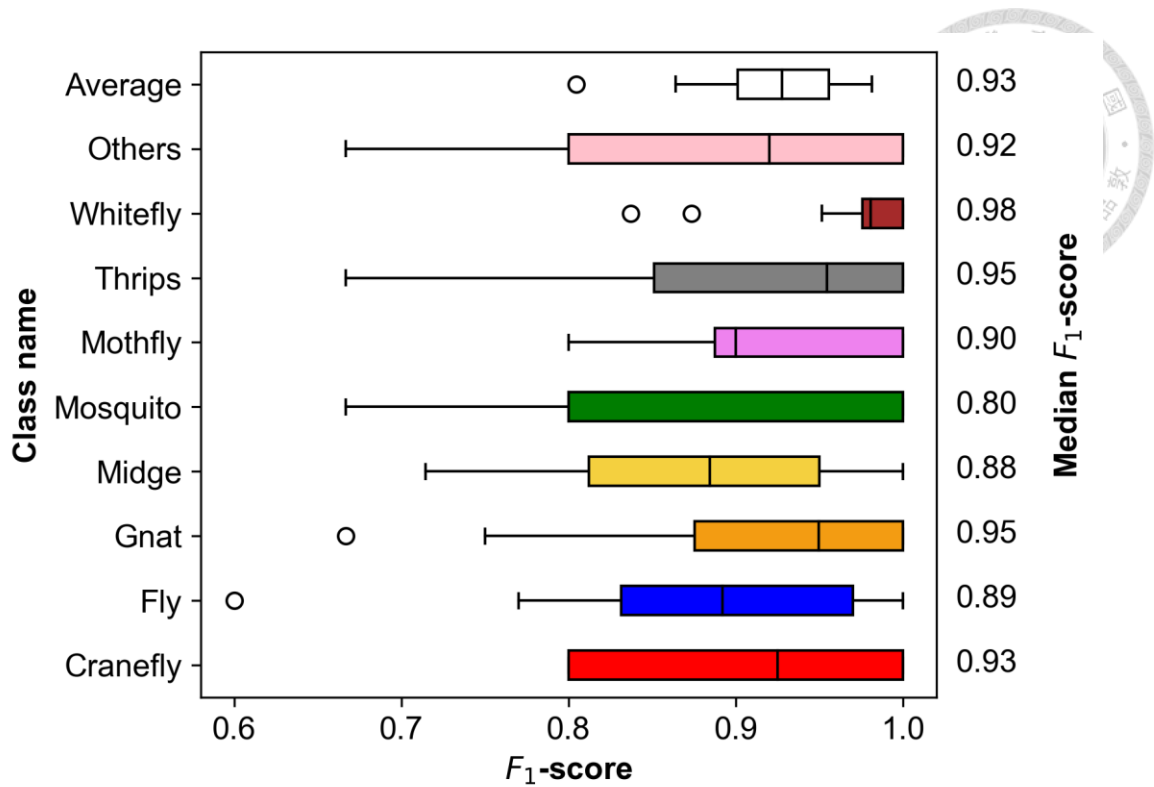
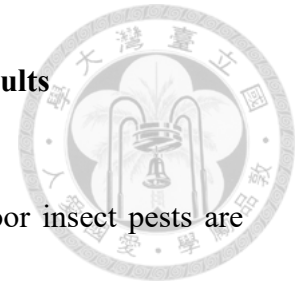


Fig. 4-25. Algorithm image level F_1 -score boxplots per class as tested on testing set 1, after applying the spatio-temporal voting method.

recognizing the gnat and whitefly classes. However, it can be seen that there was a big improvement in the thrips class as the F_1 -score increased 0.86 from to 0.95. This means that the presence of foreign objects on the sticky paper trap images did really have an effect on the algorithm results. The rest of the classes also had improved F_1 -scores, accordingly. It can be seen that the image-level average F_1 -score increased from 0.89 to 0.93. This shows that the spatio-temporal voting method was relevant to the algorithm since it can solve missed detection, overlapping detection, and misclassification issues caused by non-insect objects.

4.1.3.2 Outdoor insect pest detection and recognition testing results



Sample processed images of detecting and recognizing outdoor insect pests are shown in Fig. 4-26. The first sample image (Fig. 4-26a) shows that most of the insect pests were detected and classified correctly, most especially small insects such as leafhoppers, gnats, and thrips. It also shows two common errors in detection and classification. In the middle of the image, a missed detection error was encountered due to touching bodies of insect pests. This was mostly due to the large bodies of fruitflies. This issue can be avoided in the future if more object detector samples were obtained. The bottom of the image shows a misclassification of a fruitfly into gnat. This was a common error since some gnats do appear like fruitflies. The second sample image (Fig. 4-26b) shows examples of classifying oriental fruitflies, mothflies, and mango leafhoppers. The first error in the sample processed image shows a sample of duplicate detection. This was occasionally encountered since the anchor box of the object detector sees the object twice. The error found in the bottom of the sample processed image shows a classification error of dirt into thrips, which was also commonly encountered even in classifying indoor insects. The sample processed images show that the developed algorithm successfully detected and recognized the target insect pests with minor issues that may be solved by re-training the models in the future.

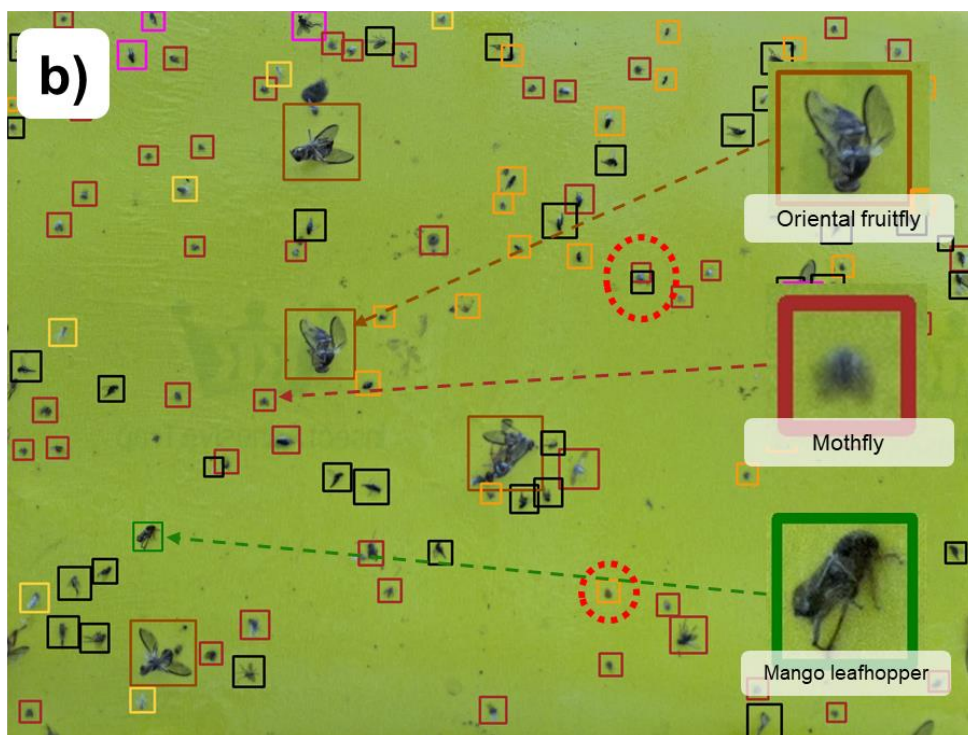
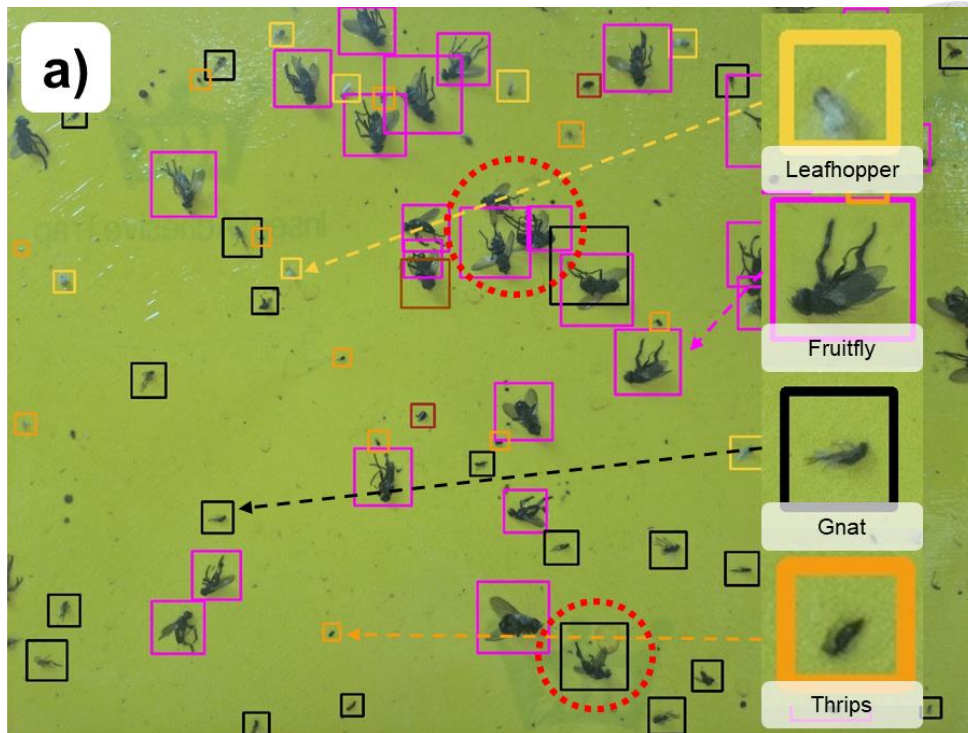
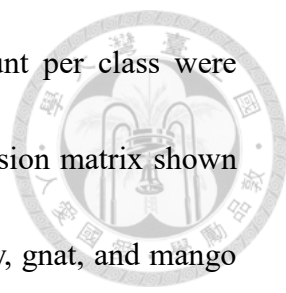


Fig. 4-26. Outdoor insect pest detection and recognition sample processed images: a) Image with leafhopper, fruitfly, gnat, and thrips; b) Image with oriental fruitfly, mothfly and mango leafhopper. Sample errors were marked with dotted circles.



The scatter plots of the manual count vs. the automatic count per class were obtained, as shown in Fig. 4.27. It shows that, similar to the confusion matrix shown previously in Fig. 4-27, there was misclassification between fruitfly, gnat, and mango leafhopper due to their close appearances. This led to overestimation of fruitflies and underestimation of gnats and mango leafhoppers. However, it shows that the wrong estimation between the three classes were not too large as the accuracies of fruitfly, gnat and mango leafhopper were 0.84, 0.90, and 0.88, respectively. It also shows that the accuracy of counting oriental fruitflies was quite low. This was also due to some wrong classification of some oriental fruitflies into fruitflies. It can also be noticed that the number of oriental fruitflies was quite few, which also made the accuracy and r^2 sensitive to the small difference in the numbers. The counting of thrips had an r^2 of 0.86 and accuracy of 0.87, which was acceptable considering the cleanliness of the sticky paper traps. It was also observed that there were still many unidentified insects that were not classified into any insect class. Upon inspection, the other insects include other insect types such as large mothflies, worms, and more. Unfortunately, there were very few samples of the other insect types for training and were not too important to the farm managers. In conclusion, most insect pests were counted with an r^2 of 0.98 and average accuracy of 0.88, including only the 7 classes. Meanwhile, the image-level testing F_1 -score boxplots are shown in Fig. 4-28.

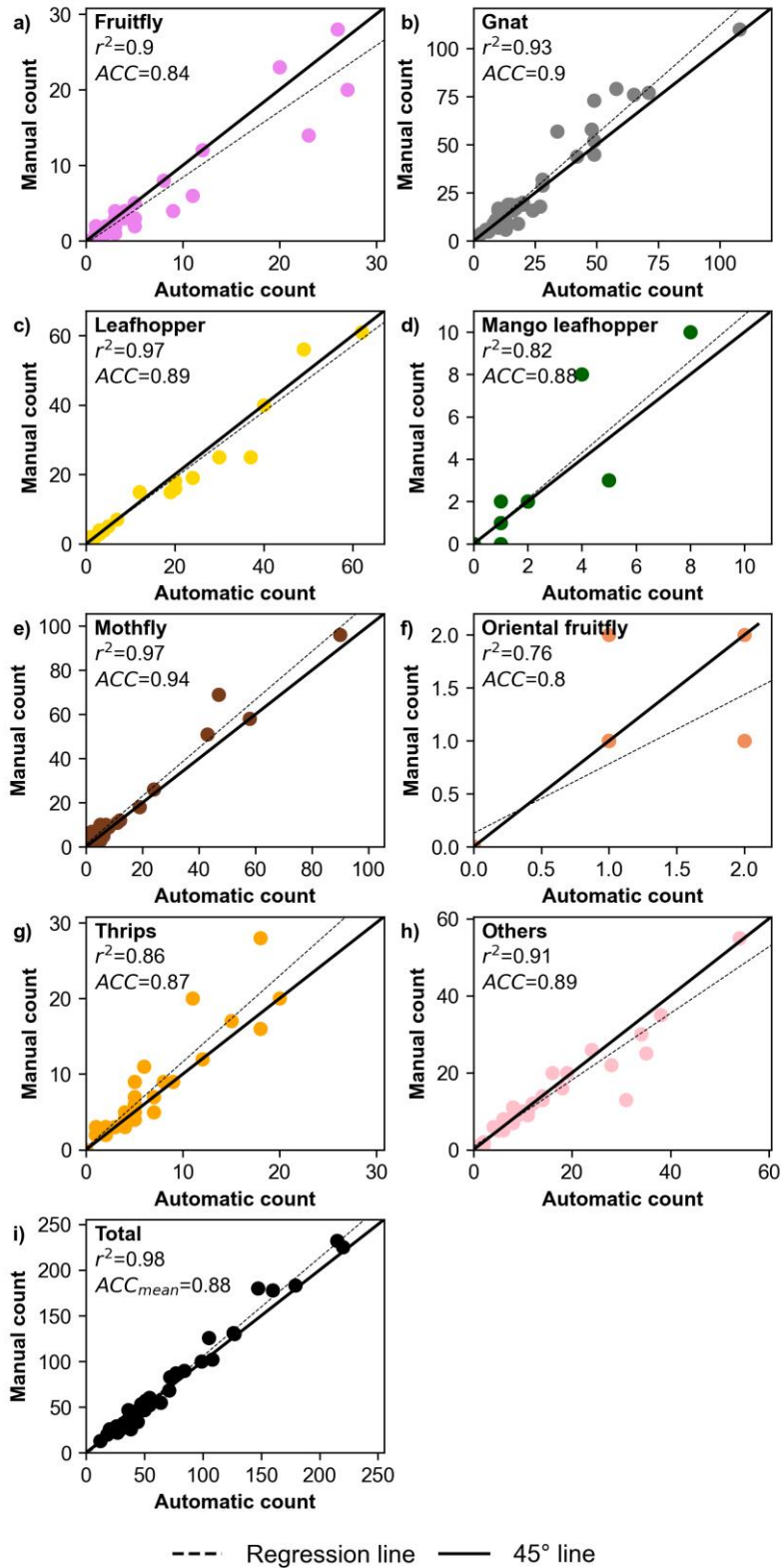


Fig. 4-27. Scatter plots of the manual and automatic insect counts from the sticky paper images collected in Farm M1, from the recent 6 months.

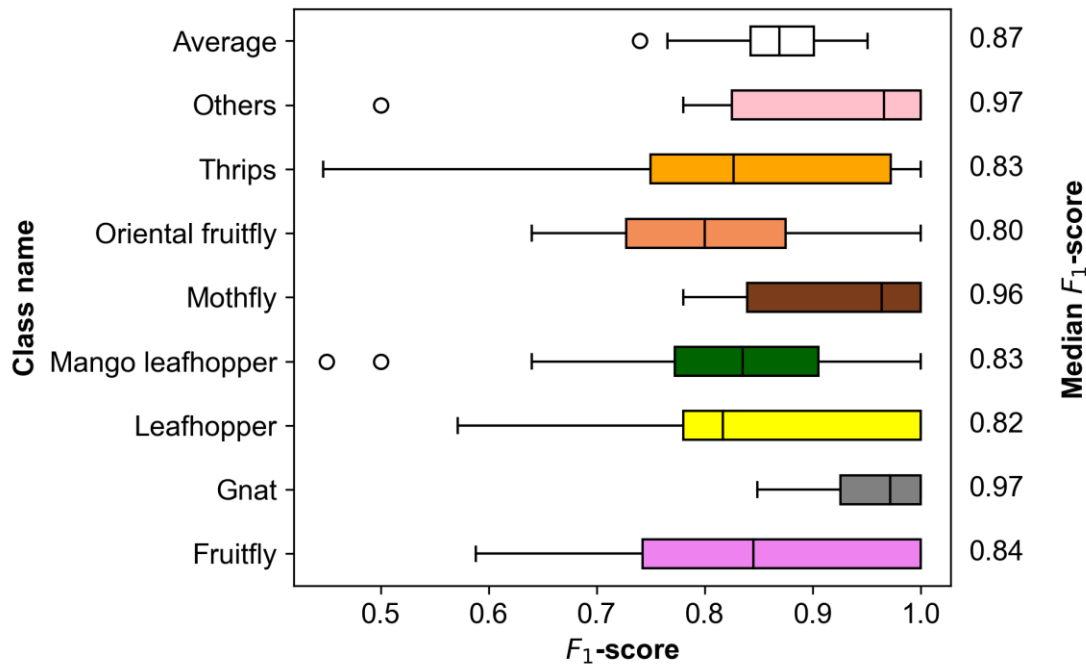


Fig. 4-28. Algorithm image level F_1 -score boxplots per class as tested on testing set 2.

The image level testing results shown in Fig. 4-28 show that most of the insect classes were correctly classified with median F_1 -scores of 0.8 and above. The only issue observed was the detection and classification of oriental fruitflies. However, this was also caused by the few oriental fruitflies detected in each sticky paper trap image. There were also few cases of misclassified oriental fruitflies. The algorithm also did not perform that well in detecting and classifying leafhoppers, mango leafhoppers, and thrips. This issue was found to be caused by non-insect objects that were misclassified into the three classes. In average, a median F_1 -score of 0.87 was obtained. To improve the performance of the algorithm, the spatio-temporal voting method was applied. The new scatter plots of the manual and automatic counts are shown in Fig. 4-29.

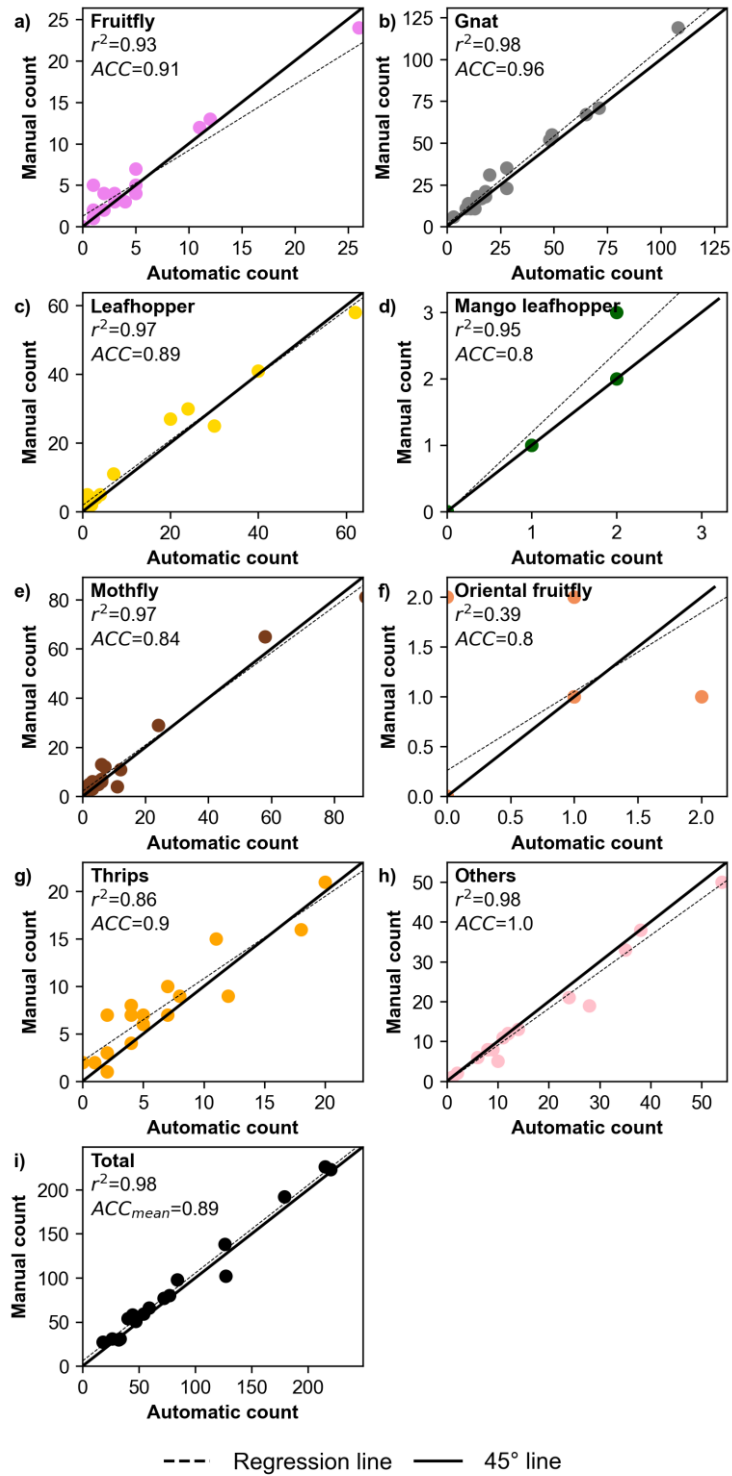


Fig. 4-29. Scatter plots of the manual and automatic insect counts from the sticky paper images collected in Farm M1, from the recent 6 months, after applying the spatio-temporal voting method.

Generally, it can be seen that there was an improvement in the r^2 values and accuracies of the algorithm on each class. The best improvement was the detection and recognition of fruitfly from an accuracy of 0.84 to 0.91. This was because wrong classifications of fruitfly was reduced through the spatio-temporal voting method. The image-level F_1 -score boxplots also showed improvements to the algorithm, as shown in Fig. 4-30. There were improvements of gnat from 0.90 to 0.96, thrips from 0.87 to 0.9, and others from 0.89 to 1.0. There were drops in F_1 -score of the other classes but were almost negligible. It also slightly improved the average accuracy of the algorithm from 0.88 to 0.89. The results show that the timely information from the sticky paper trap images was valuable in improving the latest after applying the spatio-temporal

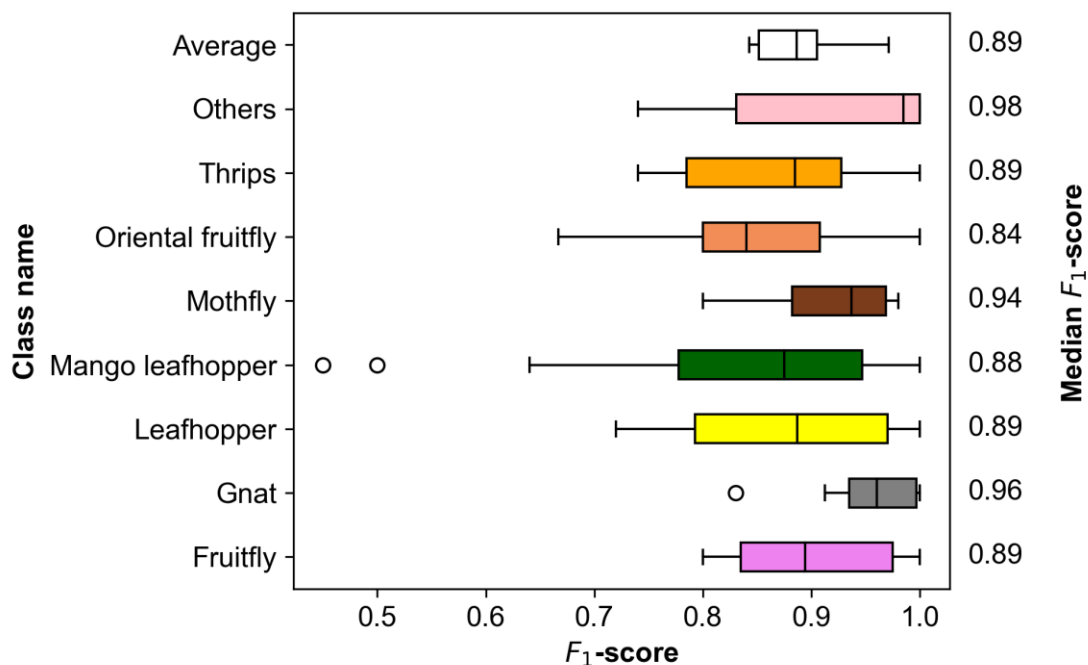
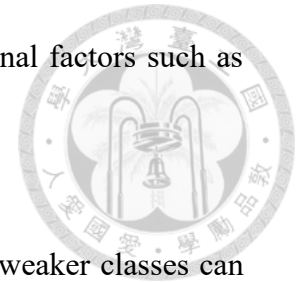


Fig. 4-30. Algorithm image level F_1 -score boxplots per class as tested on testing set 2.

voting method. It was able to reduce the potential effects of external factors such as lighting condition, presence of non-insect objects, and more.



It can be concluded from the results that classification of the weaker classes can still be improved by retraining the classifier models with more training image samples.

It also showed that the multi-stage approach in classification was effective in classifying the insect pests even up to the species level such as mango leafhopper and oriental fruitfly. However, this also means that classifying into the species level will require adequate training samples to be more effective and accurate. The unknown insect pest types should also be classified in the future as soon as more training samples were collected. This may also depend on the target insect pests of the farm managers. In the case of the mango farm, the most important insect pests to monitor were thrips, mango leafhopper, and oriental fruitfly. In the current state of the algorithm, it was able to satisfactorily detect and recognize the mentioned insect pests.

The semi-supervised learning method, which was applied on the indoor multi-class insect classifier model, may also be used to improve the classifier model over time. However, it should be optimized for fine-tuning more than one model in every semi-supervised learning cycle to make it more flexible.



4.2 System performance testing

The performance of the system was measured in terms of its data throughput and trapping efficacy; indicating the reliability and validity of the data collected by the system and its accuracy. The following sub-sections discuss about the results of the tests and analyses performed.

4.2.1 System data throughput analysis

The data throughput performance of each location was analyzed. The data throughput analyses of the longest running installation sites: Farms TS1, TS2, and O1 are shown in Fig. 4-31, while Table 4-2 shows the summary of all sites.

The results illustrated in Fig. 4-31 show that the system was able to generally obtain more than 80% of the expected data. The first installation site, Farm TS1, had a good data throughput of 93.23%; indicating that there were few sensor node breakdowns. On the other hand, Farms TS2 and O1 had lower data throughput. The different between the data throughput on each farm was found to be caused by numerous and site-specific issues as shown in the summary in Table 4-2.

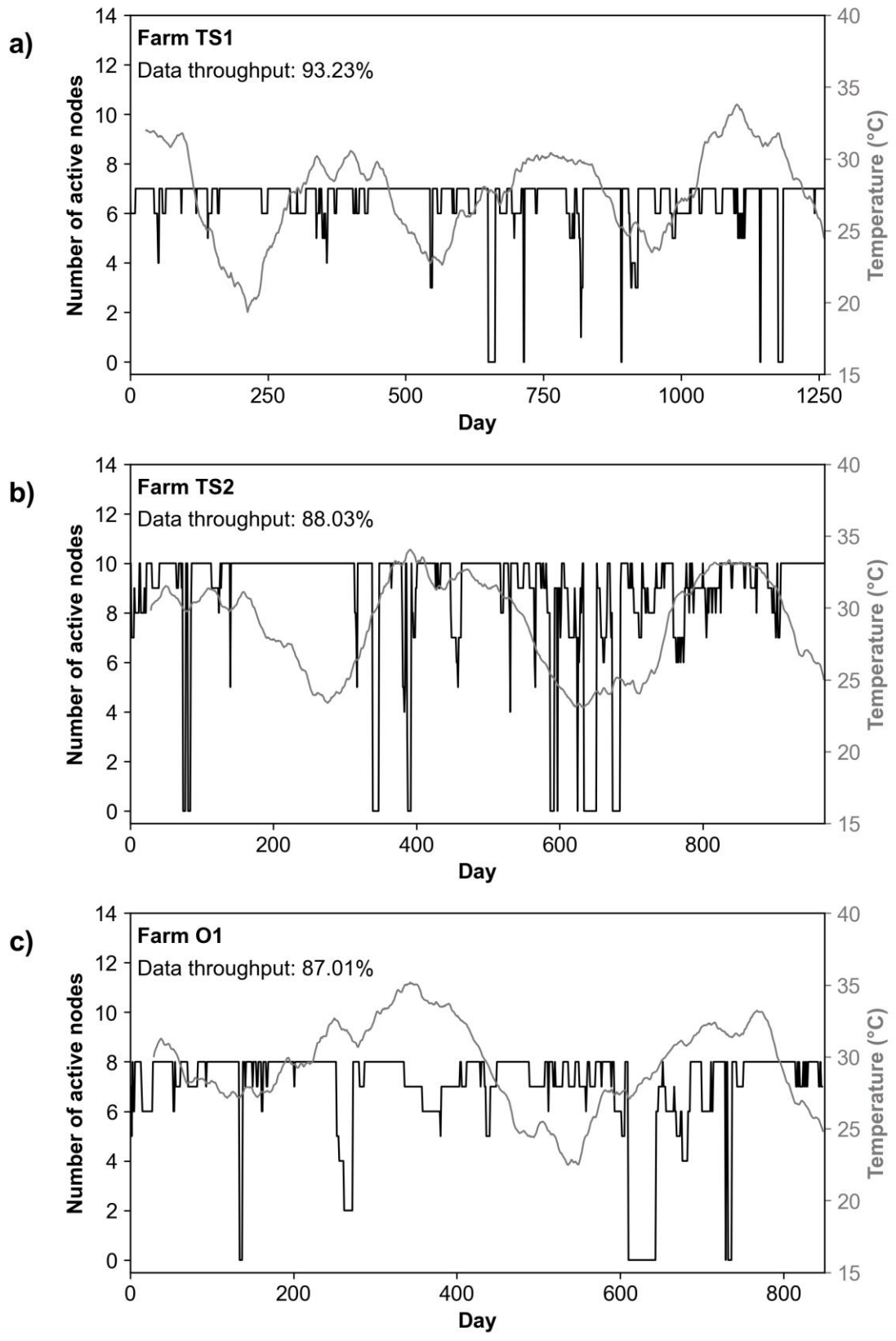


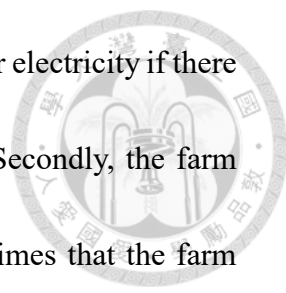
Fig. 4-31. System data throughput analyses of Farms a) TS1, b) TS2, and c) O1.

Table 4-2. Data throughput analysis of the sensor nodes of each installation site.

Farm name	Data			Data throughput percentage	Data loss percentage	Common issue/s
	Received	Expected	Loss			
TS1	8216	8813	597	93.23	6.77	a
TS2	9690	8530	1160	88.03	11.97	a
TS3	1825	2334	1825	78.19	21.81	a, b
T1	1869	2136	267	87.50	12.50	c
T2	1725	2136	411	80.76	19.24	d
O1	5903	6784	881	87.01	12.99	a
O2	3312	4038	726	82.02	17.98	b
O3	3304	4710	1406	70.15	29.85	a
S1	2897	5224	2327	55.46	44.54	a, b
C1	1591	1740	149	91.43	8.57	e
M1	3516	5080	1564	69.21	30.79	a, b, c, d

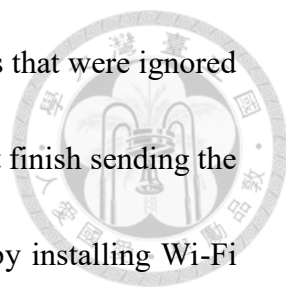
where a = farm management, b = sensor node hardware, c = Wi-Fi issue, d = Internet issue, e = external factors

First of all, some of the farm owners will cut off their electricity for a few days. According to them, there were times they had to clean and disinfect the farm. In so doing, they would have to greatly increase the temperature in their farm. To protect the sensor nodes, turning off the power would be safer. Instances like this was encountered around days 680 to 690 of Farm TS1 (Fig. 4-31a) and days 620 to 630 of Farm TS2



(Fig. 4-31b). On the other hand, Farm O1 will sometimes cut off their electricity if there were no crops in the farm such as in days 600-630 (Fig. 4-31c). Secondly, the farm managers did not know how to maintain the system. There were times that the farm managers were not aware that the sensor nodes were off or broken. This was mostly encountered in Farms TS1, TS2, TS3, O3, S1, and M1. Despite our persistence in notifying the farm managers with regards the issue, they were often not unavailable to address the problem. Unfortunately, these problems are in the hands of the farm managers. The results also show that the data throughput of the system partially depended on the management of the farm owners.

From the system side, the primary cause of breakdowns was due to hardware problems. The issues include sensor node overheating, stuck up operating system, weak Wi-Fi signal, sensor node waterproof issues, and power supply issue. Overheating of the sensor node leads to CPU throttling and stuck up operating system of the sensor nodes. The overheating problem was noticeable in all farms whenever the average temperature in the farm was higher than 30°C. During the initial stages of the system, this was solved by including an on-board fan connected to the Raspberry Pi. However, this leaves the sensor node vulnerable to water splashes that caused permanent damage to the sensor nodes. Thus, making the sensor node waterproof was made a higher priority. The weak Wi-Fi signal also contributed to the number of inactive nodes since



the farms had various area sizes. This caused corrupted data packets that were ignored in the server side. There were also times that the sensor node cannot finish sending the sticky paper trap images. Depending on the site, this was solved by installing Wi-Fi signal extenders. Waterproofing was also found as an important issue and was encountered usually in Farms TS3, O2, S1 and M1. The indoor farms TS3, O2, and S1 had different irrigation methods compared to the other installation sites. The farms had denser water sprinklers that causes water seepage going to the inside of the device. To solve this problem, the latest design of the sensor node had all holes sealed with acrylic sealants. Meanwhile, Farm M1 had a harsher environment and occasionally experiences continuous rainfall. The same solution was applied to the sensor nodes of Farm M1.

An external factor that caused device breakdowns was internet signal. In Farms T2 and M1, the internet signal was not stable since the telecommunications company informed us that there were few cell sites around the farms. Amplification of the signal had to be requested to make sure the installation sites had stable internet signal. Moreover, the internet promo of the sim cards changed from time to time causing data limits. This shows that the stability of the system depended on the reliability of the internet service. Edge computing is a possible solution to this issue. For an instance, the sticky paper trap images can be processed on the device and not in the server. This will shrink the size of the data that the sensor nodes have to send to the server.

4.2.2 System insect pest trapping efficacy analysis



The results of the system insect pest trapping efficacy experiment was analyzed.

Each insect pest count data set consists of 18 data points obtained every 2 weeks.

The analysis results per node of Farm TS1 are shown in Fig. 4-32 and Fig. 4-33, for whiteflies and thrips, respectively. It can be seen from the results that the raw automatic insect pest counts IC_{auto} were quite different to the manual insect pest counts of the sticky paper trap below the sensor node IC_{below} and the manual count insect pest count of the sticky paper trap held by the sensor node IC_{device} . The difference between IC_{auto} and IC_{below} was caused by three factors: shape of the sticky paper trap, vertical distance of the crops and the sensor node, and design of the sensor node. It was mentioned previously in the text that the shape of the sticky paper trap had a significant effect to the number of insects trapped and was verified through the results presented. The effect of the height distance was also proven in other related studies (Atakan & Canhilal, 2004; Kaas, 2005). Atakan and Canhilal (2004) found that every vertical distance of 20 cm from the crops to the sticky paper trap caused at least 12-15% decrease in the number of whiteflies trapped. The sensor node also may occasionally block the flight pathway of the insects. Considering these factors, it was reasonable that there was at least three-fold difference between the insect pest counts.

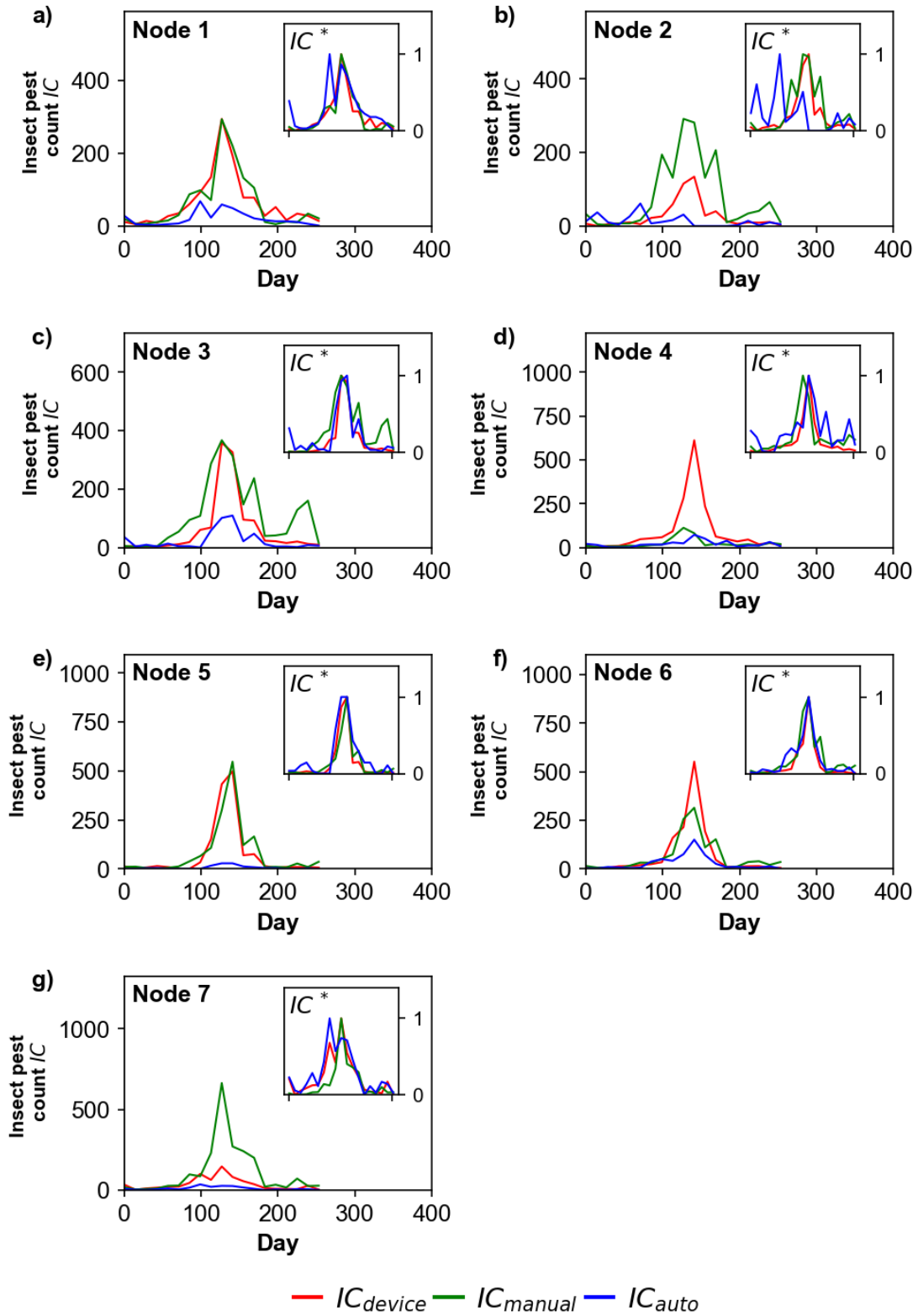


Fig. 4-32. Whitefly trapping efficacy experiment results obtained from each sensor

node of Farm TS1.

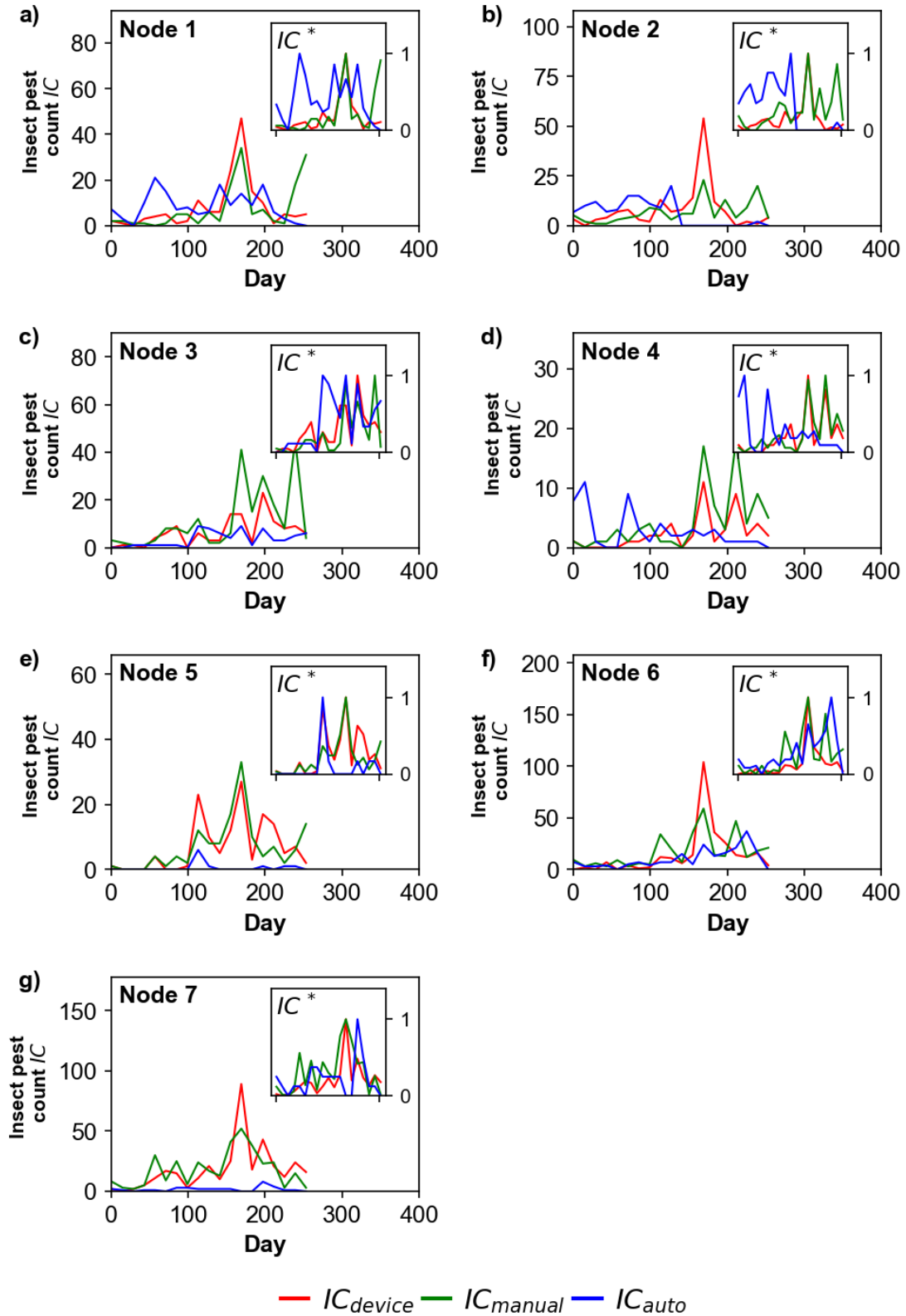
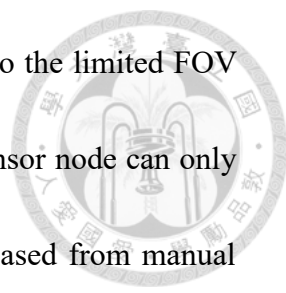


Fig. 4-33. Thrips trapping efficacy experiment results obtained from each sensor node

of Farm TS1.



Meanwhile, the difference IC_{auto} and IC_{device} can be attributed to the limited FOV of the sensor node's camera. As mentioned in Section 3.2.1, the sensor node can only see 11.5 cm x 15 cm of the 14.8 cm x 21 cm sticky paper trap. Based from manual inspection of the sticky paper traps, there was quite a number of insects stuck on the edges of the sticky paper trap. This issue can be solved in the future if there was a compatible camera that has a better resolution than the current one.

It can be noticed that the number of thrips automatically counted by the system were different to the number of manually counted thrips. This was mainly attributed to the misclassification of dirt to thrips. However, it can also be seen than the number of thrips in Farm TS1 were few; therefore, the results were still acceptable.

Despite the raw value differences between the insect pest counts, the normalized values showed that the values had generally similar trends. As the total insect pest counts were computed, the results were found more reasonable, as shown in Fig. 4-34 and Fig. 4-35. It can be seen from Fig. 4-34 that the system was able to reliably count whiteflies with an r^2 of 0.87. Paired t-test results also show that the null hypothesis cannot be rejected ($p\text{-value} = > \alpha$, $\alpha = 0.05$, $df = 18$). However, as expected, the r^2 of counting thrips was quite low and the paired t-test results show significant difference. The results show that the insect pest detection and recognition algorithm can still be improved in the future for classifying thrips and the cleanliness of the greenhouse still

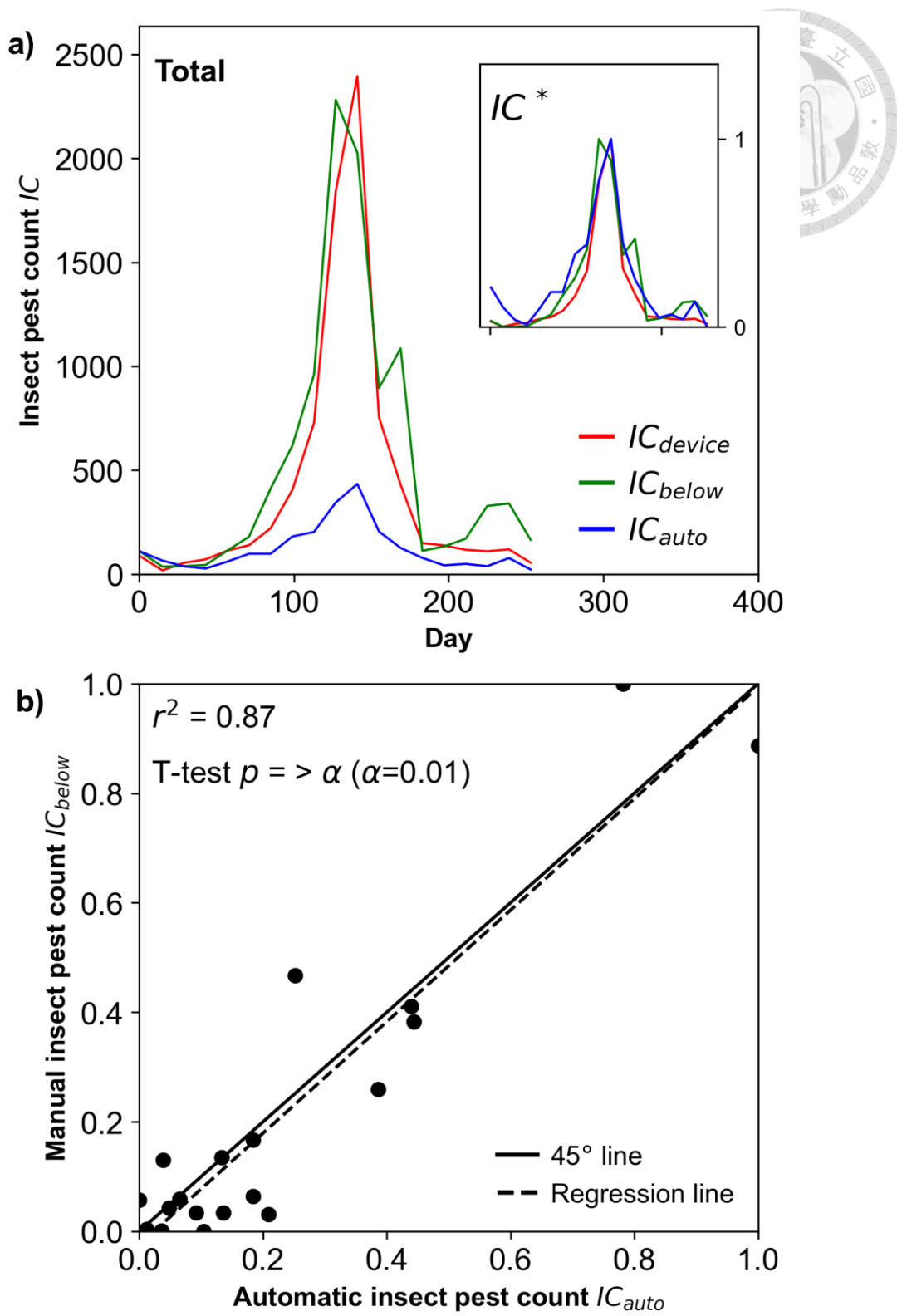


Fig. 4-34. Total whitefly trapping efficacy experiment results obtained from Farm

TS1. a) Raw and normalized insect pest counts; b) Linear regression analysis.

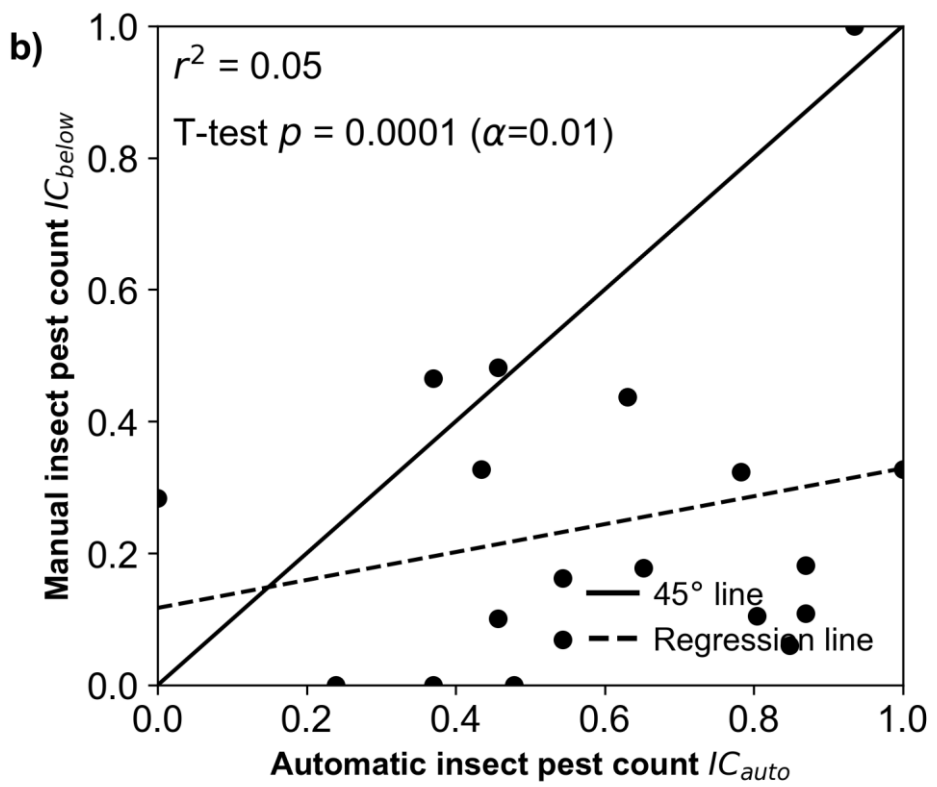
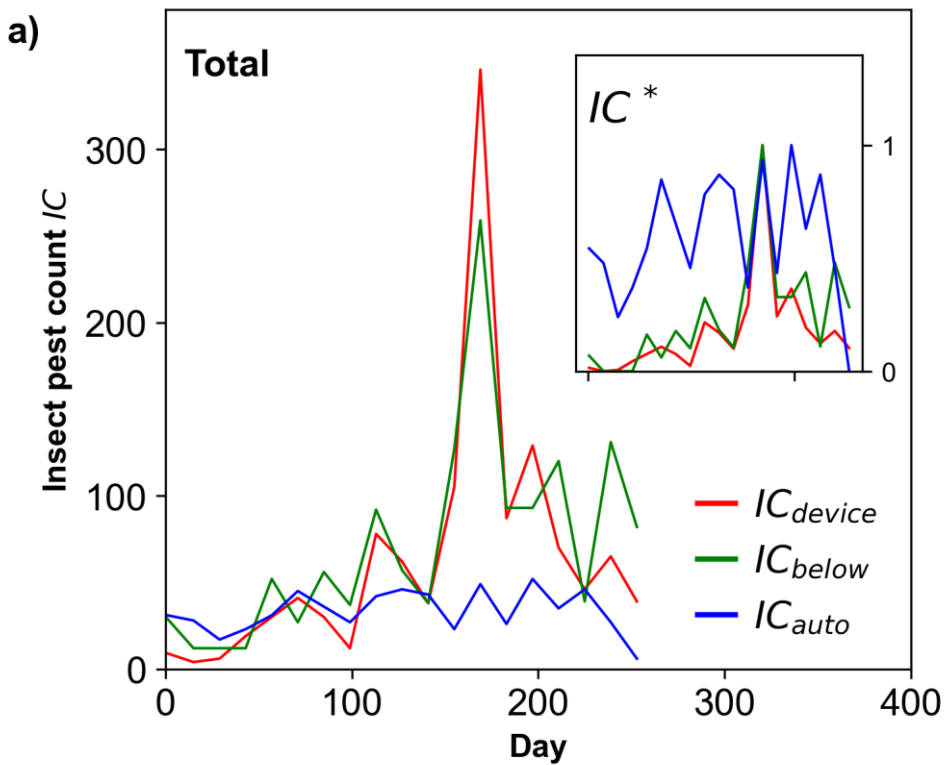


Fig. 4-35. Total thrips trapping efficacy experiment results obtained from Farm TS1.

a) Raw and normalized insect pest counts; b) Linear regression analysis.

had an effect to the automatic insect pest counting results



The trapping efficacy analysis results of Farms TS2 and TS3 are shown in Figs. 4-36 to 4-43. It can be normally seen that the trapping efficacy of whiteflies were also good for Farms TS2 and TS3. Both t-test results performed on the dataset of Farms TS2 and TS3 showed that IC_{auto} and IC_{below} were not significantly different. Compared to Farms TS1 and TS3, the efficacy of trapping and counting thrips in Farm TS2 were relatively better. Since there were a lot of thrips found in Farm TS2, it was found that the system did reasonably perform well when there were more insect pests trapped. However, there was still a remaining issue of underestimation as shown in Fig. 4-39.

Nevertheless, it can be seen from the comparison results that the insect pest counts of the system had a similar trend to the insect pest counts of the sticky paper trap below the sensor node. This means that the issues including: shape of the sticky paper trap, height distance of the sensor node from the crops, and sensor node design, can be ignored. This conclusion can also be inferred since the comparison was done during similar time periods. The difference can be further examined through a more structured test in which the sensor node was separated from the cylindrical sticky paper trap and the experiment was conducted in different time periods.

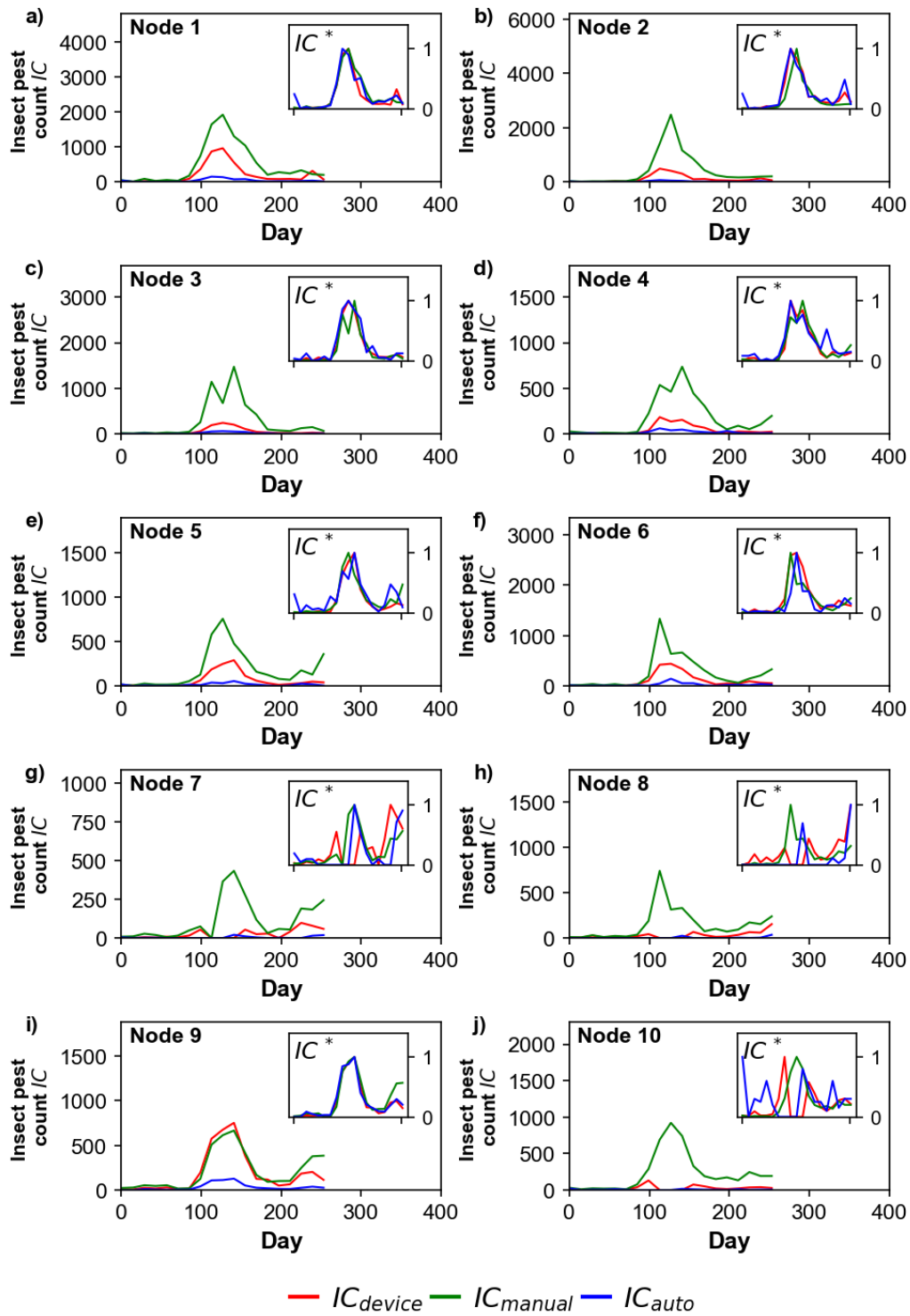


Fig. 4-36. Whitefly trapping efficacy experiment results obtained from each sensor

node of Farm TS2.

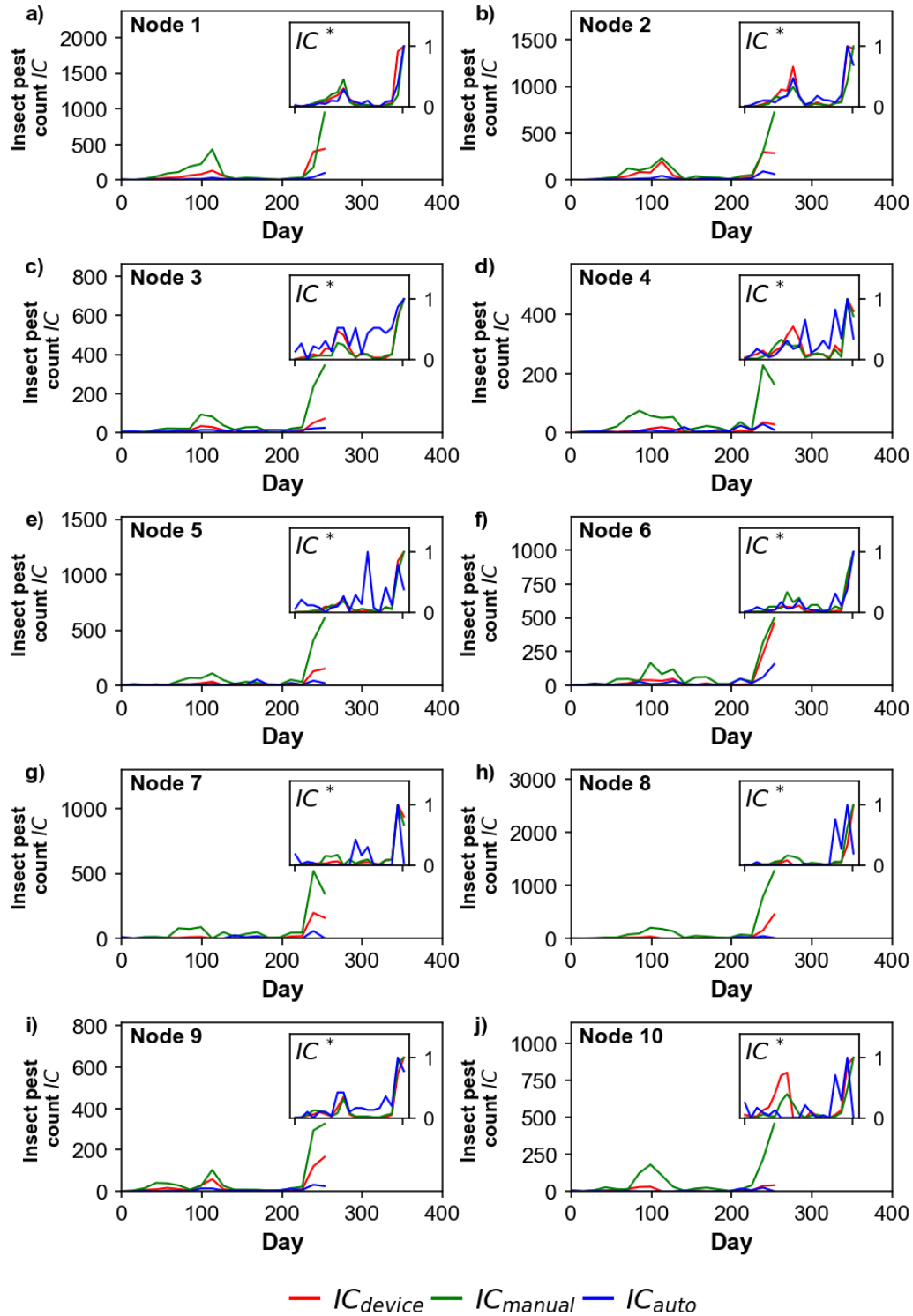


Fig. 4-37. Thrips trapping efficacy experiment results obtained from each sensor node of Farm TS2.

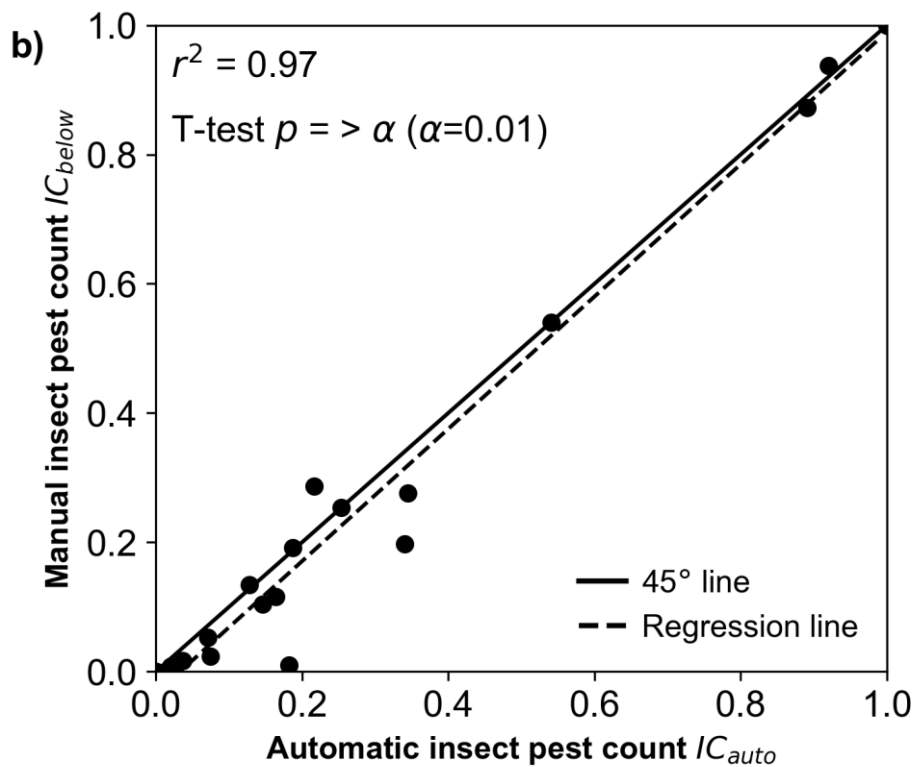
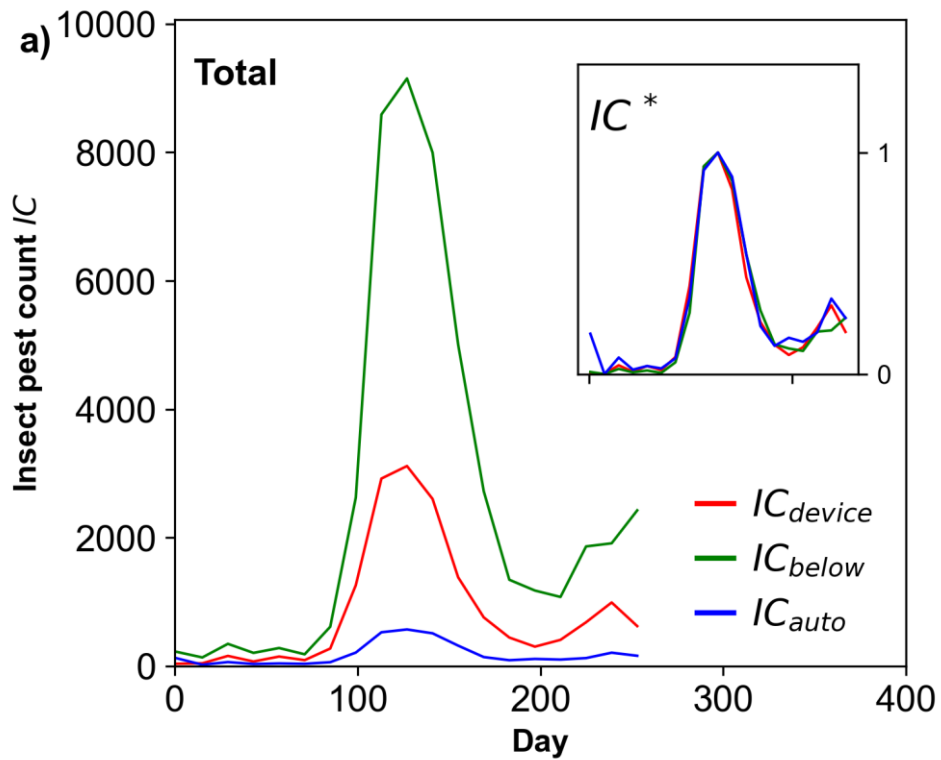


Fig. 4-38. Total whitefly trapping efficacy experiment results obtained from Farm

TS2. a) Raw and normalized insect pest counts; b) Linear regression analysis results.

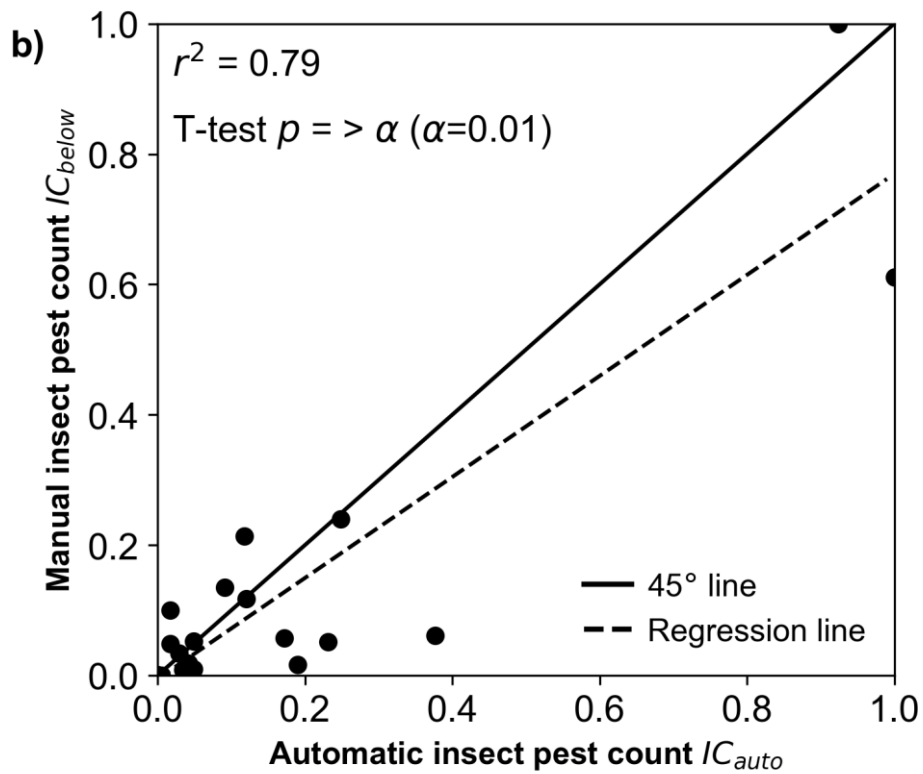
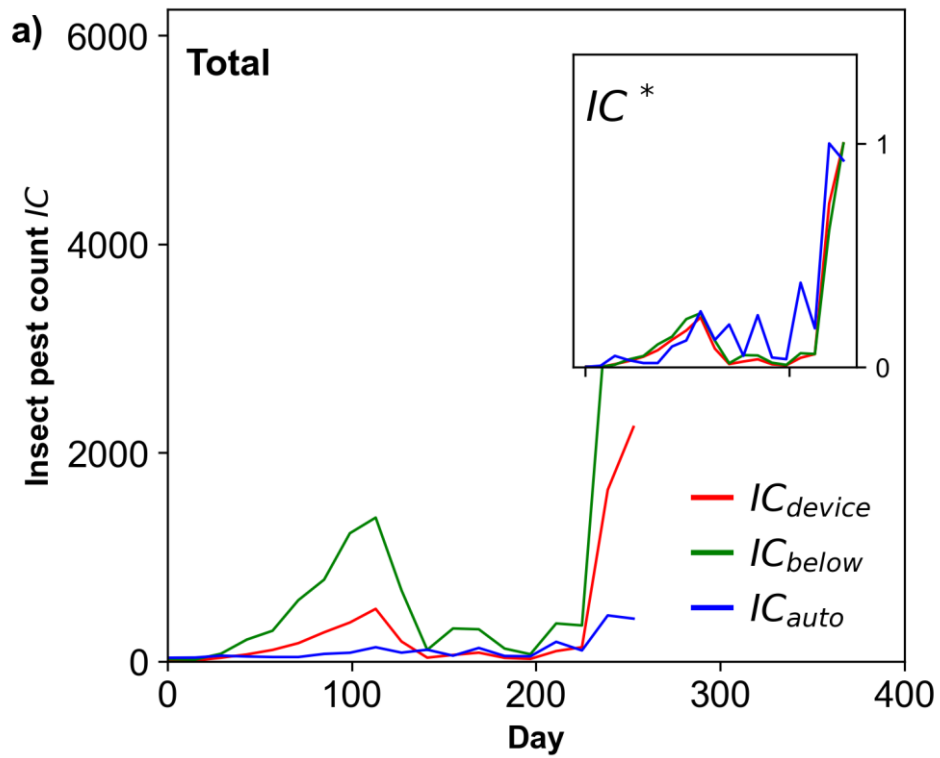


Fig. 4-39. Total thrips trapping efficacy experiment results obtained from Farm TS2.

a) Raw and normalized insect pest counts; b) Linear regression analysis results.

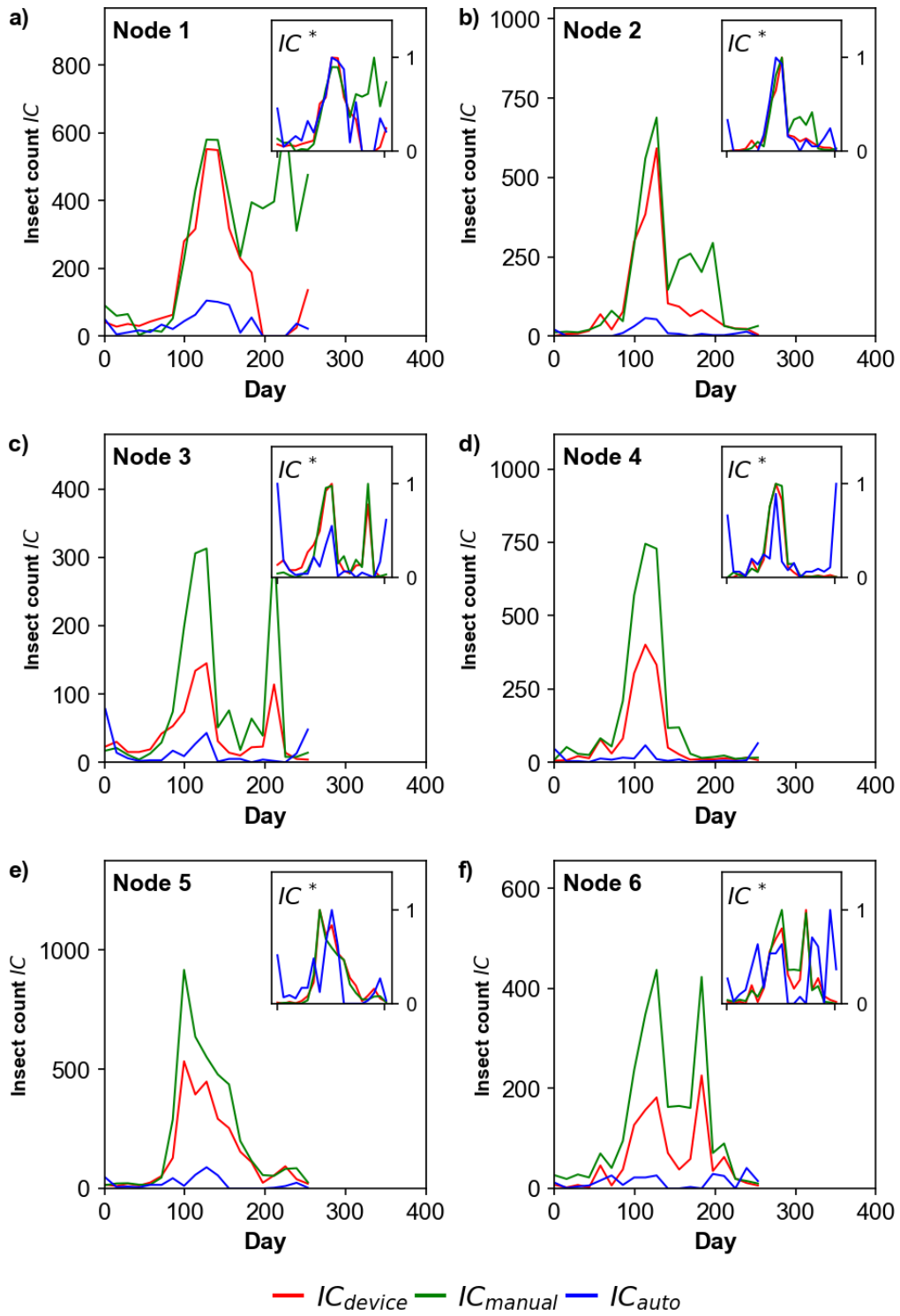


Fig. 4-40. Whitefly trapping efficacy experiment results obtained from each sensor

node of Farm TS3.

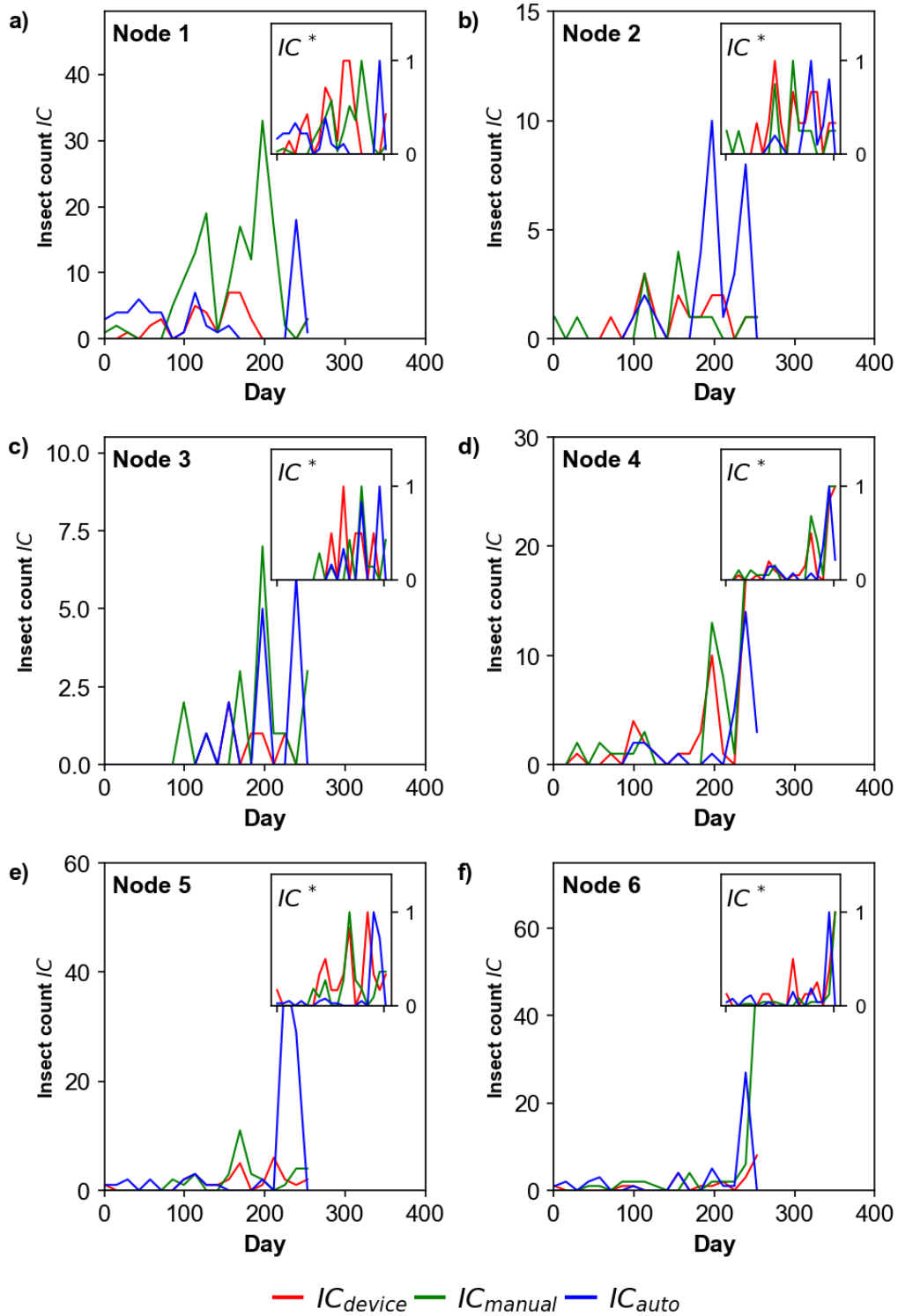


Fig. 4-41. Thrips trapping efficacy experiment results obtained from each sensor node of Farm TS3.

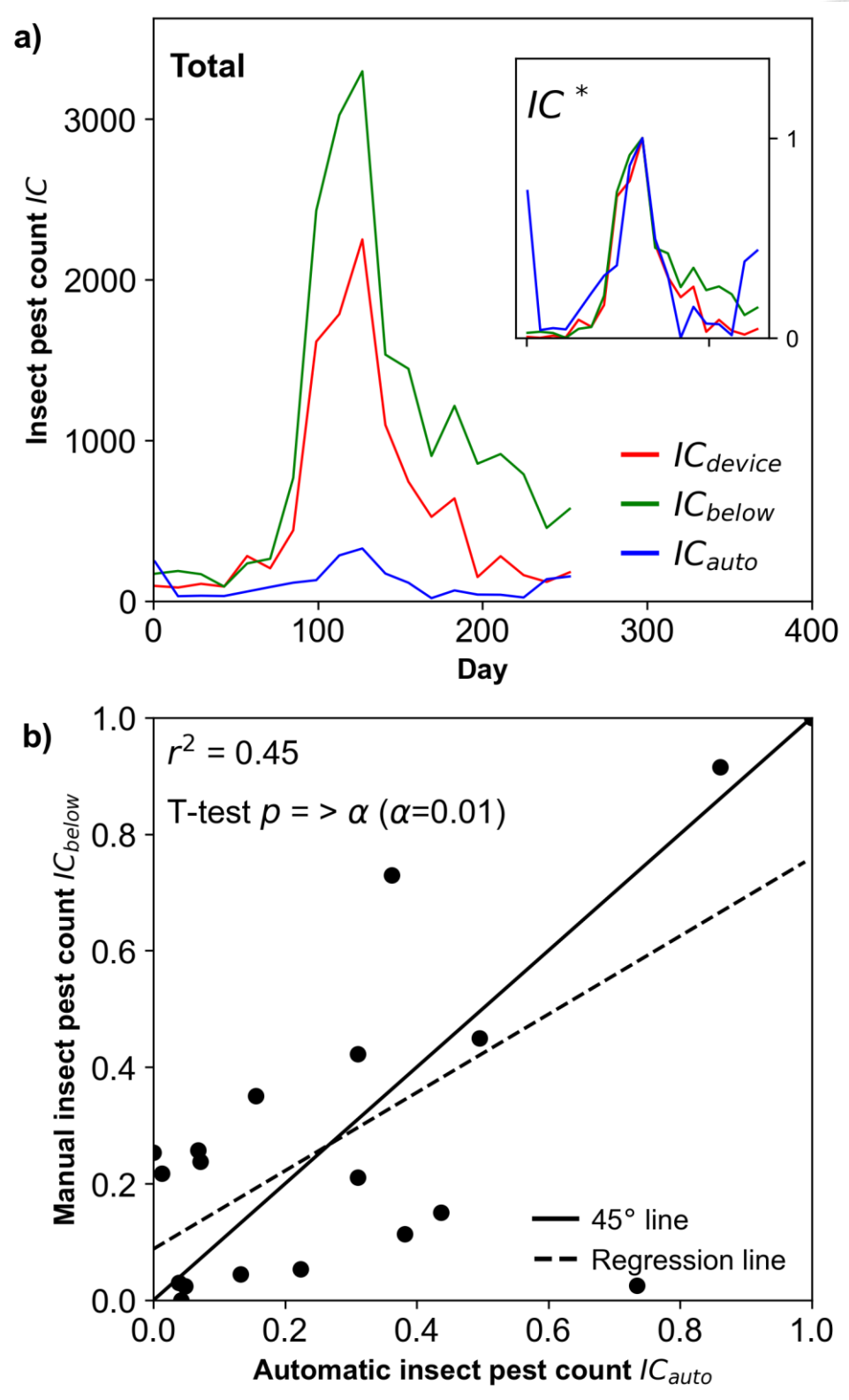


Fig. 4-42. Total whitefly trapping efficacy experiment results obtained from Farm

TS3. a) Raw and normalized insect pest counts; b) Linear regression analysis results.

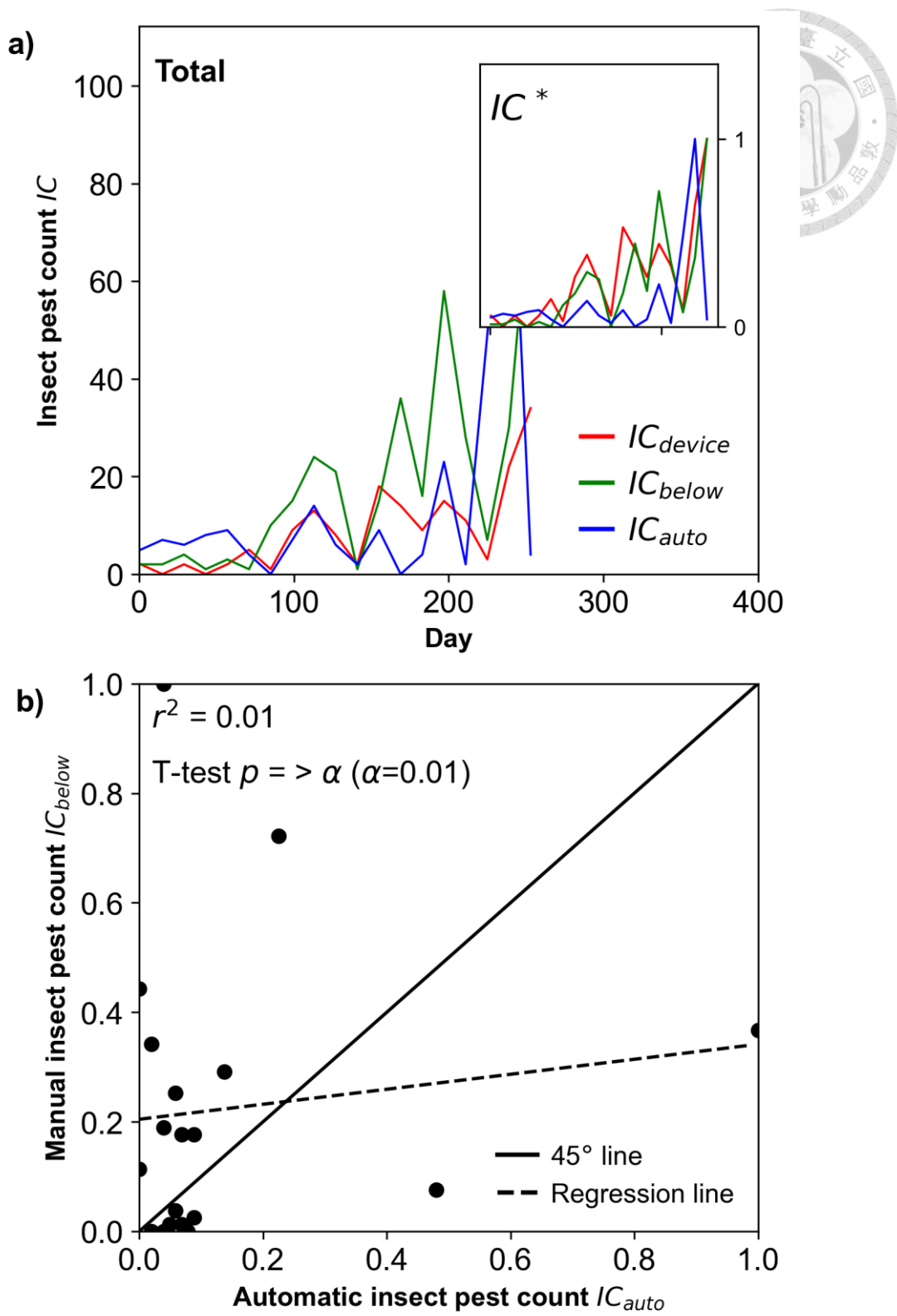


Fig. 4-43. Total thrips trapping efficacy experiment results obtained from Farm TS3.

a) Raw and normalized insect pest counts; b) Linear regression analysis results.

4.3 Data analytics model development



4.3.1 Insect pest count alarm model

Insect pest count alarm models were prepared based on the historical insect pest count data of each installation site. The developed models based on the data of Farms TS1 and TS2 are shown in Fig. 4-44 and Fig. 4-45, respectively. It can be seen from both sets of insect pest count alarm models that both farms have their distinct frequency distributions of increase in insect pest count. It can be also be seen that both did not have many craneflies or mosquitoes in their farm; thus, the alarm model was not used. However, it can be noticed that the increase in count of the important insect pests, thrips and whitefly, had different frequency distributions. It can be seen that Farm TS1 had lower increase in count of thrips, with a SEVERE level of 12 and above. In contrast, Farm TS2 can have an increase in the count of thrips up to more than 46 in a day. Meanwhile, both had close SEVERE level insect pest count thresholds of 74 and 67, for Farms TS1 and TS2, respectively. This reflected that both farms regularly had whiteflies in their farm. Considering the area size of both farms, the alarm models how that it was really necessary to develop and fit individual models for each site. In general, the automatic insect pest count alarm determination method was able to successfully find reasonable increase in insect pest count threshold values for both farms.

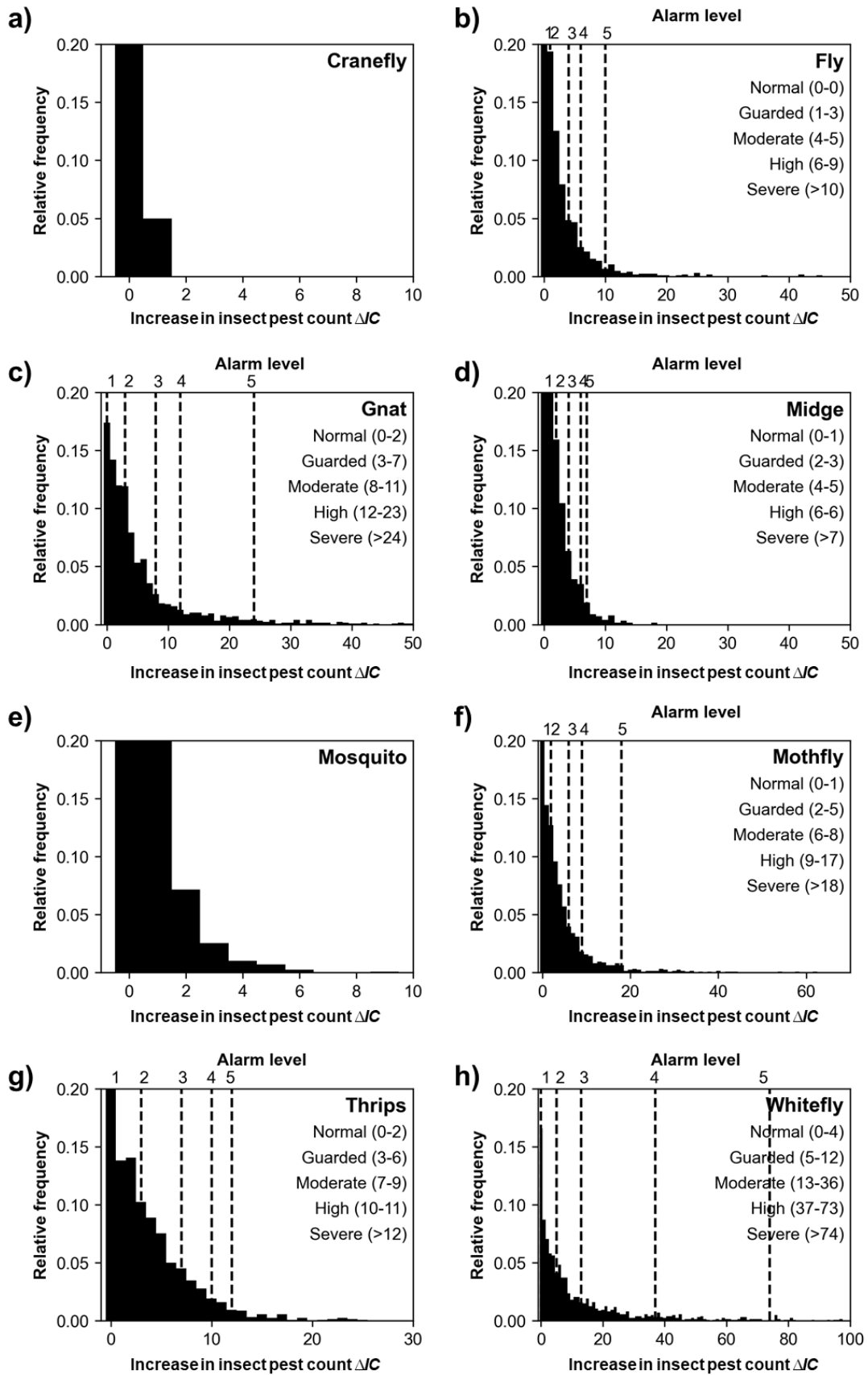


Fig. 4-44. Insect pest count alarm models of Farm TS1.

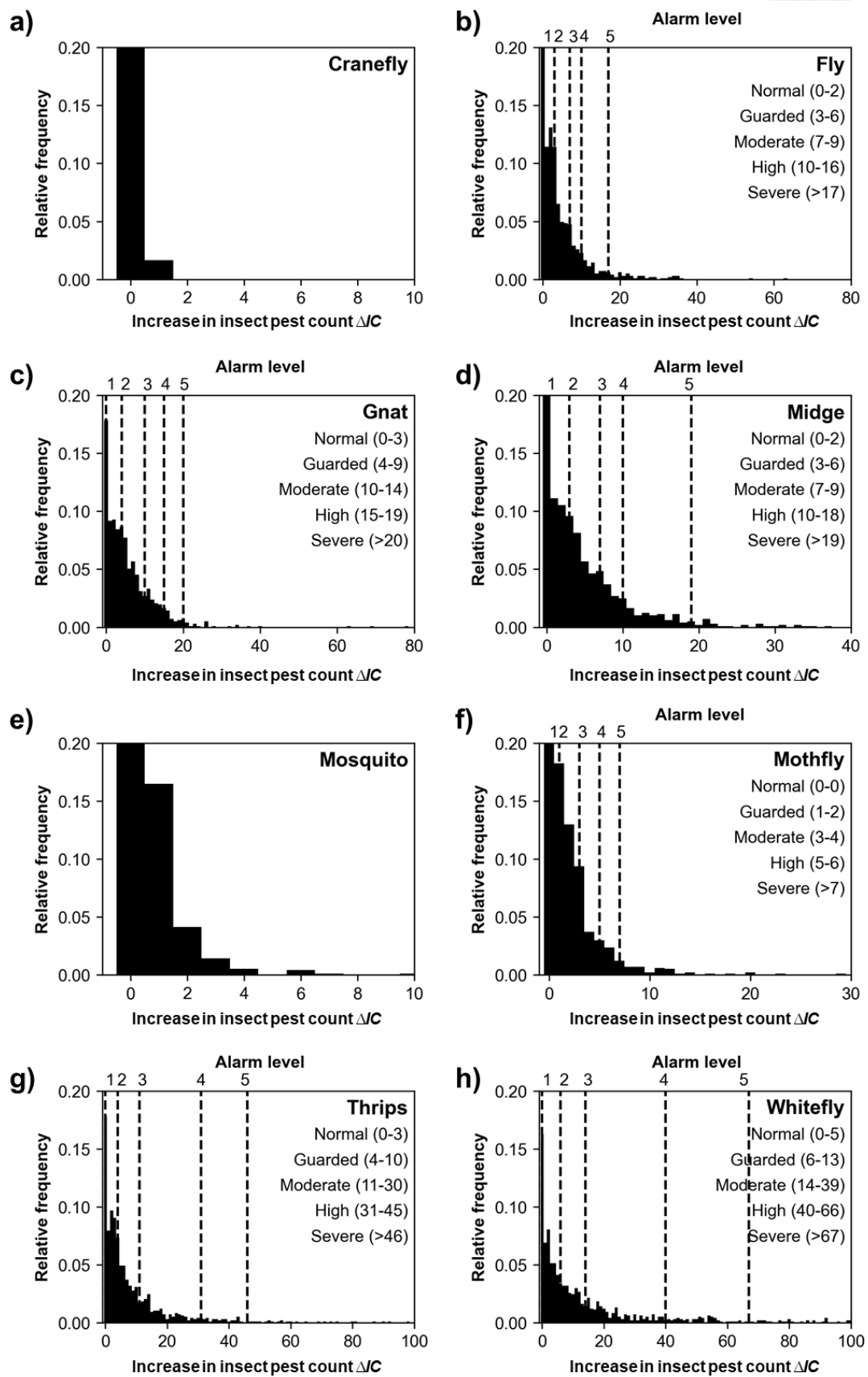
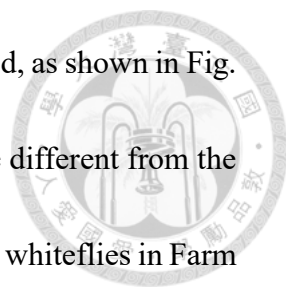


Fig. 4-45. Insect pest count alarm models of Farm TS2.



The insect pest count alarm models of Farm O1 was also obtained, as shown in Fig. 4-46. As expected, the alarm models of Farm O1 was different quite different from the first two farms. It can be noticed that the SEVERE level of thrips and whiteflies in Farm O1 was a lot lower than in Farms TS1 and TS2. This was important to know since orchids are naturally more sensitive to damages of the said insect pests. Based on the feedback of the farm managers, this was reasonable since they need to be notified more promptly if thrips and whiteflies were detected in their farm. In the orchid industry, if an inspector finds a single thrips or whitefly in the leaves or petals of the orchid, it will be rejected for export. It also shows that there were a lot of gnats in Farm O1. It was still considered normal since gnats do not inflict serious damage to orchids. The insect pest count alarm models show that the other five insect pests: crane fly, fly, midge, moth fly, and mosquito, did not have any alarm threshold values. This was because there were very few of the insect pests found in Farm O1.

The results of the first three farms show that the alarm models can also explain the trend of the increase in insect pest counts of each farm. The three cases show that most of the distributions can be explained by an exponential function. It was also found that some of the insect pests were not usually found in each farm. However, it was still important to know what kind of insect pests were usually found in each farm.

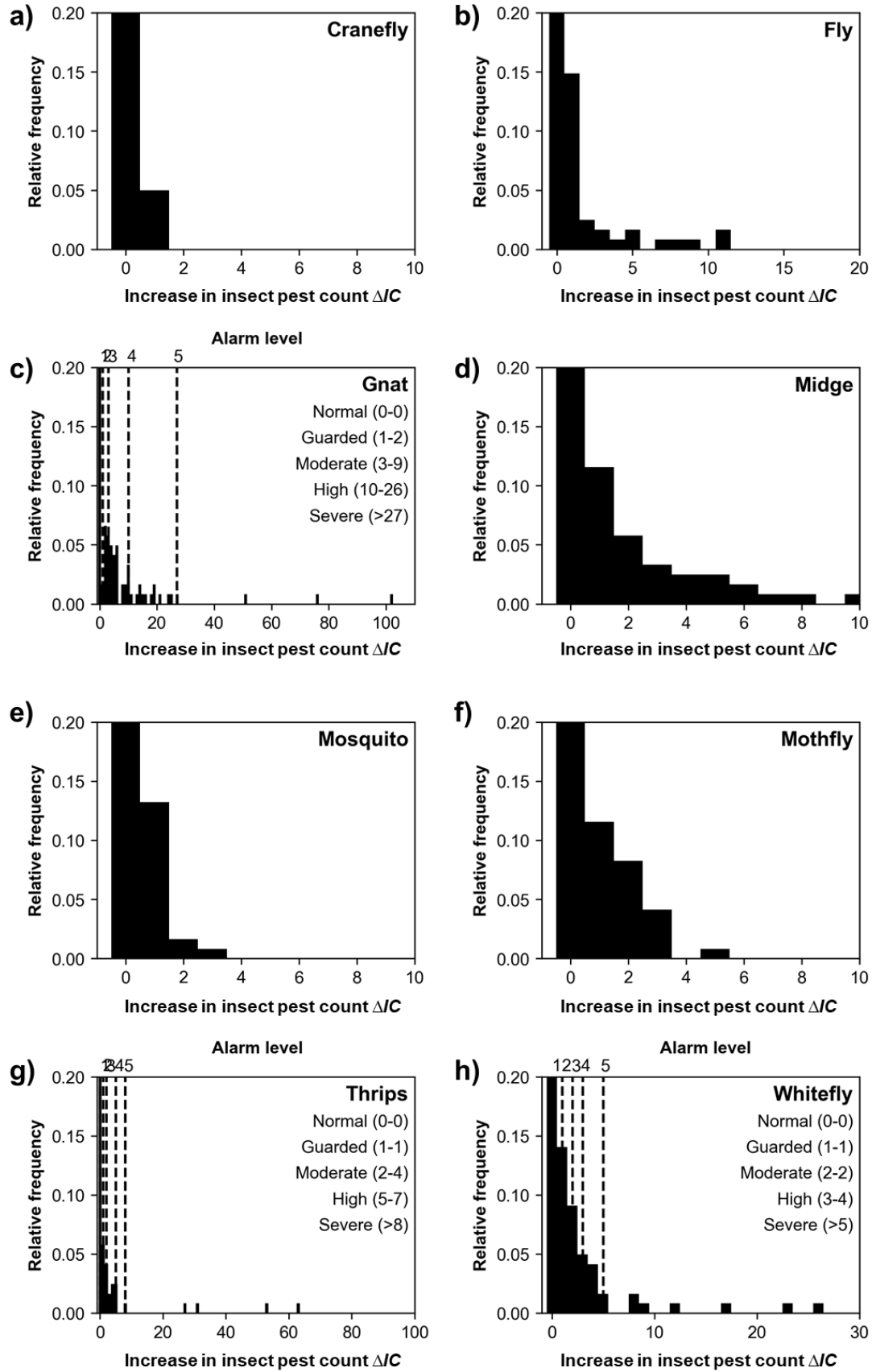
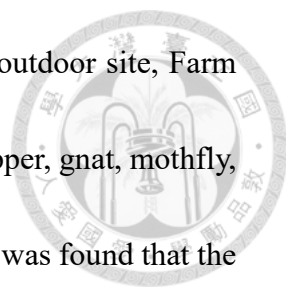


Fig. 4-46. Insect pest count alarm models of Farm O1.



An insect pest count alarm model was also developed for the outdoor site, Farm M1, as shown in Fig. 4-47. The results show that only thrips, leafhopper, gnat, mothfly, and fruitfly had automatically determined alarm threshold values. It was found that the frequency of increase in the count of thrips was quite low. This shows that the farm managers of Farm M1 was able to successfully control the population of thrips. It also shows that there was a constant increase in the number of fruitflies. Fortunately, common fruit flies do not inflict damage to the crops. Only the oriental fruitflies and mango leafhoppers may cause serious damage to the crops but did not frequently appear in the farm. In general, it shows that their strategy was effective to prevent issues caused by insect pest outbreaks.

The insect pest count alarm models of the rest of the installation sites were also obtained, as shown in Table 4-3. In summary, Farms TS1 and TS3 mostly had similar insect pest count alarm models. However, Farm TS2 had an unusual trend in the increase in count of thrips. This shows that Farm TS2 need to focus more on controlling the population of thrips in their farm. Farms O1, O2 and O3 also had generally safe trends in insect pest counts. This was found reasonable due to the sensitivity of their crops. Farms S1 and C1 mostly only had issues in mothflies, which indicate that the cleanliness of the farm should be improved. The insect pest count alarm models were used to analyze the historical insect pest count data of each farm later in Section 4.4.

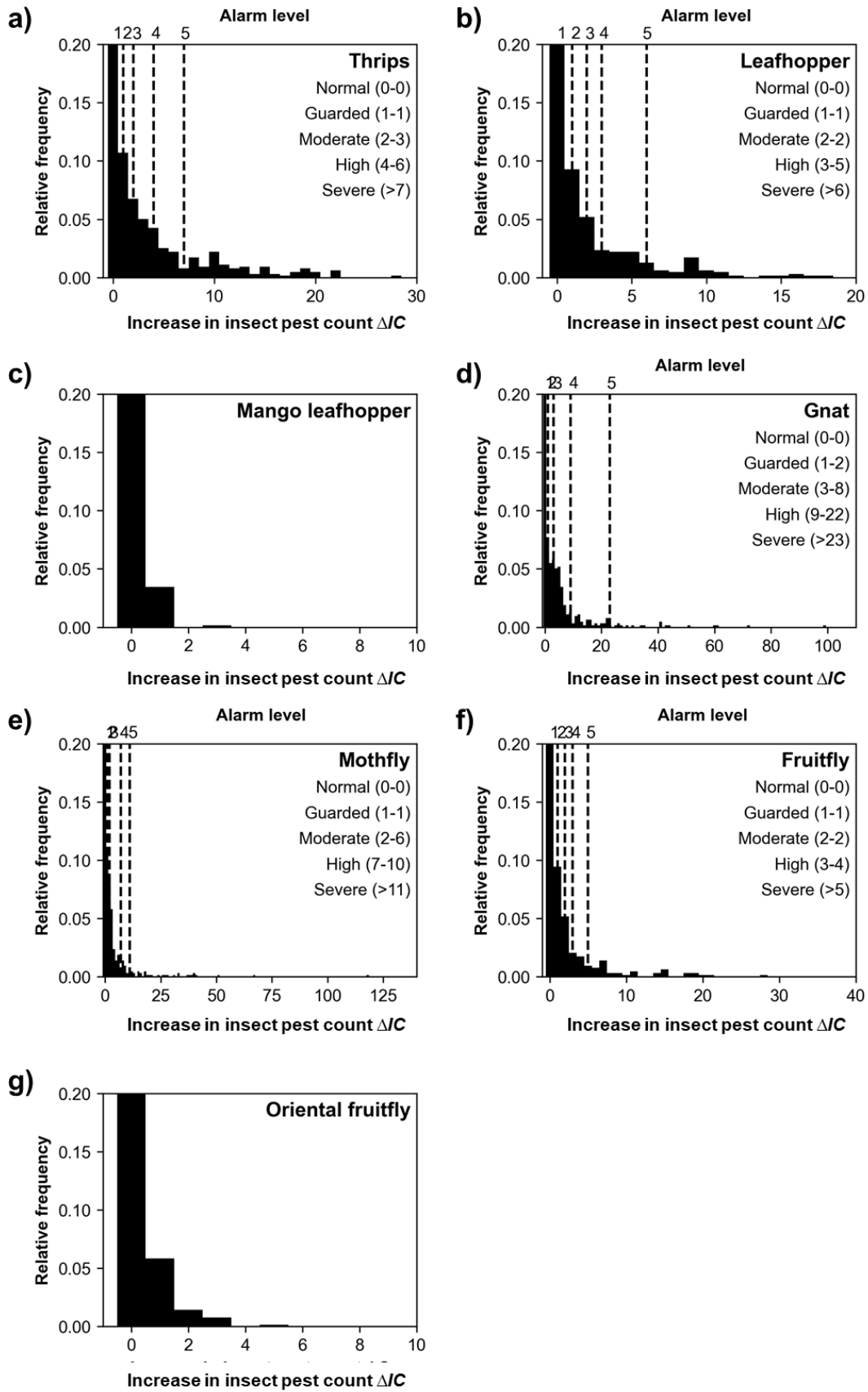


Fig. 4-47. Insect pest count alarm models of Farm M1.

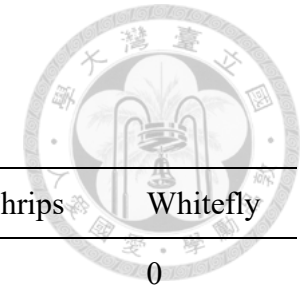
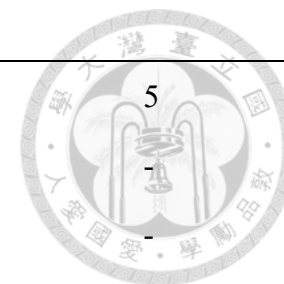


Table 4-3. Insect pest count alarm threshold values of all the installation sites.

Farm name	# of days	Alarm level	Cranefly	Fly	Gnat	Midge	Mosquito	Mothfly	Thrips	Whitefly
TS1	1260	Normal	-	0	0	0	-	0	0	0
		Guarded	-	1	3	2	-	2	3	5
		Moderate	-	4	8	4	-	6	7	13
		High	-	6	12	6	-	9	10	37
		Severe	-	10	24	7	-	18	12	74
TS2	970	Normal	-	0	0	0	-	0	0	0
		Guarded	-	3	4	3	-	1	4	6
		Moderate	-	7	10	7	-	3	11	14
		High	-	10	15	10	-	5	31	40
		Severe	-	17	20	19	-	7	46	67
TS3	382	Normal	-	0	0	0	-	0	-	0
		Guarded	-	1	2	1	-	1	-	1
		Moderate	-	2	5	2	-	2	-	3

T1	268	High	-	6	15	3	-	3	-	8
		Severe	-	10	44	5	-	4	-	34
		Normal	-	0	0	0	-	-	0	0
		Guarded	-	1	2	2	-	-	1	2
		Moderate	-	2	6	6	-	-	3	5
T2	268	High	-	7	19	18	-	-	9	15
		Severe	-	21	25	20	-	-	11	301
		Normal	-	-	0	-	-	-	-	0
		Guarded	-	-	1	-	-	-	-	1
		Moderate	-	-	2	-	-	-	-	2
O1	851	High	-	-	3	-	-	-	-	8
		Severe	-	-	5	-	-	-	-	84
		Normal	-	-	0	-	-	-	0	0
		Guarded	-	-	1	-	-	-	1	1
		Moderate	-	-	3	-	-	-	2	2
		High	-	-	10	-	-	-	5	3





O2	674	Severe	-	-	27	-	-	-	8	5
		Normal	-	0	0	-	-	0	-	-
		Guarded	-	1	1	-	-	1	-	-
		Moderate	-	2	2	-	-	2	-	-
		High	-	5	3	-	-	3	-	-
O3	786	Severe	-	6	6	-	-	5	-	-
		Normal	-	0	0	0	0	0	0	0
		Guarded	-	1	1	1	1	1	1	1
		Moderate	-	2	3	3	2	2	3	3
		High	-	7	10	8	5	3	8	9
S1	654	Severe	-	15	64	39	6	6	15	23
		Normal	-	0	-	0	-	0	0	0
		Guarded	-	1	-	5	-	10	1	1
		Moderate	-	5	-	10	-	11	3	3
		High	-	7	-	11	-	23	8	4
		Severe	-	10	-	15	-	35	10	5



C1	601	Normal	-	0	-	0	-	0	0	0
		Guarded	-	3	-	2	-	2	3	5
		Moderate	-	5	-	4	-	6	5	13
		High	-	10	-	6	-	9	7	23
		Severe	-	12	-	7	-	23	8	55
			Thrips	Leafhopper	Mango leafhopper	Gnat	Mothfly	Fruitfly	Oriental fruitfly	
M1	645	Normal	0	0	-	0	0	0	-	
		Guarded	1	1	-	1	1	1	-	
		Moderate	3	2	-	3	2	2	-	
		High	6	3	-	9	7	3	-	
		Severe	7	6	-	23	13	4	-	

4.3.2 Insect pest flight rate model



The historical insect pest count data and temperature data was analyzed to develop the insect pest flight rate models. The models of Farms TS1 and TS2 are shown in Fig. 4-48 and 4-49. It can be seen from the results that there were slight differences between the models formed from the two sites despite having similar crops. This shows that the environmental control strategy of both farms had an effect to the insect pest flight behavior. It can be seen that the insects that belonged to the order *Diptera* namely, crane fly, fly, gnat, midge, mothfly, and mosquito, had closely similar flight behaviors. Particularly, the insects were less active during cold weather and more active during hot weather. The insects also had wide niche temperature widths, which agreed with literature (Damos & Savopoulou-Soultani, 2012).

In Farm TS1, thrips and whiteflies were more active during cold weather (Fig. 4-48g and Fig. 4-48h). On the other hand, thrips and whiteflies were more active during hot weather in Farm TS2 (Fig. 4-49g and Fig. 4-49h). According to the farm managers of Farm TS1, this happened because they tend to ignore the presence of insect pests because based on their experience, the crops do not usually get affected by diseases during the winter season. But upon inspecting the raw data, it did show that there were almost two peaks in the flight rate of thrips and whiteflies in Farm TS1, which was

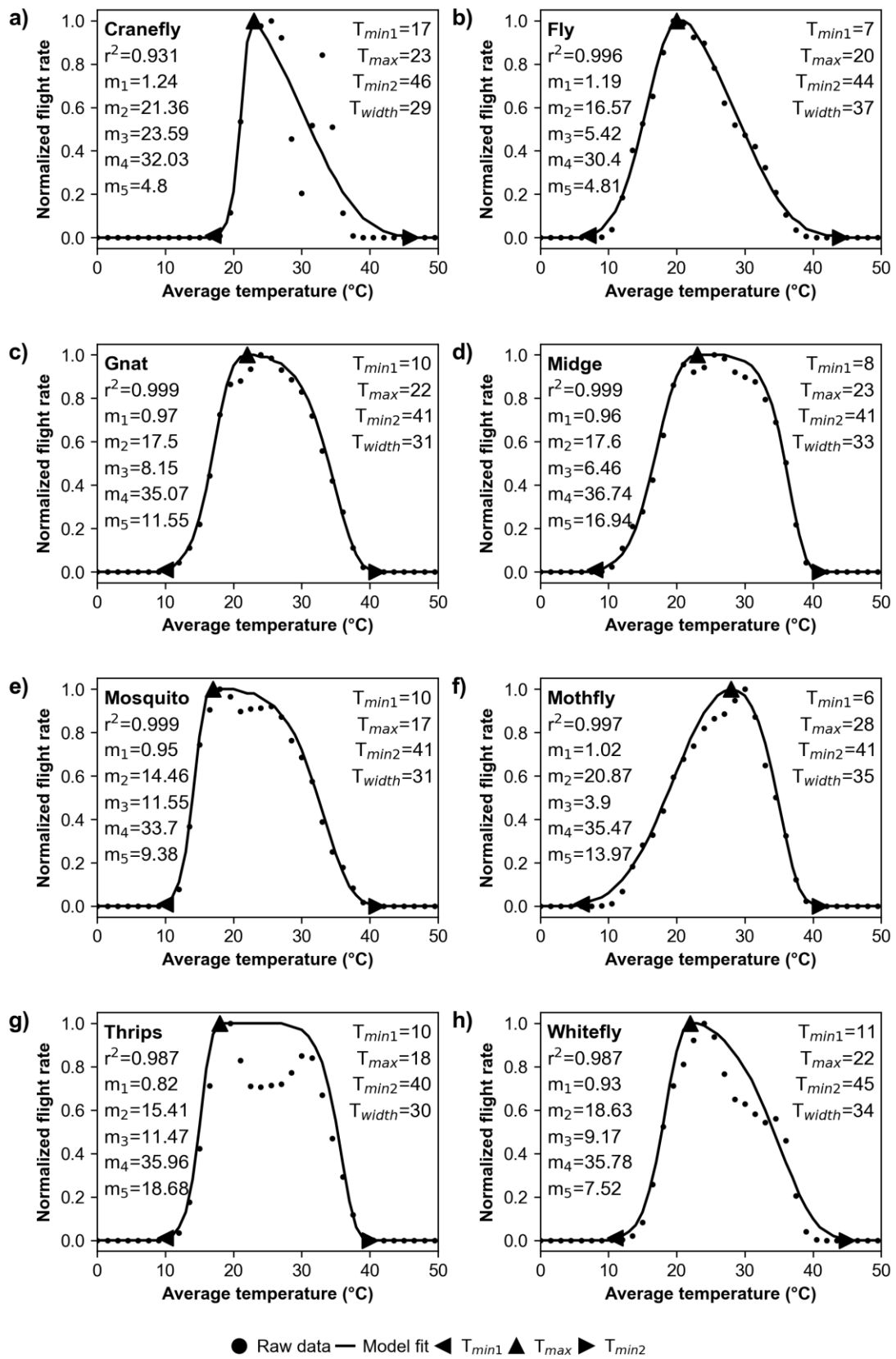


Fig. 4-48. Fitted temperature vs. flight rate model of insects found in Farm TS1.

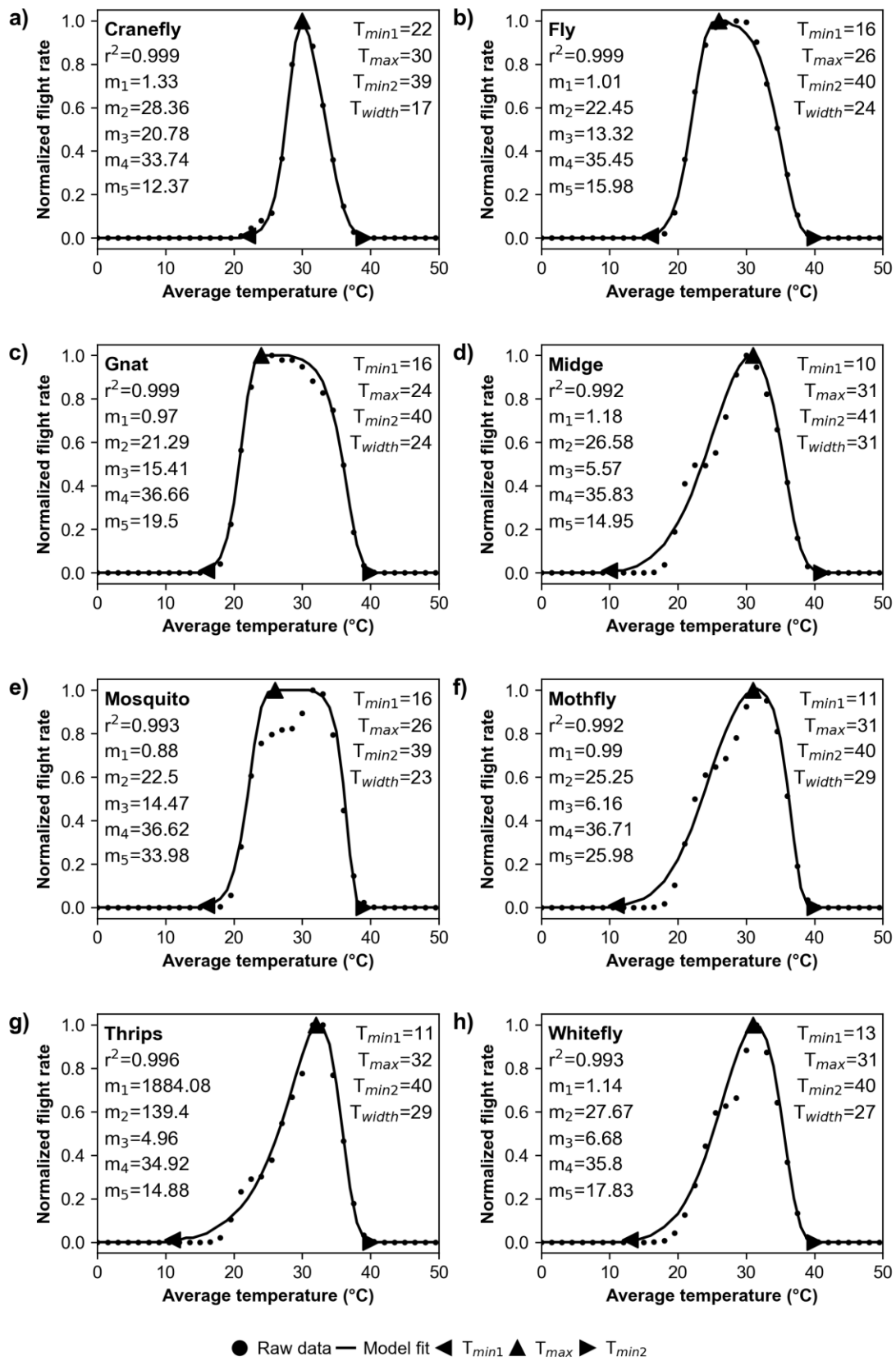
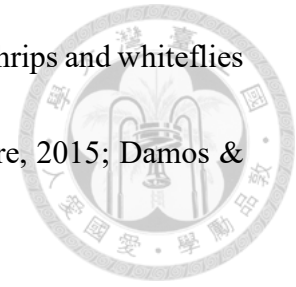


Fig. 4-49. Fitted temperature vs. flight rate model of insects found in Farm TS2.

close to the results obtained from Farm TS2. The flight behavior of thrips and whiteflies in Farm TS2 were more in agreement to related studies (Bonsignore, 2015; Damos & Savopoulou-Soultani, 2012; Rhainds et al., 2007).



In general, it shows that the double Weibull function was able to properly describe the flight behavior of the insect pests with r^2 values higher than 0.9. The results also show that the data from literature were not absolute in all cases. Developing adaptive biological models can be more useful to the farm managers since it adjusts according to their management strategies and unique environmental conditions.

The flight behavior models of the insect pests in Farm O1 were also developed, as shown in Fig. 4-50. It can be noticed from the results that most of the insect pests in the orchid farm had narrower niche temperature widths. This was most likely since the environment was strictly controlled by the farm managers. Similarly, the thrips and whiteflies in Farm O1 were more active at moderately warm temperatures of around 25°C to 31°C (Fig. 4-50g and Fig. 4-50h). The other insect types were also more active during the similar condition. The peak of the models in Farm O1 were also relatively sharper than the models of Farms TS1 and TS2. This shows that there were certain conditions that the insect pests were most active. Despite that, the biological modelling method was still able to describe the insect behaviors with r^2 values higher than 0.95.

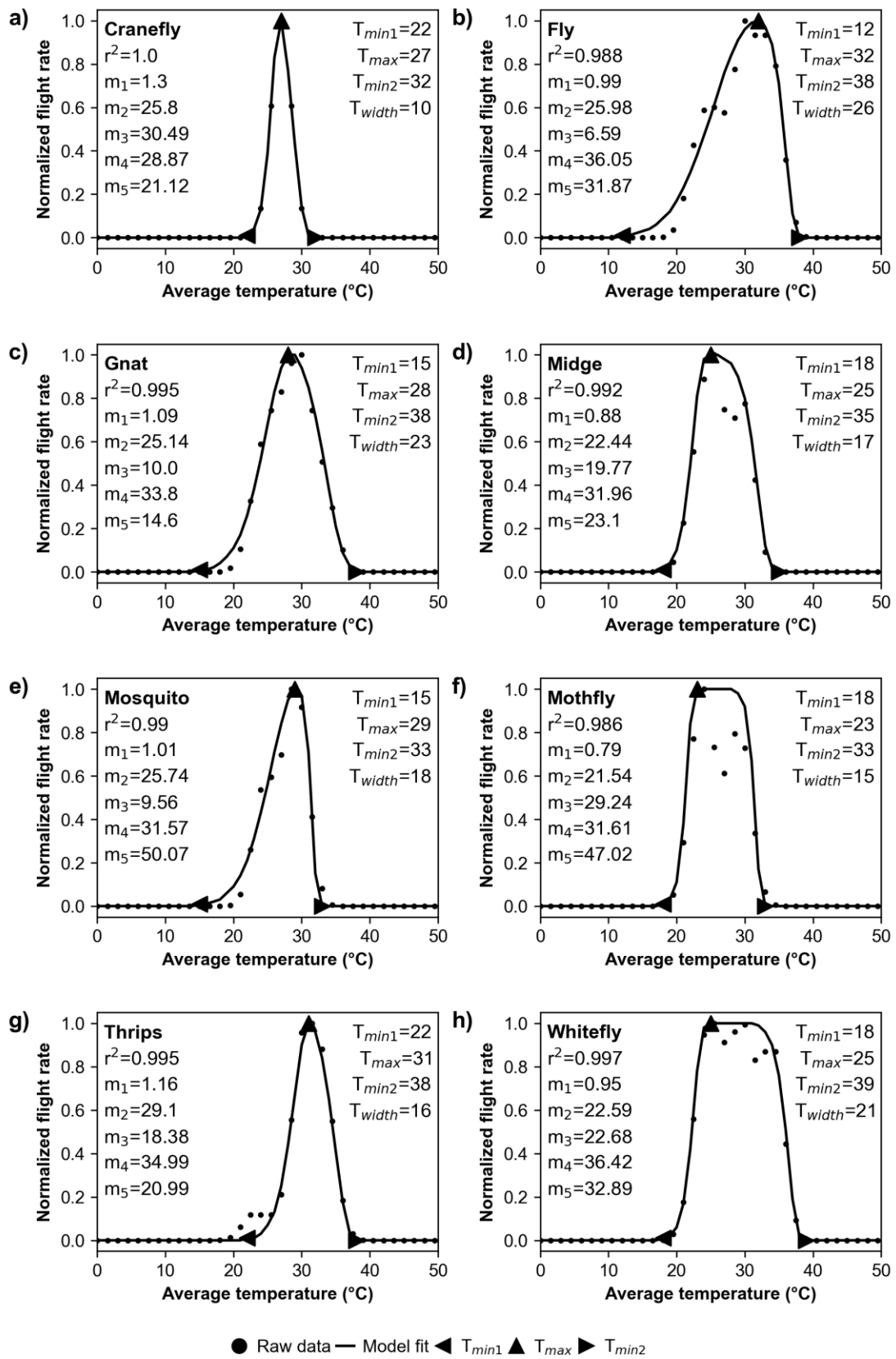
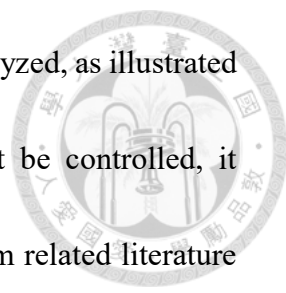


Fig. 4-50. Fitted temperature vs. flight rate model of insects found in Farm O1.



The flight behavior of the insect pests in Farm M1 was also analyzed, as illustrated in Fig. 4-51. Since the environmental condition outdoors cannot be controlled, it showed that most of the insect pests followed the data referred from related literature (Jha et al., 2009; Rieger et al., 2007; Wang et al., 2012). In particular, the thrips in Farm M1 were most active at temperature levels from 23°C to 30°C, similar to the findings in literature of 27°C. It can be also seen that the oriental fruitflies and mango leafhoppers had similar flight behaviors. However, it was found that this was caused by both natural conditions and the growth condition of the crops. More discussion on the behavior of the two insects are presented in Section 4.4.2.

The insect pest flight behavior of the insects of the rest of the installation sites was also analyzed and summarized in Table 4-3. Most of the developed biological models were in correlation to the results of related literature. It was found that the niche temperature width in the tomato seedling and tomato farms were wider compared to the orchid farms. This was due to the stricter environmental control strategies applied in the orchid farms to stimulate the growth of the crops. This led to unique models that were formed due to external factors in the installation sites, most especially the farm managers management strategy. The results obtained showed that the monitoring system was able to collect valuable environmental information that can guide the farm managers about the behavior of the insect pests in their farm.

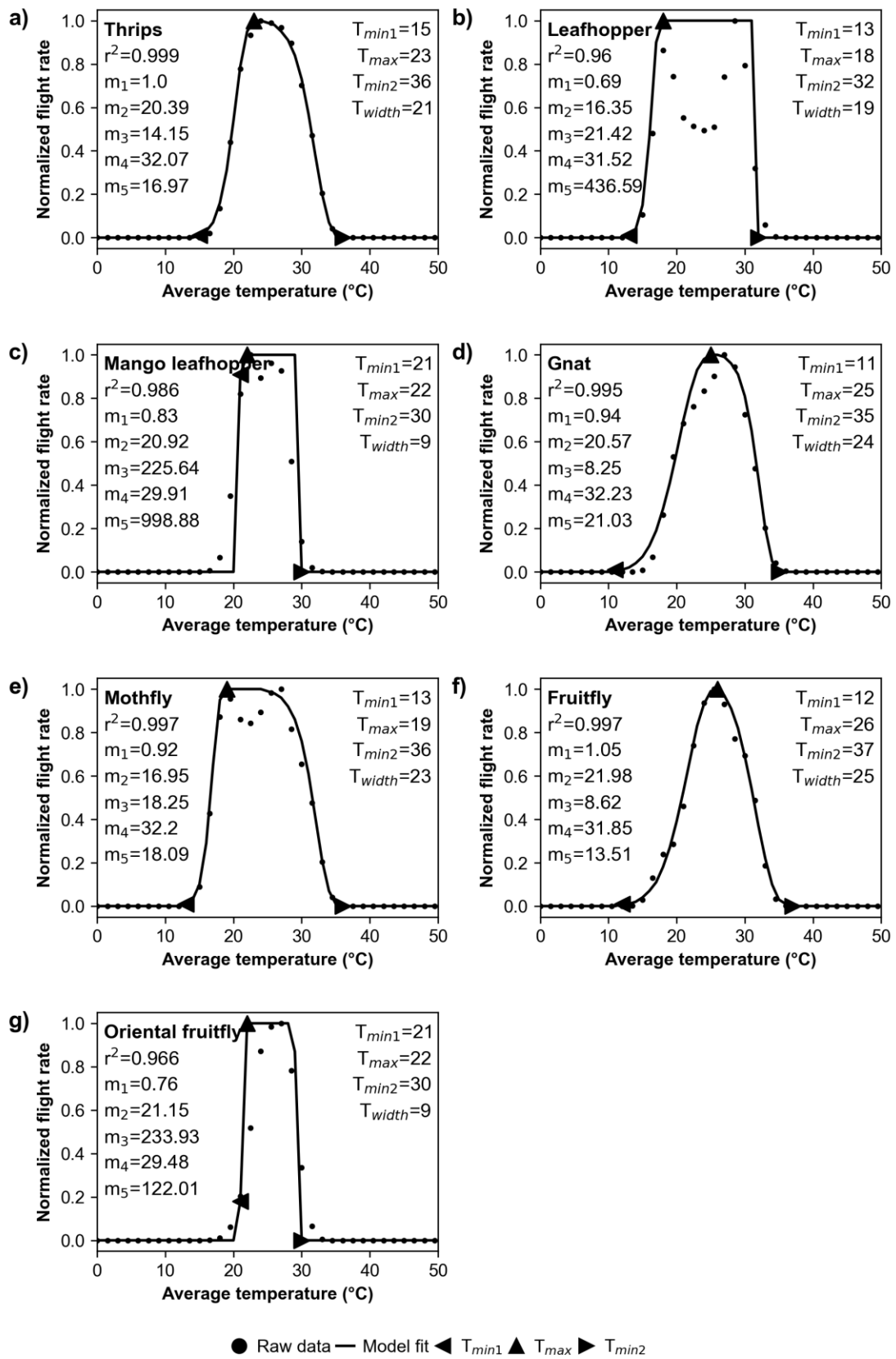


Fig. 4-51. Fitted temperature vs. flight rate model of insects found in Farm M1.

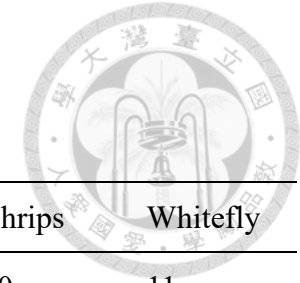
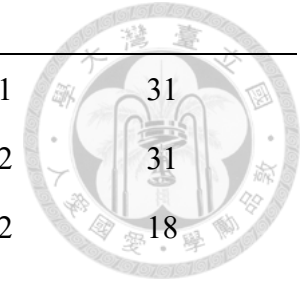


Table 4-4. Insect pest flight rate model niche temperature per installation site.

Farm name	# of days	Niche $T(^{\circ}\text{C})$	Cranefly	Fly	Gnat	Midge	Mosquito	Mothfly	Thrips	Whitefly
TS1	1260	$T_{niche,min1}$	17	7	10	8	10	6	10	11
		$T_{niche,max}$	23	20	22	23	17	28	18	22
		$T_{niche,min2}$	46	44	41	41	41	41	40	45
TS2	970	$T_{niche,min1}$	22	16	16	10	16	11	11	11
		$T_{niche,max}$	30	26	24	31	26	31	32	31
		$T_{niche,min2}$	39	40	40	41	39	40	40	40
TS3	382	$T_{niche,min1}$	20	16	13	16	17	21	16	15
		$T_{niche,max}$	27	25	24	27	22	22	31	28
		$T_{niche,min2}$	37	41	43	42	38	33	38	42
T1	268	$T_{niche,min1}$	26	17	15	17	18	16	16	21
		$T_{niche,max}$	32	26	26	24	25	23	24	30
		$T_{niche,min2}$	38	38	42	44	35	34	43	40
T2	268	$T_{niche,min1}$	-	18	15	19	19	27	19	12



		<i>T_{niche,max}</i>	-	24	26	25	24	28	31	31
		<i>T_{niche,min2}</i>	-	35	35	35	42	31	32	31
O1	851	<i>T_{niche,min1}</i>	22	12	15	18	15	18	22	18
		<i>T_{niche,max}</i>	27	27	28	25	29	23	31	25
		<i>T_{niche,min2}</i>	32	38	38	35	33	33	38	39
O2	674	<i>T_{niche,min1}</i>	-	15	18	22	23	17	15	18
		<i>T_{niche,max}</i>	-	27	29	23	23	24	28	28
		<i>T_{niche,min2}</i>	-	33	35	43	32	33	34	34
O3	786	<i>T_{niche,min1}</i>	22	12	9	6	14	9	6	6
		<i>T_{niche,max}</i>	30	24	27	39	22	28	39	31
		<i>T_{niche,min2}</i>	36	50	48	43	45	50	43	45
			Thrips	Leafhopper	Mango leafhopper	Gnat	Mothfly	Fruitfly	Oriental fruitfly	
M1	601	<i>T_{niche,min1}</i>	15	13	21	11	13	12	21	
		<i>T_{niche,max}</i>	23	18	22	25	19	26	22	
		<i>T_{niche,min2}</i>	36	32	30	35	36	37	30	



4.4 Insect pest count data analysis

This section discusses about the data collected from the installation sites. The insect pest count alarm models in Section 4.3.1 and insect pest temperature vs. flight rate models in Section 4.3.2 of each site were incorporated in the data analyses to extract more valuable information from the data. Data analysis was broken into two parts according to the indoor site and outdoor site data to differentiate the obtained results.

4.4.1 Indoor site data analysis

The data from two indoor sites were selected for comparison in the data analysis: Farm TS1 and TS2. The two were selected since they had similar crops and had the longest collection of data. It was also a good comparison since the farm managers of TS1 still sprayed pesticides in a regular basis despite the availability of the system. Meanwhile, the farm managers of Farm TS2 were constant users of the system and monitored the insect pest counts in their farm daily through the mobile APP. To demonstrate the usefulness of the system in developing smarter IPM strategies, sample effective pesticide application data is shown in Fig. 4-52. The data shown in Fig. 4-52 are of Farm TS2. The data provided shows periods in the data when there was an insect

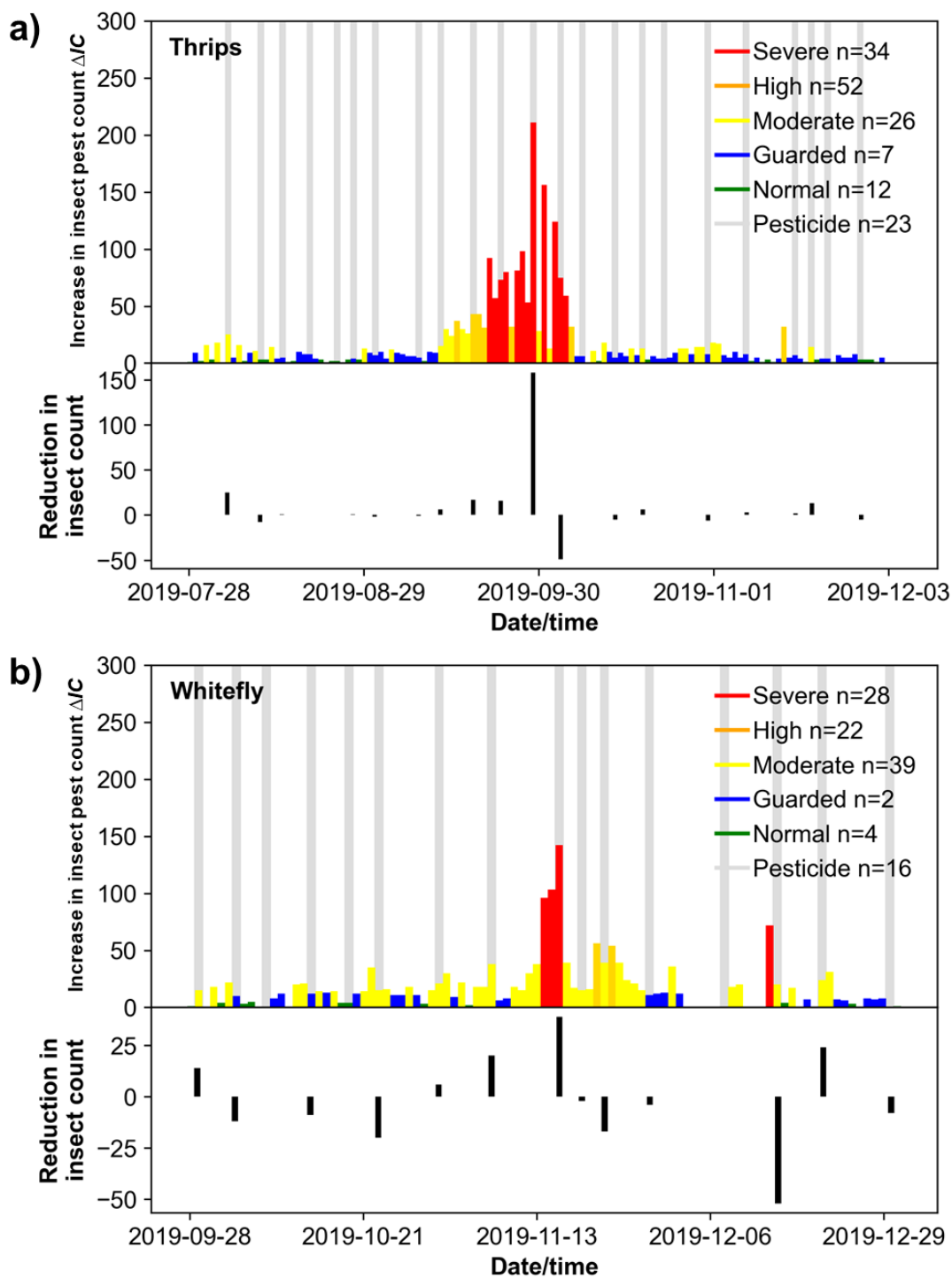
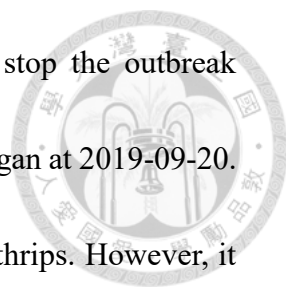


Fig. 4-52. Sample effective pesticide application data of Farm TS2.



pest outbreak and the farm managers of Farm TS2 was able to stop the outbreak effectively. It can be seen in Fig. 4-52a that the outbreak of thrips began at 2019-09-20. After two days, they sprayed pesticide to reduce the population of thrips. However, it shows that there was still a constant daily increase in the number of thrips detected. But as there was a peak in the increase in thrips count in 2019-09-29, they sprayed pesticides again to stop the outbreak. This greatly reduced the increase on the number of thrips by 137 counts the next day. However, the outbreak did not end that quickly. Finally, after spraying pesticides again at 2019-10-03, they were able to successfully stop the outbreak and the alarm level for thrips went back to guarded.

Meanwhile, during another time in 2019-11, there was an outbreak of whiteflies in Farm TS2. But it can be seen in Fig. 4-52b that it immediately ended after the farm managers sprayed pesticides in 2019-11-17. This shows that preventing insect pest outbreaks require immediate attention. Based on the feedback of the farm managers of Farm TS2, this was not possible if the system was not able to notify them immediately during those times.

The long-term insect pest count data of Farm TS1 is shown in Fig. 4-53. The insect pest count data was presented as moving averages to clearly see the trend and patterns in the insect pest count data. It can be immediately seen that the farm managers of Farm TS1 followed a regular pesticide spraying schedule. It can also be seen that the daily

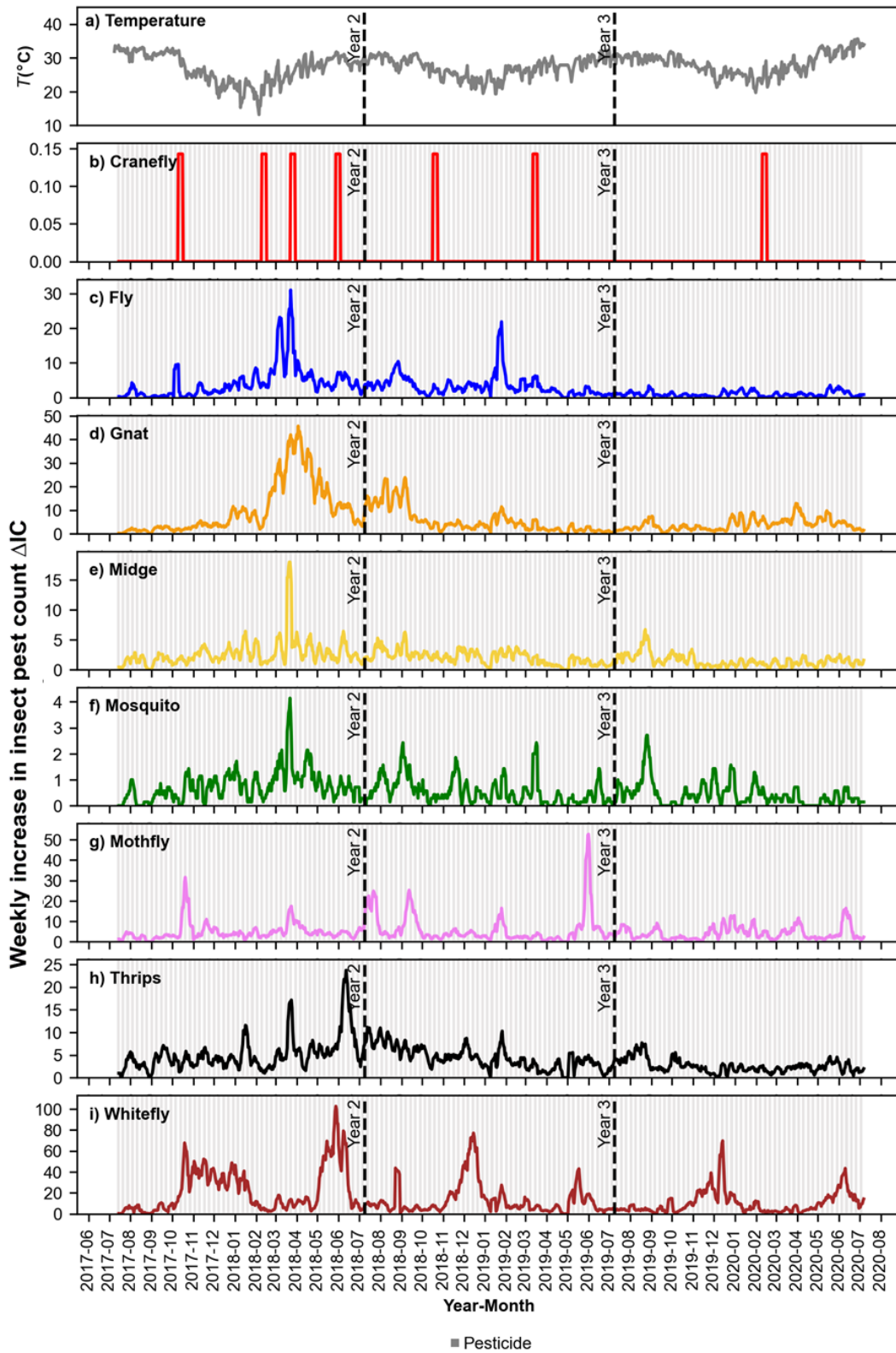
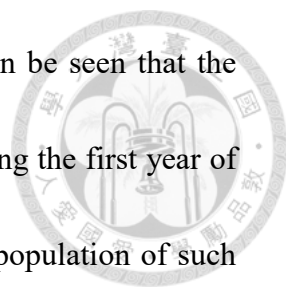


Fig. 4-53. Long-term temperature and insect pest count data collected from Farm TS1.

Insect pest counts are presented as moving averages with a window size of 7.



temperature level per year was consistent (Fig. 4-53a). First, it can be seen that the insects belonging to the order *Diptera* were mostly found only during the first year of collecting data in the farm. They were slowly able to manage the population of such insect pests based on the data provided to them. It can also be seen that there was a trend in the number of thrips detected during the first year (Fig. 4-53h). Particularly, there were peaks every two months starting 2018-02. Based on the initial feedback of the farmers, this was caused by changing weather that was still able to affect the condition of the farm. It was also observed that the increase in whitefly counts followed a seasonal pattern on all three years. Whiteflies started to appear every June and October of each year (Fig. 4-53i). The farm managers mentioned that these times were indeed the peak times they see more whiteflies on the sticky paper traps. This was because these were the times that there was a transition from cold to hot weather or vice versa. It was found from our previous studies that this really caused insect pest to be more active due to the sudden changes in temperature (Rustia et al., 2020b; Rustia & Lin, 2019).

The long-term data of Farm TS2 is shown in Fig. 4-54. Similar to the data of Farm TS1, the trend in temperature was quite similar. This was because the weather on both locations were almost identical. It can be observed that there was no particular pattern

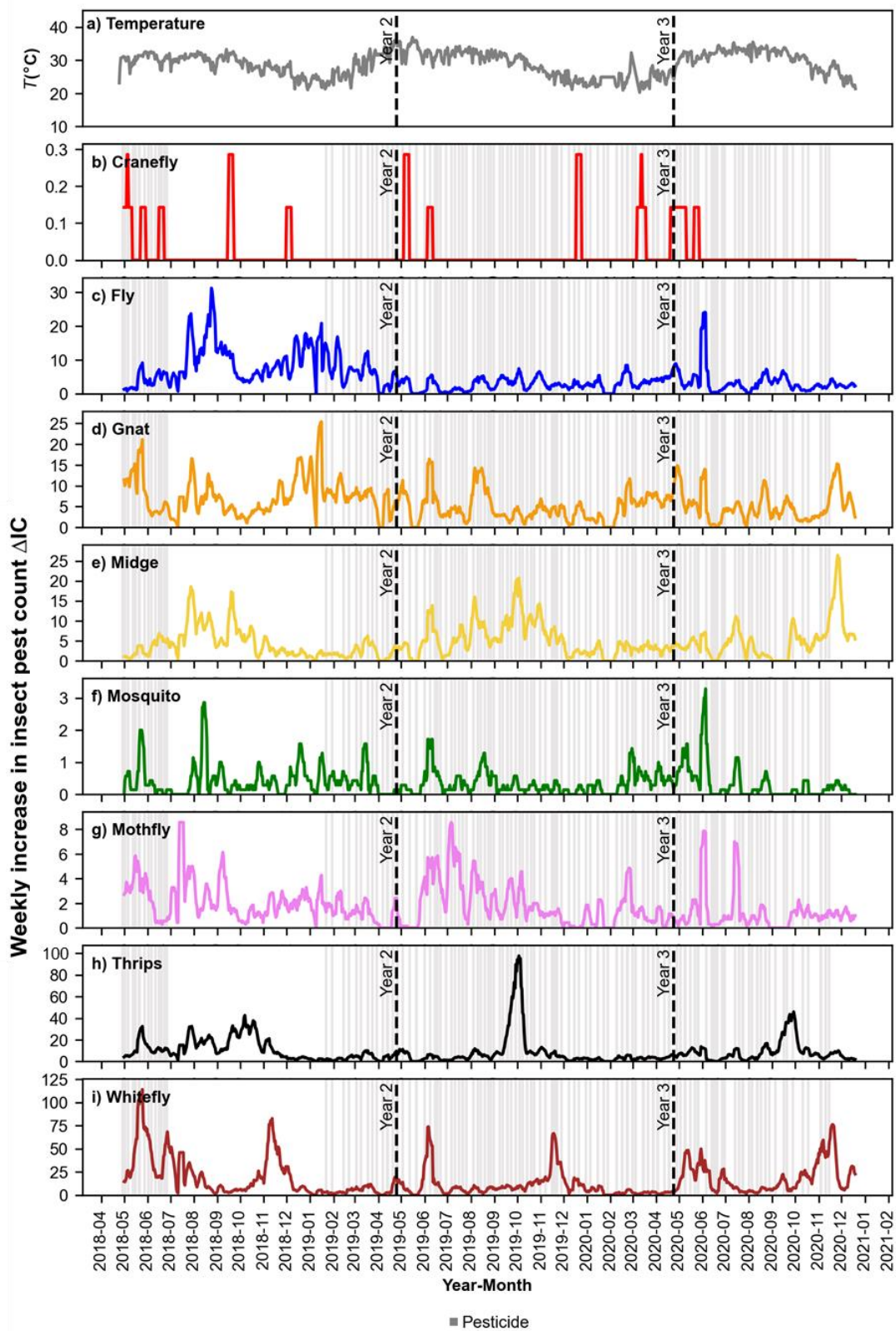
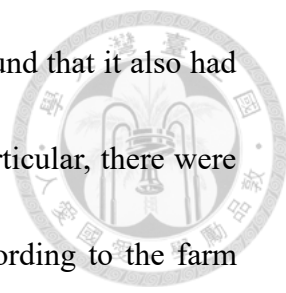


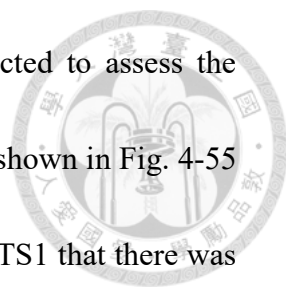
Fig. 4-54. Long-term temperature and insect pest count data collected from Farm TS2.

Insect pest counts are presented as moving averages with a window size of 7.



in the counts of *Diptera* insects in Farm TS2. Apparently, it was found that it also had unique patterns in the increase of thrips and whitefly counts. In particular, there were outbreaks of thrips every October of each year (Fig. 4-54h). According to the farm managers, those were the times that the farm was fully occupied with seedlings. Therefore, there was frequent moving of seedling trays and the door had to be opened whenever the seedlings had to be transported. This has caused thrips to enter the farm and lay eggs for reproduction. It was also found that there were more whiteflies every June and November of each year (Fig. 4-54i). This was almost the same as the trend in Farm TS1. Apparently, the trend was also caused by the same reason stated by the farm managers of Farm TS1; there were abrupt changes of weather in the farm.

It can be seen from both long-term dataset that the farms experienced insect pest outbreaks during very specific times of the year. This meant that there were several factors that caused it. First, the movement and activity in the farms was able to bring insect pests into the farms. This includes preparation and movement of seedling trays, entry of farm personnel, and more. Weather also had an effect in the activity and behavior of the insect pests. Despite having adequate equipment for controlling the environmental condition in the farms, the weather condition surrounding the greenhouse still drew insect pests to the farm. Some causes may include sudden gusts of wind, abrupt changes in temperature, or varying humidity levels.



The accumulated insect pest counts on each farm was also collected to assess the efficacy of the system in reducing the insect pests in both farms, as shown in Fig. 4-55 and summarized in Table 4-5. It can be seen from the data of Farm TS1 that there was definitely a reduction in the detected whitefly (Fig. 4-55a) and thrips (Fig. 4-55b) in the farm. After calculation, it was found that the number of whiteflies detected reduced as much as 40% from Year 1 to Year 2, and 19% from Year 2 to Year 3. This shows that the proposed system was able to successfully assist the farmers in their IPM decision-making. The data of Farm TS2 showed quite similar results. As a matter of fact, there was a higher reduction in thrips count of 25% from Year 1 to Year 2 than the 3 years of Farm TS1. The number of whiteflies was also halved from Year 1 to Year 2 in Farm TS2. But from Year 2 to Year 3, the number of whiteflies increased again and it is still to be determined whether the number of thrips will decrease as well. However, it shows from the historical data that the number of thrips will succeed during the succeeding months. These results proved that the system and proper guidance was beneficial to the farm managers.

The number of alarms raised in farms TS1 and TS2 was also tallied, as shown in Table 4-6. It can be inspected that the number of NORMAL alarm levels increased after each year. At the same time, the number of SEVERE alarm levels dropped. This shows that the farm managers were able to avoid less insect pest outbreaks after each year

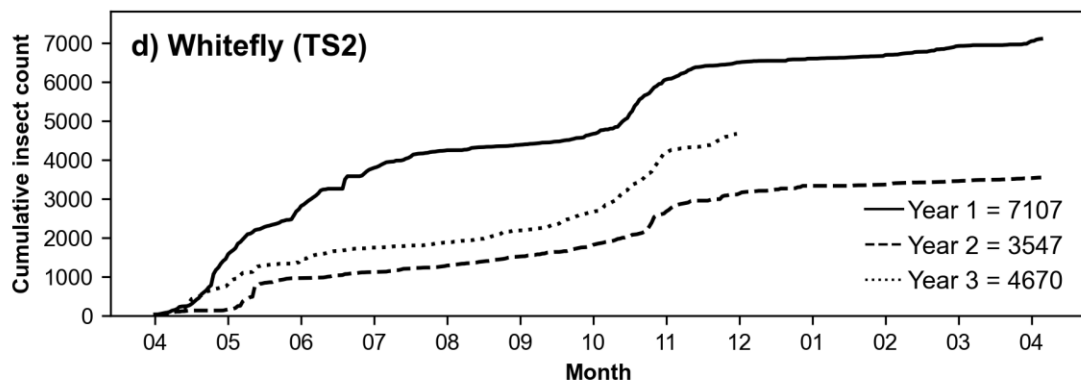
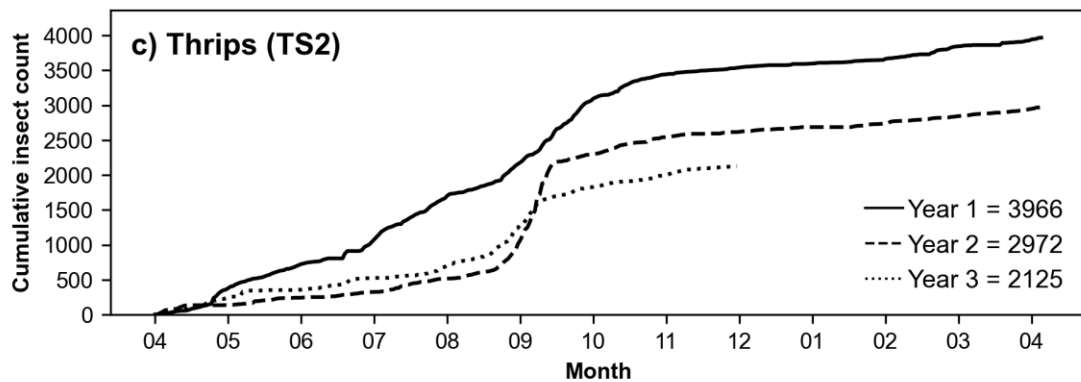
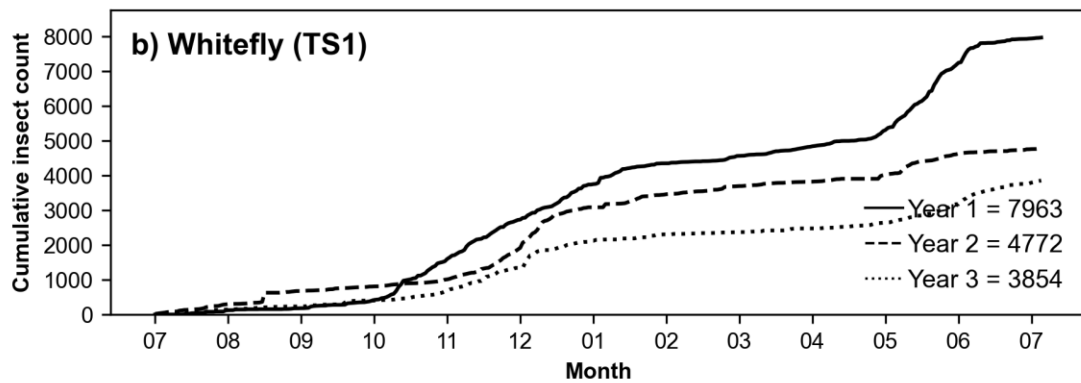
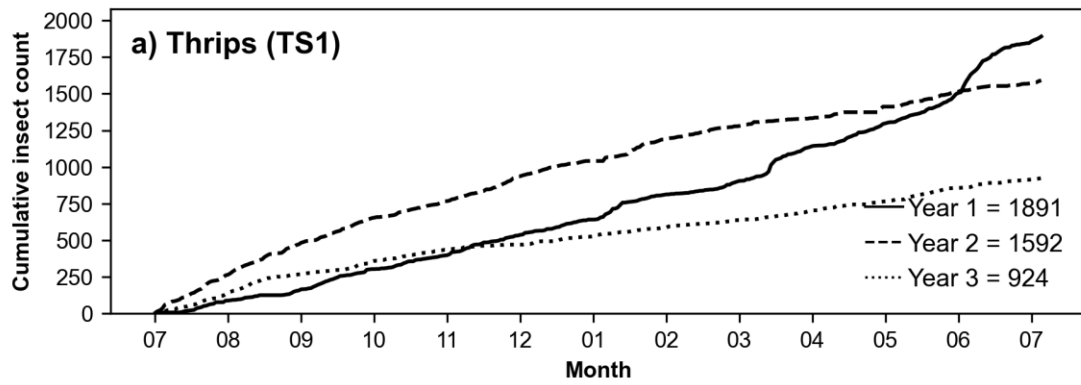


Fig. 4-55. Daily accumulated thrips and whitefly counts of Farms TS1 and TS2 per year. The total counts per year are showed in the legends.

Table 4-5. Yearly reduction in thrips and whitefly counts of Farms TS1 and TS2.

Farm	Year	Total thrips detected	% reduction (thrips)	Total whiteflies detected	% reduction (whiteflies)
TS1	1	1891	-	7963	-
	2	1592	16%	4772	40%
	3	924	19%	3854	19%
TS2	1	3966	-	7107	-
	2	2972	25%	3547	50%
	3	2125	TBD	4670	TBD

TBD: To be determined

through the use of the system. It was also observed that the total number of SEVERE alarms raised in Farm TS1 was relatively more than Farm TS2. It was suspected that the reason was that the farm managers of Farm TS1 will check the notifications in their mobile APP but still spray pesticides according to their own discretion. This led to more SEVERE alarm level notifications since the insect pest outbreaks were not immediately stopped. In contrast, Farm TS2 has less SEVERE alarms since they followed the recommendations given by the proposed system. This also led to higher reduction in insect pests as shown previously in Table 4-5. This also shows one limitation of the system. The system can only notify and give recommendations to the farm managers on how to handle the insect pest condition in their farms. This was also a safer approach since the management of the farm is still in the hands of the farm managers, themselves.

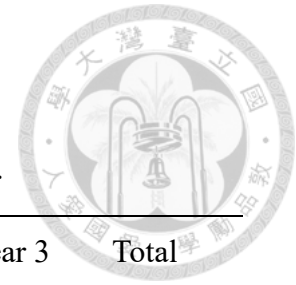
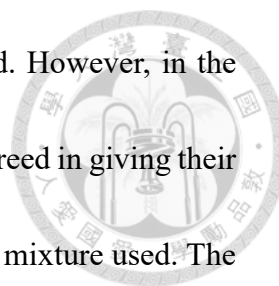


Table 4-6. Thrips and whitefly counts alarm level statistics per year.

Location	Insect pest	Alarm level	Year 1	Year 2	Year 3	Total
TS1	Thrips	Normal	139	147	226	512
		Guarded	118	130	106	354
		Moderate	43	32	19	94
		High	38	37	12	87
		Severe	27	19	2	48
	Whitefly	Normal	120	152	186	458
		Guarded	86	108	77	271
		Moderate	79	73	82	234
		High	66	26	17	109
		Severe	14	6	3	23
TS2	Thrips	Normal	138	193	103	434
		Guarded	96	117	80	293
		Moderate	101	37	43	181
		High	26	8	11	45
		Severe	4	10	3	17
	Whitefly	Normal	163	205	83	451
		Guarded	65	87	73	225
		Moderate	78	61	59	198
		High	52	9	24	85
		Severe	7	3	1	11



The pesticide usage on each installation site was also recorded. However, in the case of Farms TS1 and TS2, only the farm managers of Farm TS2 agreed in giving their detailed pesticide usage data that included the dosage, purpose, and mixture used. The reduction in the usage of pesticides of Farm TS2 was computed as shown in Table 4-7. The farm managers of Farm TS2 was only able to record their pesticide usage from the past 2 years. However, it already showed that there was already at least 16% reduction in the total dosage of pesticides used from Year 2 to Year 3. The farm managers said that this reduction was due to the more efficient pesticide spraying strategy they applied through the proposed system. It appeared that their frequency of pesticide usage was about the same but they were able to reduce their dosages since they had the quantitative data as reference. This shows that proper guidance was able to help the farmers in every decision-making for IPM. It was also important to collect as much information to assess the usefulness of the proposed system in different aspects of IPM.

Table 4-7. Pesticide usage data of Farm TS2.

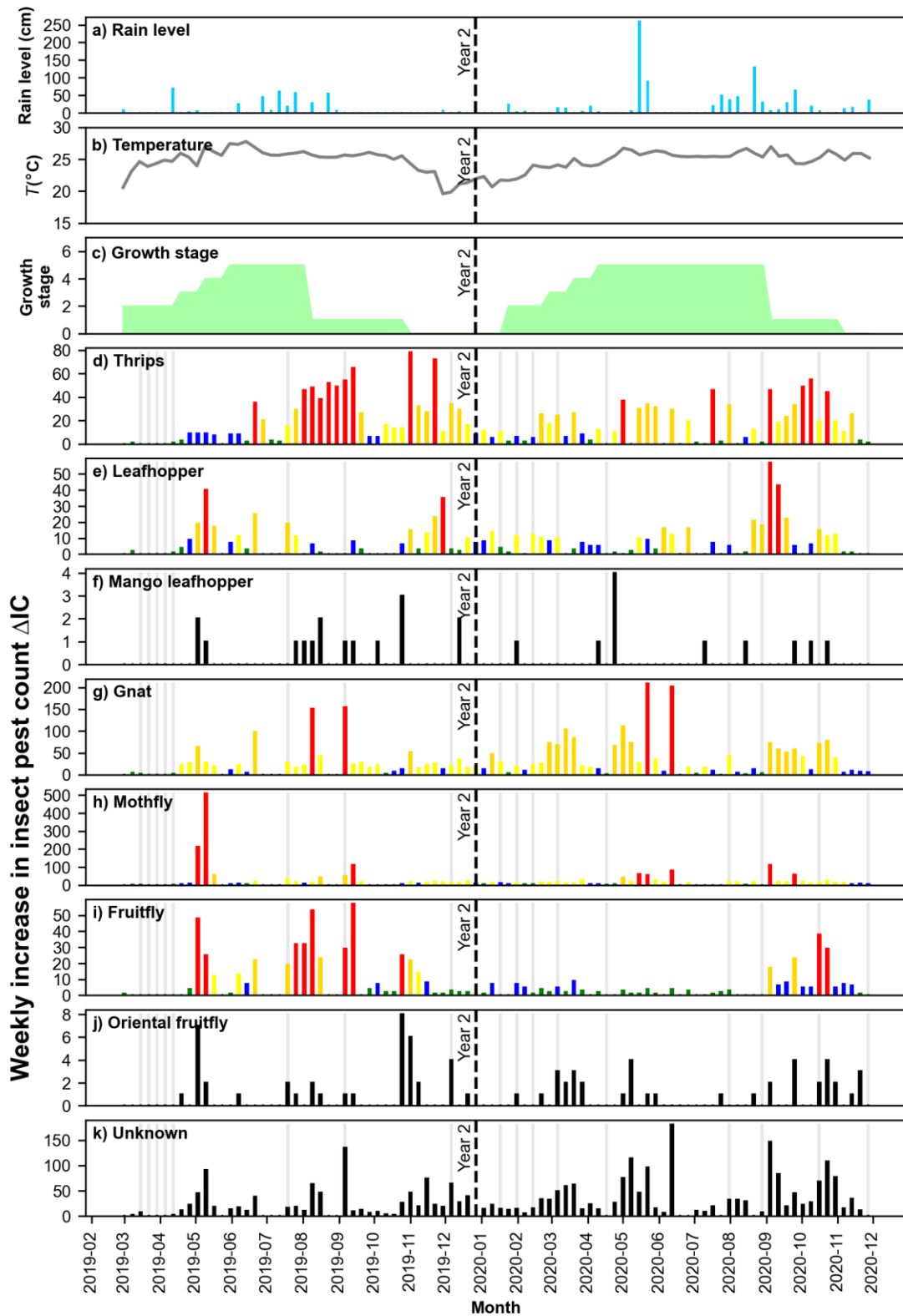
	Pesticide usage frequency	Total dosage (L)	% dosage reduction
Year 1	-	-	-
Year 2	55 times	12950	-
Year 3	53 times	10820	16%

4.4.2 Outdoor site data analysis



The analysis of the data obtained from Farm M1 is shown in Fig. 4-56. There was several important information found from the first-year data obtained from Farm M1. First, it was observed that thrips were mostly seen after the fruits fully ripened and not while they were ripening. Based on the IPM strategy of the farm managers, they said that they would usually spray pesticides continuously for several weeks before the fruits start to mature, as seen from 2019-03 to 2019-04. This will exterminate any harmful insect pests that will damage the crops as they mature. They also bagged the fruits while the fruits were ripening. After the fruits were harvested from 2019-07 to 2019-08, there was an outbreak of thrips from 2019-08 to 2019-09. This was matched by continuous rainfall which showed that only the growth stage affected the activity of thrips. The data verified that their IPM strategy worked as expected. It was also seen that a few leafhoppers, mango leafhoppers and oriental fruitflies were detected while the fruits were ripening. But as soon as the fruits were bagged, the activity of the insect pests dropped.

During the second year, the farm managers slightly changed their IPM strategy. In the beginning of that year, they sprayed pesticides every after each week. The data showed that it was still effective and there was no serious outbreak of the insect pests.



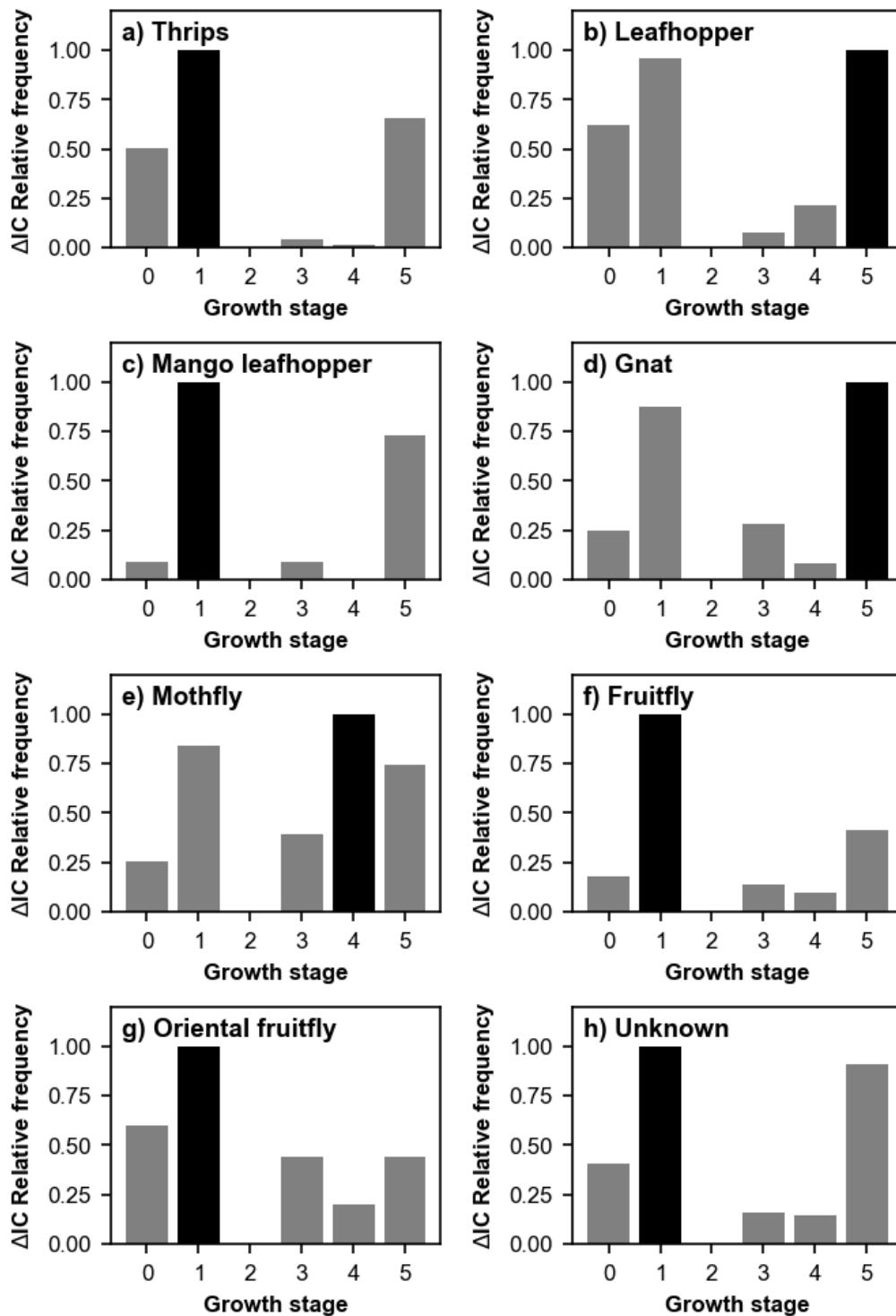
Growth stage: 0=None, 1=Pruned tips, 2=Flowering, 3=Early ripe, 4=Partially ripe, 5=Fully ripe

■ Normal ■ Guarded ■ Moderate ■ High ■ Severe ■ Pesticide

Fig. 4-56. Weekly insect pest count, growth stage, and rain level data of Farm M1.

Little activity was observed from the insect pests while the fruits were ripening. However, a similar phenomenon occurred in which there was an outbreak of thrips after the fruits were harvested (Fig. 4-56d). There was also continuous rainfall before the outbreak. But at the same time, leafhoppers appeared in 2020-09 after the fruits were harvested. Generally, for both years, the fruits were safe from insect pest outbreaks; avoiding potential economic loss.

The relative frequency of the insect in insect pest count for each growth stage was also analyzed, as shown in Fig. 4-57. The results in Fig. 4-57 show valuable information of when farmers can be more careful in handling the insect pests in their farm. For an instance, it was found that thrips were mostly appear after continuous rain. This means that it might potentially have been problematic if there was no continuous rainfall while the fruits were maturing. In that case, the farm managers should inspect the bagged fruits for possible damage. Fortunately, the data shows that thrips mostly showed up after the tree tips were pruned. Pruning the tips of fruit trees is a practice to stimulate better growth of new fruits. During this time, it was not critical to control the population of thrips since there were no fruits the thrips can damage. The results also showed that there was a considerable number of leafhoppers and mango leafhoppers as the fruits fully ripened. Mango leafhoppers typically feed on plant leaves and cause leaf curling. But based on the data, the farm managers were also able to prevent the damage caused

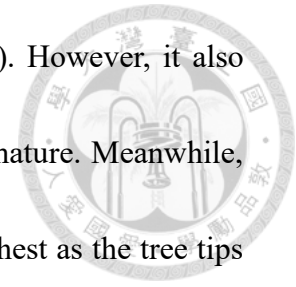


Growth stage: 0=None, 1=Pruned tips, 2=Flowering, 3=Early ripe, 4=Partially ripe, 5=Fully ripe

Fig. 4-57. Relative frequencies of increase in insect pest count per class during each

growth stage of the crops in Farm M1.

by mango leafhoppers during flowering (Peng & Christian, 2005). However, it also showed that the fruits should be instantly harvested as they fully mature. Meanwhile, the relative frequency of the detection of oriental fruitflies was highest as the tree tips were pruned. But there were also some instances that oriental fruitflies appeared as the fruits ripened.



The results showed that the system was able to successfully monitor the population of the insect pests in an outdoor agricultural production site. The data and information obtained from the system showed that growing mango trees require a certain amount of experience to develop a good IPM strategy. This reflects the relevance of the monitoring system such that new farm managers can be guided accordingly and know what kind of actions they can perform during certain situations. The system can act as a reference to verify if the IPM strategy of the farm managers were really effective or not.

It was also found that the rain level was an important factor to consider in monitoring the insect pests in outdoor sites and shall also be automatically recorded in the future. The current monitoring system can still be improved by incorporating the growth stage information in building predictive models that can assist the farmers in decision-making. This can be done by asking the farm managers to input the growth stage and activities done in the farm as additional information.



4.5 User feedback

With permission from the users, their login activity was recorded in the server database to know if they were using the system, as shown in Fig. 4-58. It can be easily noticed in Fig 4-58 that some of the users actively used the APP and monitored the insect pest condition of their farm. Particularly, the farm managers of TS1 had the most logins since the day that the farm managers downloaded the APP, accounting 45.26% of the time. Through their constant usage of the system, it was found that they were able to schedule their pesticide applications more efficiently as shown previously in the data analysis in Fig. 4-52.

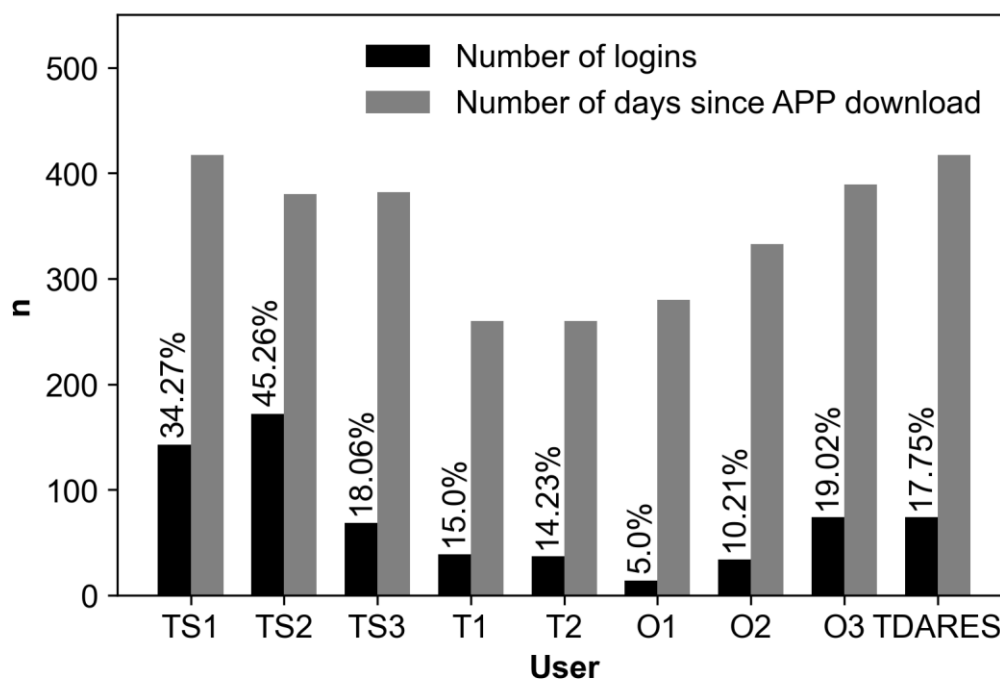
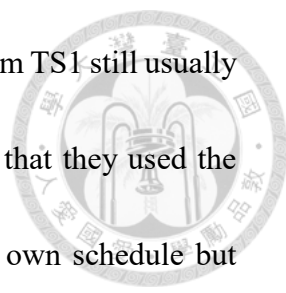
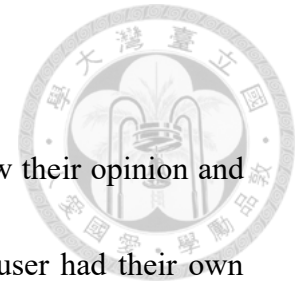


Fig. 4-58. I²PDM APP user login activity data.



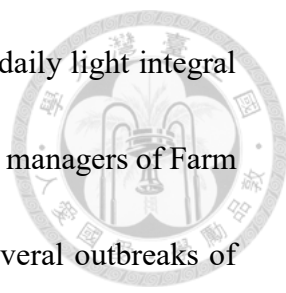
It was previously seen in Fig. 4-53 that the farm managers of Farm TS1 still usually sprayed pesticides based on their schedule. Nevertheless, it shows that they used the APP frequently. Based on their feedback, they still followed their own schedule but would refer to the APP for further action. They used the system as a reference so that they can respond if their IPM strategy was not efficient enough. This proves that the system was still able to help them by providing quantitative information that they can guide them in many ways. The farm managers of Farm TS3 used the APP less frequently. However, this was only in the beginning in which the system was still not fully functioning due to issues encountered in the site. But as soon as the system worked well, they used the APP more often. The farm managers of Farm T1 and T2 were some of the most recent users of the system. They did not frequently check the APP whenever there were no crops in the farm. The tomatoes in the farm were harvested from time to time and there was no need to use the APP. Meanwhile, the farm managers of the orchid farms used the APP only whenever the crops were still small. This was because it was critical to check if the seedlings grew up properly in the beginning. Lastly, our partner group, TDARES, usually checked the APP whenever they will visit the farms. They access the APP if they need to explain the insect pest condition to the farmers. This showed that the APP was used by both the farm managers and IPM experts. The login information gave us ideas on how to improve the APP and convince the users to access

the APP more frequently.



The feedback of each user of the system was obtained to know their opinion and recommendations to improve the system. It was found that each user had their own focus on how to make the system more useful. First, the farm managers of TS1 were more interested in adding more functions to the system; making it smarter. For an instance, one of their suggestion was to add a surveillance camera to track their activity. They also suggested that a portable sensor node will be also useful for other farm owners. The farm managers of TS2 also had the same opinion. Meanwhile, the farm managers of TS3 focused more on the stability of the sensor nodes. In that site, there were quite a number of lost data due to failing sensor nodes. From our experiences in managing the sensor nodes in that site, one of the problems was that the internet signal there was very weak using our own 4G sim card; therefore, making it unstable at first. However, the main issue was that the farm managers of TS3 used denser sprinklers that occasionally causes water to seep into the device. Thus, improvement in the hardware should be considered.

Users of the system from the tomato farms T1 and T2 had more positive feedback on the system. They said that the system was useful in properly tracking the number of insect pests in their farm. This allowed them to inspect their crops for diseases before any outbreak occurred. Meanwhile, the farm managers of O1 and O2 had similar



opinions. They were looking for other useful functions such as the daily light integral data and degree day data, which were recently added in the APP. The managers of Farm O2 was impressed of how our system found out that there were several outbreaks of gnats when there was a time they moved plants from greenhouse to greenhouse. However, their data showed that their current IPM strategy was already very effective because there were very few insect pests found in their farm. This was reasonable since their crops require strictly controlled environmental conditions to grow.

Lastly, our partner group, TDARES, also shared their opinion of the system. Their main concern was how the number of insects trapped by the sensor nodes was different from the number of insects found on their manually prepared cylindrical sticky paper traps. Unfortunately, as constrained by the current technology, it is very costly and inconvenient to take the image of a cylindrical sticky paper trap. One solution to this is to use an unmanned aerial vehicle. This researched used wireless sensor nodes, which were limited to taking images of flat sticky paper traps. Nevertheless, it was shown in Section 4.2.2 that the system was still able to trap as many insect pests to match the trend of the cylindrical traps.

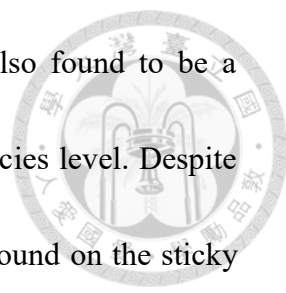
Chapter 5 Conclusions and Recommendations



5.1 Conclusions

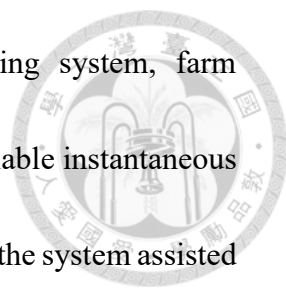
An intelligent and integrated system for IPM application was successfully developed in this research. Techniques in deep learning, machine learning, and data analytics were applied to promote smart and data-driven IPM to farmers. The system was able to collect and provide insect pest information that were proven to be useful for implementing IPM in different agricultural production sites.

The deep learning approaches applied for the two types of installation sites were successfully implemented. Based on the results obtained, the semi-supervised learning method applied for indoor sites was used as a convenient and reliable tool for collecting new training images and continuously retraining an adaptive multi-class insect classifier model. Based on the testing results, the final model presented in this research can achieve F_1 -scores of 0.93 based on both object level and image level testing. This was found to be a stepping stone towards developing a fully automated system that can learn from newly acquired images with minimal human supervision. The spatio-temporal voting method was also able to further improve the over-all algorithm performance.



The hierarchical deep learning classification approach was also found to be a plausible solution for identifying outdoor insect pests up to the species level. Despite the effect of many external factors and the variety of insect pests found on the sticky paper trap images obtained from the outdoor sites, the algorithm had average F_1 -scores of 0.91 and 0.89 by object level and image level testing, respectively. This was considered an accomplishment since most technologies were limited to counting single type of insect pests using specific types of sensors. In this research, it was proven that current imaging technology was not too far from identifying insect pests using wireless sensor nodes.

The system was installed in several agricultural production sites and valuable information were discovered from long-term monitoring. The data analyses showed that each agricultural production site had its own unique environment and insect pest condition. It was found that insect pest outbreaks were still possible in any site even the environmental conditions were carefully monitored and controlled. Several factors were found to affect the activity of insect pests. First, it was found from developed biological models that insect pests were most active in specific environmental conditions. For an instance, thrips and whiteflies emerged during warm weather and were less active when it was cold. This behavior of the insect pests was common for all the installation sites. It was also found that insect pest outbreaks occurred continuously



unless immediate intervention was done. Without the monitoring system, farm managers may suddenly face economic loss since they have no available instantaneous information regarding the insect pests in their farm. It was found that the system assisted them in properly timing their pesticide application to end insect pest outbreaks effectively. As a matter of fact, farm managers of two of the installation sites were able to reduce their yearly total number of insect pests through constant usage of the system.

The main issue found with the system was its maintenance. Limited by current technology, there were still problems that cannot be easily solved. One of the issues found was the replacement of sticky paper traps. Some farm managers forgot to replace the sticky paper traps when its already full of insects. This caused inaccuracy in detection and recognition due to overlapping insect bodies that cannot be resolved by using standard RGB cameras. The resolution of current cameras also limits the system to classifying certain insect types only up to order or family taxonomic level. Classification up to the species level was tested in this research with satisfactory results. However, it can still be improved if images with higher resolution were used for image recognition. Another issue was the maintenance of the sensor nodes. The current hardware design was limited by certain factors such as the size of commercially-available sticky paper traps and over-all cost. In order to promote the usage of the system, the cost of the materials and components used for building each sensor node

had to be considered. Despite the efforts in building a robust design, the harsh environment in the farms still damage the device to certain extents.



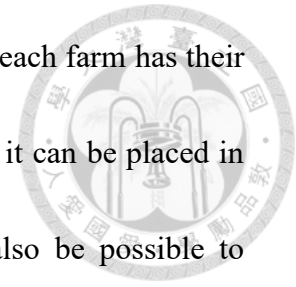
Based on the findings in this research, the proposed system was considered capable of bringing data-driven IPM to farmers, researchers, and agriculturists. Artificial intelligence was also proven to be capable of guiding and assisting people in their daily lives.

5.2 Recommendations

The presented system can still be improved in several ways. First, as recommended by the farmers, a surveillance system can be implemented to monitor either the farm staff activity or to detect plant disease infection from the crops. This may also help the system to automatically record the pesticide schedule of the farm managers. For an instance, the area in which pesticides were sprayed can be monitored to ensure that the population of the insect pests on detected hotspots were controlled efficiently. The additional function may also help in notifying the farmers if their crops were infected with a disease and determine whether it was caused by the insect pests or not.

The designed sensor node can also be optimized by making it self-powered. The common problem in the installation of the system was the preparation of AC power

sockets. This was found to be very cumbersome and difficult since each farm has their unique structure and layouts. If the sensor node was self-powered, it can be placed in any location in the farm and can be used effortlessly. It may also be possible to implement the insect pest detection and recognition algorithm in the sensor node itself such as to save memory in the server.

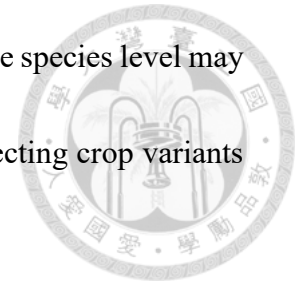


In terms of networking, a Wi-Fi mesh-based connection can also be implemented such that Wi-Fi extenders are no longer needed to keep the sensor nodes connected to the internet. The sensor nodes may connect to each other via Wi-Fi and form a larger network. This can make the system more flexible based on the layout and structure of the farms it will be installed in.

The semi-supervised learning method may also be redeveloped to adapt based on newly found insect pest image classes. The current limitation of the proposed method was that it needed priori knowledge of the insect pest classes to be trained on. Discovering new classes from the newly acquired insect pest images may make the system fully self-supervised. This can decrease the workload in managing the system.

By the time that imaging technology advances further, identification of the insect pests can be improved by classifying each insect according to its taxonomic species level. The current algorithm is now capable of classifying insect pests up to genus or

family level. However, there are instances that classification up to the species level may be necessary. Classification up to the species level may help in selecting crop variants that are immune to the damages dealt by specific insect pests.



There are still many possible ways to use the quantitative insect pest information obtained by the system. This may include insect pest outbreak forecasting, automated environmental control and pesticide application, and a lot more. Through the said functions, a smarter decision support system can be developed.

References



Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., . . . Zheng, X. (2016).

TensorFlow: A system for large-scale machine learning. In: Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation, Savannah, Georgia, USA.

Amorim, W. P., Tetila, E. C., Pistori, H., & Papa, J. P. (2019). Semi-supervised learning with convolutional neural networks for UAV images automatic recognition.

Computers and Electronics in Agriculture, 164, 104932.

<https://doi.org/10.1016/j.compag.2019.104932>

Atakan, E., & Canhilal, R. (2004). Evaluation of yellow sticky traps at various heights for monitoring cotton insect pests. Journal of Agricultural and Urban Entomology, 21, 15-24.

Barbedo, J. G. A. (2014). Using digital image processing for counting whiteflies on soybean leaves. Journal of Asia-Pacific Entomology, 17(4), 685-694.

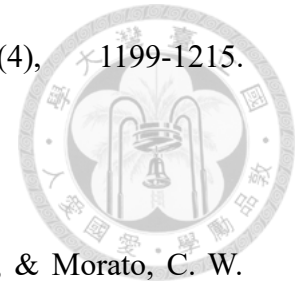
<https://doi.org/10.1016/j.aspen.2014.06.014>

Barzman, M., Bàrberi, P., Birch, A. N. E., Boonekamp, P., Dachbrodt-Saaydeh, S., Graf,

B., . . . Sattin, M. (2015). Eight principles of integrated pest management.

Agronomy for Sustainable Development, 35(4), 1199-1215.

<https://doi.org/10.1007/s13593-015-0327-9>



Baucum, M., Belotto, D., Jeannet, S., Savage, E., Mupparaju, P., & Morato, C. W.

(2017). Semi-supervised deep continuous learning. In: Proceedings of the 2017 International Conference on Deep Learning Technologies, Chengdu, China.

<https://doi.org/10.1145/3094243.3094247>

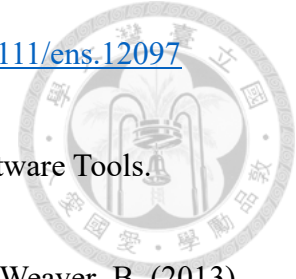
Bendale, A., & Boulton, T. E. (2016). Towards open set deep networks. In: Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, Nevada, United States.

Böckmann, E., Hommes, M., & Meyhöfer, R. (2015). Yellow traps reloaded: What is the benefit for decision making in practice? *Journal of Pest Science*, 88(2), 439-449. <https://doi.org/10.1007/s10340-014-0601-7>

Böckmann, E., & Meyhöfer, R. (2017). Sticky trap monitoring of a pest–predator system in glasshouse tomato crops: Are available trap colours sufficient? *Journal of Applied Entomology*, 141(5), 339-351. <https://doi.org/10.1111/jen.12338>

Bonsignore, C. P. (2015). Effect of environmental factors on the flight activity of *Trialeurodes vaporariorum* (Westwood) under greenhouse conditions.

Entomological Science, 18(2), 207-216. <https://doi.org/10.1111/ens.12097>



Bradski, G. (2000). The OpenCV library. Dr. Bobb's Journal of Software Tools.

Burr, T., Hamada, M. S., Howell, J., Skurikhin, M., Ticknor, L., & Weaver, B. (2013).

Estimating alarm thresholds for process monitoring data under different assumptions about the data generating mechanism. Science and Technology of Nuclear Installations, 2013, 705878. <https://doi.org/10.1155/2013/705878>

Butler, G. D., Rimon, D., & Henneberry, T. J. (1988). Bemisia tabaci (Homoptera: Aleyrodidae): Populations on different cotton varieties and cotton stickiness in Israel. Crop Protection, 7(1), 43-47. [https://doi.org/10.1016/0261-2194\(88\)90037-3](https://doi.org/10.1016/0261-2194(88)90037-3)

Cai, D., Chen, K., Qian, Y., & Kämäräinen, J.-K. (2019a). Convolutional low-resolution fine-grained classification. Pattern Recognition Letters, 119, 166-171. <https://doi.org/10.1016/j.patrec.2017.10.020>

Cai, J., Xiao, D., Lv, L., & Ye, Y. (2019b). An early warning model for vegetable pests based on multidimensional data. Computers and Electronics in Agriculture, 156, 217-226. <https://doi.org/10.1016/j.compag.2018.11.019>

Canziani, A., Paszke, A., & Cukurciello, E. (2016). An analysis of deep neural network models for practical applications. arXiv, abs/1605.07678.

Chen, L., Bentley, P., Mori, K., Misawa, K., Fujiwara, M., & Rueckert, D. (2019). Self-supervised learning for medical image analysis using image context restoration.

Medical Image Analysis, 58, 101539.

<https://doi.org/10.1016/j.media.2019.101539>

Chevalier, M., Thome, N., Cord, M., Fournier, J., Henaff, G., & Dusch, E. (2016). Low resolution convolutional neural network for automatic target recognition. In: 7th International Symposium on Optronics in Defence and Security, Paris, France.

<https://hal.archives-ouvertes.fr/hal-01332061>

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 greenhouse based on image analysis. International Journal of Mathematics and Computers in Simulation, 1, 46-53.

Chu, C.-C., Jackson, C. G., Alexander, P. J., Karut, K., & Henneberry, T. J. (2003).

Plastic cup traps equipped with light-emitting diodes for monitoring adult Bemisia tabaci (Homoptera: Aleyrodidae). Journal of Economic Entomology,

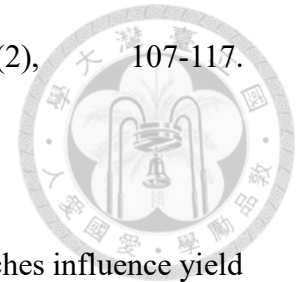
96(3), 543-546. <https://doi.org/10.1603/0022-0493-96.3.543>

Cohen, Y., Cohen, A., Hetzroni, A., Alchanatis, V., Broday, D., Gazit, Y., & Timar, D.

(2008). Spatial decision support system for Medfly control in citrus. Computers

and Electronics in Agriculture, 62(2), 107-117.

<https://doi.org/10.1016/j.compag.2007.12.005>



Csizinszky, A. A., Schuster, D. J., & Kring, J. B. (1995). Color mulches influence yield and insect pest populations in tomatoes. *Journal of the American Society for Horticultural Science*, 120(5), 778-784.

<https://doi.org/10.21273/JASHS.120.5.778>

Damos, P., & Savopoulou-Soultani, M. (2012). Temperature-driven models for insect development and vital thermal requirements. *Psyche*, 2012, 13.

<https://doi.org/10.1155/2012/123405>

Ding, W., & Taylor, G. (2016). Automatic moth detection from trap images for pest management. *Computers and Electronics in Agriculture*, 123(Supplement C),

17-28. <https://doi.org/10.1016/j.compag.2016.02.003>

Donatelli, M., Magarey, R. D., Bregaglio, S., Willocquet, L., Whish, J. P. M., & Savary,

S. (2017). Modelling the impacts of pests and diseases on agricultural systems.

Agricultural Systems, 155, 213-224. <https://doi.org/10.1016/j.agsy.2017.01.019>

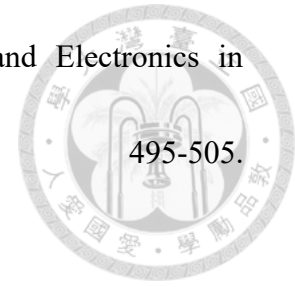
Espinoza, K., Valera, D. L., Torres, J. A., López, A., & Molina-Aiz, F. D. (2016).

Combination of image processing and artificial neural networks as a novel

approach for the identification of *Bemisia tabaci* and *Frankliniella occidentalis*

on sticky traps in greenhouse agriculture. *Computers and Electronics in Agriculture*, 127(Supplement C), 495-505.

<https://doi.org/10.1016/j.compag.2016.07.008>



Feichtenhofer, C., Pinz, A., & Zisserman, A. (2017). Detect to track and track to detect.

CoRR, abs/1710.03958.

Ferguson, A. W., Skellern, M. P., Johnen, A., von Richthofen, J.-S., Watts, N. P.,

Bardsley, E., . . . Cook, S. M. (2016). The potential of decision support systems to improve risk assessment for pollen beetle management in winter oilseed rape.

Pest Management Science, 72(3), 609-617. <https://doi.org/10.1002/ps.4069>

Fuentes, A., Yoon, S., Kim, S. C., & Park, D. S. (2017). A robust deep-learning-based

detector for real-time tomato plant diseases and pests recognition. *Sensors*

(Basel, Switzerland), 17(9), 2022. <https://doi.org/10.3390/s17092022>

García-Fernández, F. J., Verleysen, M., Lee, J. A., & Diaz, I. (2013). Stability

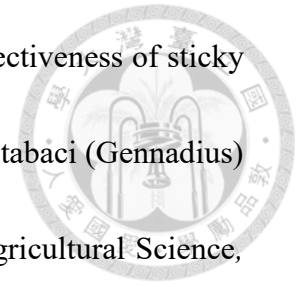
comparison of dimensionality reduction techniques attending to data and parameters variations. In: *Eurographics Conference on Visualization (EuroVis)*,

Leipzig, Germany.

Geng, C., Huang, S.-J., & Chen, S. (2018). Recent advances in open set recognition: A

survey. arXiv, arXiv:1811.08581.

Ghani, I., Khalid, S. A. N., Mohd. N., & Mohamad, R. (2012). Effectiveness of sticky trap designs and colours in trapping alate whitefly, *Bemisia tabaci* (Gennadius) (Hemiptera: Aleyrodidae). *Pertanika Journal of Tropical Agricultural Science*, 35, 127-134.



Guagliardo, S. A. J., Reynolds, M. G., Kabamba, J., Nguete, B., Shongo Lushima, R., Wemakoy, O. E., & McCollum, A. M. (2018). Sounding the alarm: Defining thresholds to trigger a public health response to monkeypox. *PLoS Neglected Tropical Diseases*, 12(12), e0007034.
<https://doi.org/10.1371/journal.pntd.0007034>

Guizilini, V., & Ramos, F. (2015). Online self-supervised learning for dynamic object segmentation. *The International Journal of Robotics Research*, 34(4-5), 559-581.
<https://doi.org/10.1177/0278364914566514>

Haefner, J. W. (2005). *Modeling Biological Systems: Principles and Applications* (2nd ed.). New York, USA: Springer.

Haridas, C. V., Meinke, L. J., Hibbard, B. E., Siegfried, B. D., & Tenhumberg, B. (2016). Effects of temporal variation in temperature and density dependence on insect population dynamics. *Ecosphere*, 7(5). <https://doi.org/10.1002/ecs2.1287>

Hazarika, L., Bhuyan, M., & Hazarika, B. (2009). Insect pests of tea and their

management. *Annual Review of Entomology*, 54, 267-284.

<https://doi.org/10.1146/annurev.ento.53.103106.093359>



Hong, S.-J., Kim, S.-Y., Kim, E., Lee, C.-H., Lee, J.-S., Lee, D.-S., . . . Kim, G. (2020).

Moth detection from pheromone trap images using deep learning object detectors. *Agriculture*, 10, 170. <https://doi.org/10.3390/agriculture10050170>

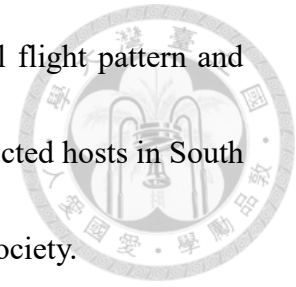
Huang, B., Reichman, D., Collins, L. M., Bradbury, K., Malof, J. M. (2019). Tiling and stitching segmentation output for remote sensing: Basic challenges and recommendations. *arXiv*, arXiv:1805.12219.

Iqbal, Z., Islam, N., Jang, B.-E., Ali, M., Kabir, S. N., Lee, D.-H., . . . Chung, S.-O. (2019). Monitoring the operating status of an automatic harmful fly collector for smart greenhouses. *Journal of Biosystems Engineering*, 44(4), 258-268. <https://doi.org/10.1007/s42853-019-00036-8>

Israni, D., & Mewada, H. (2018). Identity retention of multiple objects under extreme occlusion scenarios using feature descriptors. *Journal of Communications Software and Systems*, 14, 290-301.

Jain, L. P., Scheirer, W. J., & Boulton, T. E. (2014). Multi-class open set recognition using probability of inclusion. In: *European Conference on Computer Vision*, Zurich, Switzerland.

Jha, V. K., Seal, D. R., Schuster, D. J., & Kakkar, G. (2009). Diel flight pattern and periodicity of chilli thrips (Thysanoptera: Thripidae) on selected hosts in South Florida. In: Proceedings of the Florida State Horticultural Society.



Jones, A., Ali, U., & Egerstedt, M. (2016). Optimal pesticide scheduling in precision agriculture. In: 2016 ACM/IEEE 7th International Conference on Cyber-Physical Systems (ICCPS).

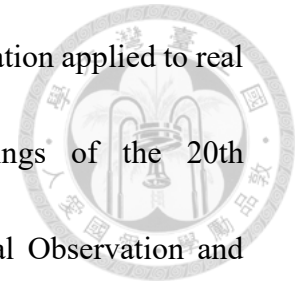
Jones, D. R. (2005). Plant viruses transmitted by thrips. *European Journal of Plant Pathology*, 113(2), 119-157. <https://doi.org/10.1007/s10658-005-2334-1>

Kaas, J. (2005). Vertical distribution of thrips and whitefly in greenhouses and relative efficiency of commercially available sticky traps for population monitoring. In: Netherlands Entomological Society Meeting.

Kingsolver, J. G., Higgins, J. K., & Augustine, K. E. (2015). Fluctuating temperatures and ectotherm growth: distinguishing non-linear and time-dependent effects. *The Journal of Experimental Biology*, 218(14), 2218-2225. <https://doi.org/10.1242/jeb.120733>

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84-90. <https://doi.org/10.1145/3065386>

Kumar, R., Martin, V., & Moisan, S. (2010). Robust insect classification applied to real Time greenhouse infestation monitoring. In: Proceedings of the 20th International Conference on Pattern Recognition on Visual Observation and Analysis of Animal and Insect Behavior Workshop, Istanbul, Turkey.



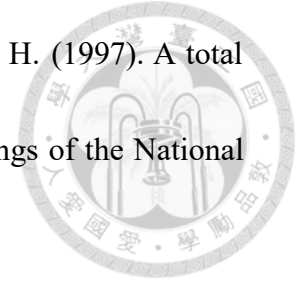
Kwasnicka, H., & Paradowski, M. (2010). Machine Learning Methods in Automatic Image Annotation (Vol. 263, pp. 387-411).

Lamichhane, J. R., Aubertot, J.-N., Begg, G., Birch, A. N. E., Boonekamp, P., Dachbrodt-Saaydeh, S., . . . Messéan, A. (2016). Networking of integrated pest management: A powerful approach to address common challenges in agriculture. *Crop Protection*, 89, 139-151. <https://doi.org/10.1016/j.cropro.2016.07.011>

Larson, R., & Falvo, D. C. (2011). *Linear Algebra*. Massachusetts, USA: Cengage Learning.

Lee, C.-H., Liu, C.-F., Lin, Y.-T., Yain, Y.-S., & Lin, C.-H. (2020). New agriculture business model in Taiwan. *International Food and Agribusiness Management Review*, 23(5), 773-782. <https://doi.org/10.22434/ifamr2019.0164>

Lee, D.-H. (2013). Pseudo-Label: The simple and efficient semi-supervised learning method for deep neural networks. In: ICML 2013 Workshop: Challenges in Representation Learning (WREPL).



Lewis, W. J., van Lenteren, J. C., Phatak, S. C., & Tumlinson, J. H. (1997). A total system approach to sustainable pest management. *Proceedings of the National Academy of Sciences*, 94, 12243-12248.

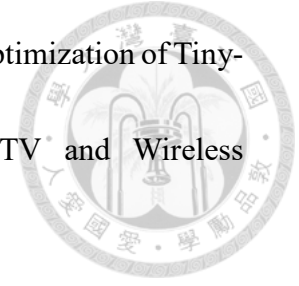
Liang, X.-H., Lei, Z. R., Wen, J.-Z., & Zhu, M.-L. (2010). The diurnal flight activity and influential factors of *Frankliniella occidentalis* in the greenhouse. *Insect Science*, 17(6), 535-541. <https://doi.org/10.1111/j.1744-7917.2010.01337.x>

Liu, F. T., Ting, K. M., & Zhou, Z. (2008). Isolation forest. In: 2008 Eighth IEEE International Conference on Data Mining.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S. E., Fu, C.-Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. arXiv, abs/1512.02325.

López Granado, O., Martínez-Rach, M., Migallón, H., Malumbres, M., Bonastre Pina, A., & Serrano Martín, J. (2012). Monitoring pest insect traps by means of low-power image sensor technologies. *Sensors*, 12(11), 15801-15819. <https://doi.org/10.3390/s121115801>

Lu, C.-Y., Rustia, D. J. A., & Lin, T.-T. (2019). A greenhouse whitefly early warning method based on autoregression recurrent neural network. In: Asia-Pacific Federation for Information Technology in Agriculture Conference, Taichung, Taiwan.



Ma, J., Chen, L., & Gao, Z. (2018). Hardware implementation and optimization of Tiny-YOLO network. In: International Forum on Digital TV and Wireless Multimedia Communications, Singapore.

Martin, V., & Moisan, S. (2017). Early pest detection in greenhouses.

McKee, G. J. (2011). Coordinated pest management decisions in the presence of management externalities: The case of greenhouse whitefly in California-grown strawberries. *Agricultural Systems*, 104(1), 94-103.
<https://doi.org/10.1016/j.agsy.2010.10.005>

McKinney, W. (2010). Data structures for statistical computing in Python. In: Proceedings of the 9th Python in Science Conference.

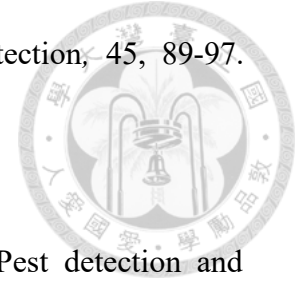
Miller, P., Lanier, W., & Brandt, S. (2018). Using Growing Degree Days to Predict Plant Stages. Retrieved from
<http://store.msuextension.org/publications/AgandNaturalResources/MT200103AG.pdf>

Milton, J. S., & Arnold, J. C. (2004). *Introduction to Probability and Statistics* (4th ed.). New York, USA: McGraw-Hill.

Miluch, C. E., Dossdall, L. M., & Evenden, M. L. (2013). The potential for pheromone-based monitoring to predict larval populations of diamondback moth, *Plutella*

xylostella (L.), in canola (*Brassica napus* L.). *Crop Protection*, 45, 89-97.

<https://doi.org/10.1016/j.cropro.2012.11.023>



Miranda, J. L., Gerardo, B. D., & Tanguilig, B. T. I. (2014). Pest detection and extraction using image processing techniques. *International Journal of Computer and Communication Engineering* 3, 189-192.

<https://doi.org/10.7763/IJCCE.2014.V3.317>

Miranda, M. Á., Barceló, C., Valdés, F., Feliu, J. F., Nestel, D., Papadopoulos, N., . . .

Alorda, B. (2019). Developing and implementation of decision support system (DSS) for the control of olive fruit fly, *Bactrocera Oleae*, in Mediterranean olive orchards. *Agronomy*, 9(10), 620. <https://doi.org/10.3390/agronomy9100620>

Mirnezhad, M., Romero-González, R. R., Leiss, K. A., Choi, Y. H., Verpoorte, R., & Klinkhamer, P. G. L. (2010). Metabolomic analysis of host plant resistance to thrips in wild and cultivated tomatoes. *Phytochemical Analysis*, 21(1), 110-117.

<https://doi.org/10.1002/pca.1182>

Moreau, T. L., & Isman, M. B. (2011). Trapping whiteflies? A comparison of greenhouse whitefly (*Trialeurodes vaporariorum*) responses to trap crops and yellow sticky traps. *Pest Management Science*, 67(4), 408-413.

<https://doi.org/10.1002/ps.2078>

Mundada, R. G., & Gohokar, V. V. (2013). Detection and classification of pests in greenhouse using image processing. *IOSR Journal of Electronics and Communication Engineering*, 5, 57-63. <https://doi.org/10.9790/2834-565763>



Murai, T., & Loomans, A. J. M. (2001). Evaluation of an improved method for mass-rearing of thrips and a thrips parasitoid. *Entomologia Experimentalis et Applicata*, 101(3), 281-289. <https://doi.org/10.1046/j.1570-7458.2001.00913.x>

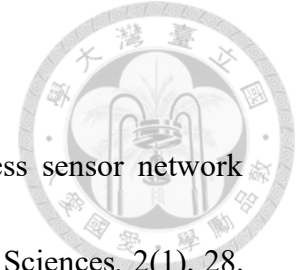
Navas-Castillo, J., Fiallo-Olive, E., & Sanchez-Campos, S. (2011). Emerging virus diseases transmitted by whiteflies. *Annual Review of Phytopathology*, 49, 219-248. <https://doi.org/10.1146/annurev-phyto-072910-095235>

Navasero, M., & Navasero, M. (2015). *Insect Pests of Tomato: Pests and Diseases of Economically Important Crops in the Philippines* (pp. 500-518): Department of Agriculture-Bureau of Agricultural Research.

Palacio-Niño, J.-O., & Galiano, F. B. (2019). Evaluation metrics for unsupervised learning algorithms. *arXiv*, arXiv:1905.05667.

Parsa, S., Morse, S., Bonifacio, A., Chancellor, T. C. B., Condori, B., Crespo-Pérez, V., . . . Dangles, O. (2014). Obstacles to integrated pest management adoption in developing countries. *Proceedings of the National Academy of Sciences of the United States of America*, 111(10), 3889-3894.

<https://doi.org/10.1073/pnas.1312693111>



Parsons, L., Ross, R., & Robert, K. (2019). A survey on wireless sensor network technologies in pest management applications. *SN Applied Sciences*, 2(1), 28.

<https://doi.org/10.1007/s42452-019-1834-0>

Partel, V., Nunes, L., Stansly, P., & Ampatzidis, Y. (2019). Automated vision-based system for monitoring Asian citrus psyllid in orchards utilizing artificial intelligence. *Computers and Electronics in Agriculture*, 162, 328-336.

<https://doi.org/10.1016/j.compag.2019.04.022>

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.

Pelgrom, K., Broekgaarden, C., Voorrips, R., Bas, N., Visser, R., & Vosman, B. (2014). Host plant resistance towards the cabbage whitefly in *Brassica oleracea* and its wild relatives. *Euphytica*, 202. <https://doi.org/10.1007/s10681-014-1306-y>

Peng, R. K., & Christian, K. (2005). The control efficacy of the weaver ant, *Oecophylla smaragdina* (Hymenoptera: Formicidae), on the mango leafhopper, *Idioscopus nitidulus* (Hemiptera: Cicadellidea) in mango orchards in the Northern Territory. *International Journal of Pest Management*, 51(4), 297-304.

<https://doi.org/10.1080/09670870500151689>

Pinto-Zevallos, D. M., & Vänninen, I. (2013). Yellow sticky traps for decision-making in whitefly management: What has been achieved? *Crop Protection*, 47, 74-84.

<https://doi.org/10.1016/j.cropro.2013.01.009>

Potamitis, I., Eliopoulos, P., & Rigakis, I. (2017). Automated remote insect surveillance at a global scale and the internet of things. *Robotics*, 6(3), 19.

<https://doi.org/10.3390/robotics6030019>

Pourdarbani, R., Sabzi, S., Hernández Hernández, M., Hernández Hernández, J. L., García-Mateos, G., Kalantari, D., & Martínez, J. (2019). Comparison of different classifiers and the majority voting rule for the detection of plum fruits in garden conditions. *Remote Sensing*, 11, 2546.

<https://doi.org/10.3390/rs11212546>

Prema, M. S., Ganapathy, N., Renukadevi, P., Mohankumar, S., & Kennedy, J. S. (2018). Coloured sticky traps to monitor thrips population in cotton. *Journal of Entomology and Zoology Studies*, 6(2).

Pundt, L. (2013). Tips on Scouting Poinsettia Insect and Mite Pests. Retrieved from

<http://ipm.uconn.edu/documents/view.php?id=658>

Qiao, M., Lim, J., Ji, C. W., Chung, B.-K., Kim, H.-Y., Uhm, K.-B., . . . Chon, T.-S.

(2008). Density estimation of *Bemisia tabaci* (Hemiptera: Aleyrodidae) in a greenhouse using sticky traps in conjunction with an image processing system.

Journal of Asia-Pacific Entomology, 11(1), 25-29.

<https://doi.org/10.1016/j.aspen.2008.03.002>

Qin, J.-L., Yang, X.-H., Yang, Z.-W., Luo, J.-T., & Lei, X.-F. (2017). New technology for using meteorological information in forest insect pest forecast and warning

systems. Pest Management Science, 73(12), 2509-2518.

<https://doi.org/10.1002/ps.4647>

Ramalho, F. S., Malaquias, J. B., Lira, A. C. S., Oliveira, F. Q., Zanuncio, J. C., &

Fernandes, F. S. (2015). Temperature-dependent fecundity and life table of the

Fennel Aphid *Hyadaphis foeniculi* (Passerini) (Hemiptera: Aphididae). PLoS

ONE, 10(4), e0122490. <https://doi.org/10.1371/journal.pone.0122490>

Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. arXiv,

abs/1804.02767.

Ren, S., He, K., Girshick, R. B., & Sun, J. (2015). Faster R-CNN: Towards real-time

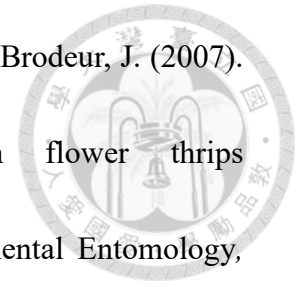
object detection with region proposal networks. IEEE Transactions on Pattern

Analysis and Machine Intelligence, 39, 1137-1149.

<https://doi.org/10.1109/TPAMI.2016.2577031>

Rhainds, M., Cloutier, C., Shipp, L., Boudreault, S., Daigle, G., & Brodeur, J. (2007).

Temperature-mediated relationship between western flower thrips
(Thysanoptera: Thripidae) and chrysanthemum. *Environmental Entomology*,
36(2), 475-483. <https://doi.org/10.1093/ee/36.2.475>



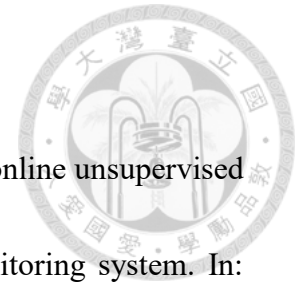
Rieger, D., Fraunholz, C., Popp, J., Bichler, D., Dittmann, R., & Helfrich-Forster, C.
(2007). The fruit fly *Drosophila melanogaster* favors dim light and times its
activity peaks to early dawn and late dusk. *Journal of Biological Rhythms*, 22,
387-399. <https://doi.org/10.1177/0748730407306198>

Rigamonti, I. E., Trivellone, V., Jermini, M., Fuog, D., & Baumgärtner, J. (2014).
Multiannual infestation patterns of grapevine plant inhabiting *Scaphoideus*
titanus (Hemiptera: Cicadellidae) leafhoppers. *The Canadian Entomologist*, 146,
67-79. <https://doi.org/10.4039/tce.2013.51>

Rossetto, C. J., Carvalho, R. L., & Walder, J. M. M. (2017). Mango resistance to fruit
fly: Method of evaluation, resistant cultivars and resistance factors. *Acta*
Horticulturae, 251-254. <https://doi.org/10.17660/ActaHortic.2017.1183.36>

Rustia, D. J. A., Chao, J.-J., Chiu, L.-Y., Wu, Y.-F., Chung, J.-Y., Hsu, J.-C., & Lin, T.-
T. (2020a). Automatic greenhouse insect pest detection and recognition based
on a cascaded deep learning classification method. *Journal of Applied*

Entomology, 1-17. <https://doi.org/10.1111/jen.12834>



Rustia, D. J. A., Chao, J.-J., Chung, J.-Y., & Lin, T.-T. (2019). An online unsupervised deep learning approach for an automated pest insect monitoring system. In: 2019 ASABE Annual International Meeting, Boston, Massachusetts.

<http://elibrary.asabe.org/abstract.asp?aid=50319&t=5>

Rustia, D. J. A., & Chung, W.-Y. (2016). A multi-functional remotely accessible monitoring and control system for optimized plant growth. In: IEEE International Conference on Applied System Innovation, Okinawa, Japan.

Rustia, D. J. A., Lin, C. E., Chung, J.-Y., & Lin, T.-T. (2017). An object classifier using support vector machines for real-time insect pest counting. In: Conference on Bio-Mechatronics and Agricultural Machinery Engineering, Taipei, Taiwan.

Rustia, D. J. A., Lin, C. E., Chung, J.-Y., & Lin, T.-T. (2018). A real-time multi-class insect pest identification method using cascaded convolutional neural networks. In: 9th International Symposium on Machinery and Mechatronics for Agriculture and Biosystems Engineering, Jeju, South Korea.

Rustia, D. J. A., Lin, C. E., Chung, J.-Y., Zhuang, Y.-J., Hsu, J.-C., & Lin, T.-T. (2020b). 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>



Rustia, D. J. A., & Lin, T.-T. (2019). Stochastic models for greenhouse whitefly flight behavior based on wireless image monitoring system measurements. *Pertanika Journal of Science and Technology*, 27(S1), 81-93.

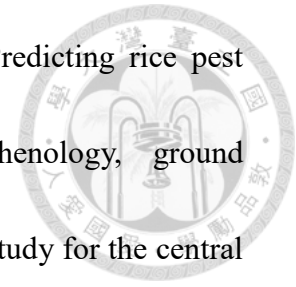
Sampson, C., Covaci, A. D., Hamilton, J. G. C., Hassan, N., Al-Zaidi, S., & Kirk, W. D. J. (2018). Reduced translucency and the addition of black patterns increase the catch of the greenhouse whitefly, *Trialeurodes vaporariorum*, on yellow sticky traps. *PLoS ONE*, 13(2), e0193064. <https://doi.org/10.1371/journal.pone.0193064>

Sengonca, C., & Liu, B. (1999). Laboratory studies on the effect of temperature and humidity on the life table of the whitefly *Aleurotuberculatus takahashi* David & Subramaniam (Hom., Aleyrodidae) from southeastern China. *Anzeiger für Schädlingkunde*, 72(2), 45. <https://doi.org/10.1007/bf02771095>

Shimoda, M., & Honda, K.-I. (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>

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv*, abs/1409.1556.

Skawsang, S., Nagai, M., Tripathi, N. K., & Soni, P. (2019). Predicting rice pest population occurrence with satellite-derived crop phenology, ground meteorological observation, and machine learning: A case study for the central plain of Thailand. *Applied Sciences*, 9, 4846.



<https://doi.org/10.3390/app9224846>

Stukenberg, N., Gebauer, K., & Poehling, H. M. (2015). Light emitting diode(LED)-based trapping of the greenhouse whitefly (*Trialeurodes vaporariorum*). *Journal of Applied Entomology*, 139(4), 268-279. <https://doi.org/10.1111/jen.12172>

Thimijan, R. W., & Heins, R. D. (1983). Photometric, radiometric, and quantum light units of measure: A review of procedures for interconversion. *HortScience*, 18(6).

Unel, F. O., Ozkalayci, B. O., & Çigla, C. (2019). The power of tiling for small object detection. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*, Longbeach, California, USA.

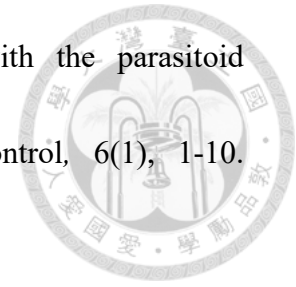
van Lenteren, J. C., & Martin, N. A. (1999). *Biological Control of Whiteflies: Integrated Pest and Disease Management in Greenhouse Crops* (pp. 202-216). Dordrecht: Springer Netherlands.

van Lenteren, J. C., van Roermund, H. J. W., & Sütterlin, S. (1996). *Biological control*

of greenhouse whitefly (*Trialeurodes vaporariorum*) with the parasitoid

Encarsia formosa: How does It work? *Biological Control*, 6(1), 1-10.

<https://doi.org/10.1006/bcon.1996.0001>



Veit, A., Alldrin, N., Chechik, G., Krasin, I., Gupta, A., & Belongie, S. (2017). Learning from noisy large-scale datasets with minimal supervision. In: Conference on Computer Vision and Pattern Recognition, Honolulu, Hawaii, USA.

Wang, X., Levy, K., Son, Y., Johnson, M., & Daane, K. (2012). Comparison of the thermal performance between a population of the olive fruit fly and its co-adapted parasitoids. *Biological Control*, 60, 247-254.

<https://doi.org/10.1016/j.biocontrol.2011.11.012>

Wen, C., & Guyer, D. (2012). Image-based orchard insect automated identification and classification method. *Computers and Electronics in Agriculture*, 89, 110-115.

<https://doi.org/10.1016/j.compag.2012.08.008>

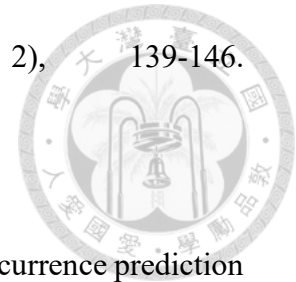
Witzgall, P., Kirsch, P., & Cork, A. (2010). Sex pheromones and their impact on pest management. *Journal of Chemical Ecology*, 36(1), 80-100.

<https://doi.org/10.1007/s10886-009-9737-y>

Xia, C., Chon, T.-S., Ren, Z., & Lee, J.-M. (2015). Automatic identification and counting of small size pests in greenhouse conditions with low computational

cost. Ecological Informatics, 29(Part 2), 139-146.

<https://doi.org/10.1016/j.ecoinf.2014.09.006>



Xiao, Q., Li, W., Kai, Y., Chen, P., Zhang, J., & Wang, B. (2019). Occurrence prediction of pests and diseases in cotton on the basis of weather factors by long short term memory network. BMC Bioinformatics, 20(25), 688.

<https://doi.org/10.1186/s12859-019-3262-y>

Yee, W. L. (2013). Soil moisture and relative humidity effects during postdiapause on the emergence of western cherry fruit fly (Diptera: Tephritidae). The Canadian Entomologist, 145(3), 317-326. <https://doi.org/10.4039/tce.2013.7>

Yi, Z., Yongliang, S., & Jun, Z. (2019). An improved Tiny-YOLOv3 pedestrian detection algorithm. Optik, 183. <https://doi.org/10.1016/j.ijleo.2019.02.038>

Yin, L. T., & Maschwitz, U. (1983). Sexual pheromone in the green house whitefly *Trialeurodes vaporariorum* Westw. Zeitschrift für Angewandte Entomologie, 95(1-5), 439-446. <https://doi.org/10.1111/j.1439-0418.1983.tb02665.x>

Zhang, C., Cai, J., Xiao, D., Ye, Y., & Chehelamirani, M. (2018). Research on vegetable pest warning system based on multidimensional big data. Insects, 9(2), 66. <https://doi.org/10.3390/insects9020066>

Zhong, Y., Gao, J., Lei, Q., & Zhou, Y. (2018). A vision-based counting and recognition

system for flying insects in intelligent agriculture. *Sensors* (Basel, Switzerland),

18(5), 1489. <https://doi.org/10.3390/s18051489>



Zong, B., Song, Q., Min, M. R., Cheng, W., Lumezanu, C., Cho, D.-K., Chen, H. (2018).

Deep autoencoding Gaussian mixture model for unsupervised anomaly

detection. In: *International Conference on Learning Representations (ICLR)*,

Vancouver, Canada.