國立臺灣大學生物資源暨農學院生物機電工程學系

碩士論文

Department of Biomechatronics Engineering
College of Bioresources and Agriculture
National Taiwan University

Master's Thesis

應用影像度用與機器學習於智慧害蟲監測系統資料品質之提升

Enhancing Data Quality in Intelligent Insect Pest Monitoring System with Image Curation and Machine Learning

鄧喬尹

Chiao-Yin Teng

指導教授：林達德 博士

Advisor: Ta-Te Lin, Ph.D.

中華民國 113 年 07 月

July 2024

doi:10.6342/NTU202404155

誌謝

在本論文即將完成之際，我衷心感謝所有在碩士班兩年生涯中給予我支持和幫助的人們。

首先是我的口試委員林達德老師、陳世芳老師和張仲良老師，感謝您們撥冗前來參加我的論文口試，分享專業知識和實務經驗，讓我的論文更加完善。特別感謝我的指導老師：林達德教授。在整個碩士班的研究過程中，不論是學術研究上提供指導與知識傳授，帶領著我一步一步完成論文的撰寫，以及在待人處事方面，學習如何跟政府部門與企業單位合作完成計畫。當我對研究或是人生方向感到迷茫時，您總是用過去的經驗為我指明方向，使我在面對挑戰時能夠堅持不懈，順利完成論文的寫作。

接下來我要感謝 405 實驗室的成員。首先是我的碩士班同學晨宇和易霖，從修課、出差到出國參加研討會，我們在碩士班的期間相互扶持，一起成長，豐富這兩年的碩士班生活。再來是實驗室的其他夥伴們，有你們的協助，我才能順利的完成政府計畫、產學合作和實驗研究。感謝大家在這段時間互相幫助，祝福各位在未來的道路上一帆風順。

特別感謝我的家人。感謝我的父母從小到大給予我很大的自由，讓我能做自己想做的事情，並在一旁默默的支持我。感謝在天上的阿公和外公從小對我的疼愛，請您們在天上繼續看著我，我不會讓你們失望的。最後，我要感謝我的女友譽瑄。從高中就一起共同努力、相互扶持，妳是我向前進的最大動力。沒有親人的照顧和支持，我不可能堅持到今天，在此獻上我最誠摯的感謝。

要感謝的人實在太多，無法一一列舉，但我想對自己說，感謝那個不輕言放棄的自己，願愛我的人和我愛的人都能平安健康。

doi:10.6342/NTU202404155

中文摘要

在農作物生產中，蟲害被認為是對農業生產的最大威脅之一。它們危害農作物的生長，降低作物產量，對農業經濟收益造成重大損失。精確獲取作物栽培環境中害蟲種類與數量的資訊對於蟲害整合管理（IPM）至關重要。在之前的研究中，我們開發出一套基於深度學習方法的 AIoT 影像設備，專門用於通過黏蟲紙獲取的影像，利用深度學習辨識害蟲種類與數量。儘管先前已取得初步研究成果，但在灰塵和水分干擾、光照條件變化及影像模糊等挑戰下，仍可能導致害蟲的錯誤分類和數量計算不準確。為確保蟲害整合管理的有效性，本研究的目標是開發一個整體框架，用於提升害蟲檢測、分類和數據分析。我們採用 YOLOv7 模型來替代之前的 YOLOv3-Tiny 模型，專門用於黏蟲紙影像的物體檢測。結果顯示，YOLOv7 模型在各個實驗場域中均達到 0.95 以上的 mAP@.5 表現。後續使用 ResNet-18 模型進行害蟲分類，達到整體害蟲分類模型的 F1-score 為 0.988。此外，利用時空間分析對一段時間內蒐集的黏蟲紙影像進行害蟲種類和數量的調整。創立一個動態的二維數組來記錄在黏蟲紙影像中檢測到害蟲的位置和時間。通過多數決投票算法與對應座標的時空間分析，修正每個時間點害蟲的種類與數量，從而最大限度地減少害蟲的錯誤分類和數量計算不準確。相比於人工統計的真實數據，MAPE 誤差僅約為 5%。分析結果顯示，本研究所提出的框架能有效提升害蟲辨識模型效能和整體資料品質。

關鍵詞: 病蟲害整合管理、影像前處理、物件偵測模型、害蟲分類模型、時空間分佈演算法分析

doi:10.6342/NTU202404155

英文摘要

Effective integrated pest management (IPM) in crop production depends on precise information about the quantity and species of insect pests in the crop cultivation environment. In our prior work, we developed AIoT imaging devices employing a deep learning approach tailored for the automated counting and classification of insect pests through acquired images from sticky paper traps. Notwithstanding the previous progress achieved, challenges persist in mitigating issues such as interference from dust and water, variations in lighting conditions, and blurred images, ultimately resulting in the misclassification and miscounting of insect pests. To ensure the effectiveness of integrated pest management, securing accurate data on the types and quantities of insect pests present within the field is crucial. Thus, the objective of this research aims to develop a holistic framework for enhancement of insect pest detection, classification, and data analysis. This involves adopting the YOLOv7 model for object detection models specifically applied to images obtained from sticky paper traps, as a replacement for the prior approach employing the YOLOv3-tiny model. The results demonstrated that the proposed method achieved an average mAP exceeding 0.95, and employing ResNet-18 for insect pest classification, achieving an overall F1-score of 0.988. Additionally, subsequent method involves insect pest count modification using spatiotemporal analysis for a series of sticky paper trap images. A dynamic two-dimensional array was created to log the location and times at which insect pests were detected on the sticky paper trap images. Employing a voting algorithm can ascertain the definitive count of insect pests for each time interval, thereby minimizing the potential for miscounts, achieved a MAPE error of only about 5% compared to ground

doi:10.6342/NTU202404155

truth. The analysis results indicate that the proposed framework effectively enhances the insect pest recognition performance and the overall data quality.

Keywords: integrated pest management, image curation, object detection, insect pest classification, spatiotemporal analysis

doi:10.6342/NTU202404155

Table of Contents

doi:10.6342/NTU202404155

doi:10.6342/NTU202404155

doi:10.6342/NTU202404155

List of Figures

doi:10.6342/NTU202404155

doi:10.6342/NTU202404155

doi:10.6342/NTU202404155

List of Tables

doi:10.6342/NTU202404155

doi:10.6342/NTU202404155

CHAPTER 1

Introduction

1.1 Background of the study

With the rapid growth of the world's population, the need for food and economic crops has increased dramatically. Ensuring the quality and production of agricultural crops is crucial. Insect pests are a major cause of lower crop quality and productivity. According to the Food and Agriculture Organization (FAO) of the United Nations, insect pests inflict at least $70 billion in global economic losses each year. Effective insect pest control can reduce crop productivity losses while simultaneously increasing earnings (Oerke, 2006). The idea of Integrated Pest Management (IPM) has been implemented around the world to address the risks caused by insect pests (Barzman et al., 2015; Kogan, 1998). IPM entails monitoring insect pest occurrences and executing appropriate management techniques (Lewis et al., 1997). Unlike traditional insect pest control strategies, which rely heavily on chemicals, IPM focuses on environmentally benign management practices. IPM can efficiently reduce insect damage to crops while also increasing production and quality. This technique combines diverse insect pest control strategies, such as biological control, cultural practices, mechanical approaches, and the selective use of chemical controls, to manage insect pest populations at acceptable levels while minimizing environmental effect (Potamitis et al., 2017). Farmers who practice IPM can reduce their reliance on chemical pesticides, which can impact non-target animals, human health, and the environment. These solutions not only maintain ecological balance, but also produce more sustainable and resilient farming

doi:10.6342/NTU202404155

enterprises (Lamichhane et al., 2016). As a result, integrated pest control contributes to the long-term viability of agricultural ecosystems by increasing biodiversity, lowering chemical residues in the environment, and ensuring crop output.

Climate change affects insect pest distribution patterns (Skendžić et al., 2021). Traditional insect pest detection systems rely on manual inspections, which are inefficient and do not provide the necessary responsiveness to tackle rapid insect pest outbreaks. In recent years, with the rapid advancement of the Internet of Things (IoT), embedded systems, machine learning, and deep learning, insect pest management has switched from manual to automated and intelligent methods (Parsons et al., 2019). Many studies on automated insect pest management use sticky paper traps, environmental sensors, and surveillance cameras into embedded systems (Abd El-Ghany et al., 2020). These systems leverage insect pest image recognition models and IoT technologies to deliver real-time monitoring findings to farm managers, eliminating the need for human labor and allowing for early interventions to mitigate possible economic losses caused by insect pests (Khan et al., 2019).

In addition to providing farm managers with real-time environmental and image information, data quality, image quality, and image recognition accuracy are also critical elements influencing management decisions (Xin & Wang, 2021). System faults during operation can result in insufficient data collection, and consumers cannot receive reliable data analysis unless high-quality data is available (Gao et al., 2020). Furthermore, several unpredictable elements in the field, such as lighting conditions, dust, or impediments during image collection, contribute to a reduction in image quality. If high-quality images are not captured, the effectiveness of future insect pest recognition model training will suffer, resulting in managers receiving inaccurate insect

doi:10.6342/NTU202404155

pest information. This insufficiency could lead to managerial judgments that don't completely align with the current needs.

1.2 Objectives

Our laboratory previously developed the Intelligent and Integrated Disease and Pest Management (I^2PDM) system, which is based on Internet of Things technology (Rustia, 2022). The I^2PDM system uses wireless imaging devices to take images of sticky traps using cameras, and then uses deep learning models to count the quantity of pests. Although this system has demonstrated significant application value in pest control, our practical experience with its operation has revealed that environmental and biological diversity issues such as rainy weather, dust, overexposure, or insufficient lighting can have an impact on image quality and, as a result, reduce recognition model accuracy. Variations within the same pest species, such as color, morphology, or size, can all have an impact on the model's accuracy. As a result, further optimizing the deep learning model to improve accuracy is an important challenge that must be address. Without precise information, managers cannot make decisions that are in line with current environmental needs, such as the proper amount of pesticide to employ or the timing of natural enemy releases, which could result in losses.

The fundamental goal of this research is to improve the model performance and enhance overall data quality. To achieve these goals, we suggest optimization techniques in three key components: image curation, model training, and data analysis. This study describes a system that uses machine learning and image enhancement approaches to increase the quality of existing pest image identification data. Improving

doi:10.6342/NTU202404155

data quality through statistical analysis and machine learning, and utilizing artificial intelligence (AI) to handle image quality and enhancement challenges. This means that the recognition model may be applied to all image data, not only high-quality images, which improves overall model training results. In terms of pest number statistics, this study develops a spatiotemporal analysis algorithm to handle missing data and increase accuracy, giving farm managers analysis results that are close to the ground reality. The particular objectives for achieving this goal are as follows:

- To design a framework for unsupervised image curation methods and integrate into existing systems
- To improve model performance through data augmentation technique and size feature extraction
- To modify the statistical results of the insect pests through voting mechanism and spatiotemporal algorithm to correct inaccurate data and fill in missing data

doi:10.6342/NTU202404155

CHAPTER 2

Literature Review

2.1 Image-based insect pest recognition techniques

In recent years, thanks to the parallel computing architecture of graphics processing units (GPUs), deep learning technology has become widely accessible. Combining machine vision and image recognition technologies, it has been successfully applied in image-related research fields, yielding numerous achievements (Voulodimos et al., 2018). Various image recognition techniques have also been applied to insect pest classification. This section reviews insect pest recognition technologies in three stages: traditional image processing methods, machine learning methods and deep learning methods.

2.1.1 Traditional image processing methods

Image processing is a type of signal processing that involves manipulating, analyzing, and interpreting images. It uses mathematical and computational techniques to improve image quality, extract image features, and interpret image data (Miranda et al., 2014). Several research have used image processing techniques to identify pests, highlighting their strengths and limitations. Deng et al. (2020) employed image segmentation to investigate three important maize diseases: corn blight, corn head blight, and maize rust. They caught illness characteristics in multiple color models by altering weight combinations, hence enhancing image segmentation efficiency and accuracy.

doi:10.6342/NTU202404155

This approach not only improved disease detection but also allowed for predictive research. Huddar et al. (2012) used typical image processing techniques to detect and discriminate whiteflies in greenhouses. They employed image segmentation to identify areas of interest (ROI) in different types of leaves, then converted the RGB color space to YCbCr to reduce computing complexity. After removing noise, their average accuracy in counting whiteflies was 0.96. Espinoza et al. (2016) created a pest recognition model using color features. They created RGB histograms and converted them to the Lab color space for image segmentation. They then converted the images to HSV color space for image enhancement and feature extraction. Using a feedforward neural network (FNN) for feature categorization, the trained model achieved recognition accuracies of approximately 0.96 for whiteflies and 0.92 for thrips.

According to the aforementioned research, existing image processing algorithms for pest recognition are limited since they can extract only shallow image features. These limits become especially apparent when dealing with the classification of several pest species or pests with minor physical distinctions. Traditional methods frequently fail in such situations, highlighting the necessity for more advanced feature extraction techniques.

To solve these issues, more advanced feature extraction approaches, such as machine learning and deep learning, are required. These sophisticated approaches can capture a greater variety of traits, resulting in better recognition and classification of pest species.

doi:10.6342/NTU202404155

2.1.2 Machine learning techniques

Unlike traditional image processing methods, machine learning frequently requires specialized knowledge and the development of various feature extraction techniques to allow the model to learn from data, hence increasing recognition accuracy. The choice of attributes, such as colour, shape, or texture, can have a significant impact on machine learning model performance. Several studies have demonstrated that machine learning is helpful for pest recognition, however these methods are not without limitations. Silveira and Monteiro (2009) developed a machine learning method for recognizing the small gray butterfly's eyespot patterns based on roundness and symmetry characteristics. This program also successfully identified patterns from many butterfly species. Wang et al. (2012) used an artificial neural network (ANN), various morphological features, and a support vector machine (SVM) to create a system with an accuracy of roughly 0.93. This technique was used to identify pests of several insect orders, such as Coleoptera, Hemiptera, Hymenoptera, Lepidoptera, Odonata, Neuroptera, and Orthoptera.

Kaya and Kayci (2014) proposed an approach based on color and texture features. They used the gray-level co-occurrence matrix (GLCM) and a feature-weighted K-means classifier to reach an accuracy of approximately 0.96 in identifying and monitoring Lepidoptera species. Li and Xiong (2018) created a mobile robot with a camera that used the Gaussian Mixture Model (GMM), bootstrap aggregating, and distance-regularized level set evolution (DRLSE) to reach a 0.95 accuracy in distinguishing Pyralidae species in fields. Liu et al. (2019) proposed a two-step recognition approach. They employed Otsu's binarization to segment color thresholds for Pyralidae species evaluation, followed by Hu moment-based form recognition and

doi:10.6342/NTU202404155

object contour detection. This model had an accuracy of 0.94 in recognizing Pyralidae species.

In addition to machine learning models with a single classifier, there exist methods for combining several classification models. Dey et al. (2016) employed the gray-level co-occurrence matrix and the gray-level run-length matrix to extract statistical features. They then used a support vector machine classifier with histogram characteristics to distinguish between pest-damaged and healthy leaf, and achieved an accuracy of 0.96.

According to the studies mentioned above, machine learning algorithms outperform classical image processing methods in terms of pest recognition. However, they still have some limits. Machine learning model training requires massive datasets and a significant time investment. Training a robust model is difficult when there is insufficient data. Furthermore, developing good feature extraction algorithms and understanding the underlying mathematics and statistical principles need significant talent, making the process dependent on specialized knowledge. Furthermore, feature engineering, or the manual design and selection of features, can be time-consuming and may fail to capture all of the information needed for complex tasks. These limits highlight the need for more advanced methods, like as deep learning, that can automatically extract features and manage larger datasets more rapidly, enhancing pest detection systems' accuracy and dependability.

doi:10.6342/NTU202404155

2.1.3 Deep learning techniques

Deep learning, as compared to machine learning, can extract features and provide output results automatically (LeCun et al., 2015). Convolutional Neural Networks (CNN) have demonstrated extraordinary performance in computer vision, with applications including image identification, object detection, and image segmentation (Deng & Yu, 2014). The most famous early CNN, LeNet, was launched in 1998 to recognize handwritten numbers (LeCun et al., 1998). At the time, many academics concentrated on creating CNN structures and optimizing models. However, LeNet performed well on small datasets but poorly on large datasets, falling behind machine learning algorithms. Consequently, for over a decade after LeNet's introduction, training on large datasets primarily relied on machine learning methods until the advent of AlexNet in 2012, which marked a significant change (Krizhevsky et al., 2012).

AlexNet had the lowest Top-5 error rate on the large-scale ImageNet dataset that year. In comparison to LeNet, AlexNet had a deeper model and benefited from advances in Graphics Processing Units (GPUs). Since then, CNNs have been intensively studied, with many architectures proposed. VGGNet (Simonyan & Zisserman, 2014) deepened the model; Network In Network (Lin et al., 2013) improved accuracy with less parameters; and GoogLeNet (Szegedy et al., 2015) used auxiliary classifiers to address the vanishing gradient problem in deep neural networks. MobileNet (Howard et al., 2017) prioritized efficiency, reducing model size and computation speed without significantly losing performance. EffieienNet (Tan & Le, 2019) successfully scaled network depth, breadth, and input image size with compound coefficients.

doi:10.6342/NTU202404155

The development of CNNs has not only optimized internal architectures but also experimented with combining different models to achieve deeper layers and faster convergence rates. ResNet combined VGGNet's increased layer count with GoogLeNet's enhanced computational efficiency within each convolutional layer, introducing residual learning to effectively address the degradation problem in deep networks (He et al., 2016). The trained model's error rate was lower than that of humans, heralding a new era for Deep Neural Networks (DNN). DenseNet differed from traditional neural networks by retaining feature reuse, preserving features from every preceding layer, making the network structure denser and mitigating the vanishing gradient problem (Huang et al., 2017).

Thenmozhi and Reddy (2019) created a classification model that identifies diverse Lepidoptera, Coleoptera, and Orthoptera pests using six convolution layers, five max pooling layers, and fully linked layers. They used image augmentation techniques like rotation, translation, and scaling to avoid overfitting. The experimental findings showed that this classification model attained an accuracy range of 0.95 to 0.97 across three publically available pest image datasets.

The deeper and more complex the convolutional neural network model, the better the training outcome, provided there are no issues with vanishing gradients or overfitting. Similarly, the datasets used to train CNN models are critical; larger, more thoroughly annotated datasets result in stronger models. However, this process demands a significant amount of time and human resources. Transfer learning, which was created on this foundation (Torrey & Shavlik, 2010), employs pre-learned weights from trained models, allowing for rapid convergence while training on new datasets and serving as a preliminary basis for model evaluation and optimization. Dawei et al. (2019) employed

doi:10.6342/NTU202404155

model weights trained on three independent insect datasets (NBXIR, Xie1, and Xie2) for transfer learning and achieved an accuracy of 0.94 in detecting ten pest kinds. To test the model's generalizability, the scientists used it to identify two types of weeds, with an accuracy of 0.99.

Deep learning's capacity to automatically identify features is less reliant on feature selection quality, resulting in higher classification results than machine learning. Wen et al. (2015) described moths using an approach that combined form, color, and texture data, and they proposed the improved pyramidal stacked de-noising auto-encoder (IpSDAE) as a deep neural network for moth identification. The model had an accuracy of 0.98 without distinguishing species and was unaffected by insect posture or angle.

In addition to image identification, deep learning is commonly utilized for object detection. Arsenovic et al. (2019) compiled a dataset of 79,265 plant disease images captured under various weather conditions. The team proposed PlantDiseasenet, a two-layer structure trained by both PDNet-1 and PDNet-2. PDNet-1 employed the You Only Look Once (YOLO) object detection algorithm to detect plant leaves, whereas PDNet-2 classified them. The trained model had an accuracy of about 0.94. Liu et al. (2017) used 1,053 apple leaf images, YOLOv3, and transfer learning to generate 13,689 synthetic images, then changed four convolutional neural network architectures (AlexNet, GoogLeNet, ResNet-20, and VGGNet-16) to detect four common pest-caused apple leaf illnesses. The updated AlexNet has an accuracy of about 0.98.

Deep learning's feature extraction capability addresses the challenge of feature selection in machine learning, allowing for quick model training without requiring extensive mathematical expertise (Lima et al., 2020). Through pre-training, optimization, and parameter adjustment, deep learning can handle more complex and

doi:10.6342/NTU202404155

challenging classification tasks. Consequently, deep learning has been widely applied in agriculture for automatic pest monitoring and management (Kamilaris & Prenafeta-Boldú, 2018).

2.2 Image quality assessment

Image quality assessment is a critical process to ensure that images meet the required visual and technical standards. Without high-quality image inputs, even the most accurate models will be rendered ineffective. However, in real-world imaging environments, it is challenging to guarantee that each captured image will be of optimal quality. As a result, undertaking an image quality evaluation is a necessary step for further image processing and model training.

2.2.1 Application in insect pest monitoring

Several research have shown that CNNs are useful in insect classification tasks, with great performance results (Thenmozhi et al., 2019). In keeping with these improvements, our team created a wireless sensor network device specifically for insect pests monitoring. This system takes images of sticky paper traps placed in front of the device and uses deep learning techniques to precisely measure the types and numbers of insect pests (Rustia et al., 2020). However, taking high-quality images for subsequent model training has a significant impact on the generalizability and overall data quality of deep learning models.

doi:10.6342/NTU202404155

2.2.2 Image curation

Recent studies have developed and implemented a variety of strategies to overcome image quality assessment difficulties. One such strategy is to use many unsupervised algorithms to improve image quality. Deswal et al. (2014) created the HSV image classifier, which is specifically used to recognize the background of sticky paper traps. The background in typical shots is usually consistent in color; for example, sticky paper traps seem yellow and green after white balance adjustment. Abnormal images, on the other hand, can have a variety of hues due to reasons such as lighting fluctuations, the presence of water droplets, obstructions, or camera wiring issues. To quantify this enhanced color volatility, the standard deviation of each color channel in the HSV space is calculated. Furthermore, using unsupervised learning to automatically identify outliers yields an overall measure of color variability (Susanto et al., 2020).

Clarity is another important consideration when evaluating image quality. Jiang et al. (2016) proposed using Laplacian variance as an indicator to assess the overall clarity of an image. This method uses unsupervised k-means clustering to calculate the degree of image blurriness. Blurred images can have a major impact on the performance of deep learning models; therefore, recognizing and filtering such images is critical for maintaining the dataset's quality.

Applying these two unsupervised techniques for image curation ensures the accuracy and stability of subsequent model training. High-quality images enable improved feature extraction, which increases the performance of deep learning models. This method not only improves model resilience, but also ensures that devices can reliably monitor and classify insect pests under a variety of environmental situations.

doi:10.6342/NTU202404155

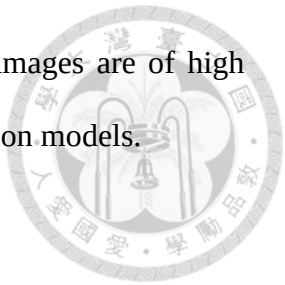
2.2.3 Image quality enhancement

Image enhancement is a technique for improving the quality, contrast, and visual perception of images, making them easier to read, analyze, and comprehend. This is often accomplished by amplifying or emphasizing key aspects and elements in the image. Image enhancement is frequently compared to image restoration; the primary distinction is that, whereas image enhancement subjectively affects the original image's features and details, image restoration focuses on objectively restoring an image to return it to a set standard. Image restoration aims to repair or recreate deteriorated images by estimating the degradation process, which includes salt-and-pepper noise, defocus blur, and motion blur (Mohapatra et al. 2014). Given the difficulties to maintain ideal image acquisition and imaging circumstances in real greenhouse settings, addressing or increasing different images is crucial to mitigate subsequent impacts on classification models caused by imaging issues.

Zamir et al. (2022) investigated the limitations of convolutional neural network models in responding to large-scale data, notably in the setting of high-resolution image restoration. The researchers created an efficient Transformer model that alters the architecture of the original forward propagation layer and multi-headed attention layer, allowing for interaction and application to high-resolution images. This model performed well in a variety of image restoration tasks, including motion blur, defocus blur, and image denoising. The experiments described above demonstrate how deep learning techniques can improve image quality, provide better features, and increase classification model performance. The application of such technologies to pest recognition studies has tremendous potential. Improving image quality through

doi:10.6342/NTU202404155

enhancement and restoration procedures guarantees that the input images are of high quality, thereby improving accuracy and reliability of pest classification models.



2.2.4 Indexes of image quality assessment

Image Quality Assessment (IQA) is considered as an image feature, with metrics used to quantify and evaluate image quality (Sara et al., 2019). These metrics aid in determining whether an image fulfills specific quality standards while also providing objective benchmarks for image processing and analysis. Accurate image quality assessment frequently requires a ground truth reference, which is difficult to get in real-world applications. Image quality is commonly assessed using reference measures such as Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR).

MSE, or squared error sum, is one of the most commonly used metrics for assessing image quality. A value closer to 0 corresponds to a higher quality image. Ebrahimi et al. (2017) used Support Vector Machines (SVM) with different kernels to detect thrips, using image length, hue, saturation, and brightness as indicators for creating the SVM structure. They also used MSE and Root Mean Square Error (RMSE) to test the classification model, and discovered that combining SVM with these image features resulted in an MSE of less than 0.023.

PSNR is another statistic used to assess image or video quality, specifically the similarity between original and processed or compressed images or films. It aids in determining the degree of information loss during compression, processing, or transmission. PSNR is measured in decibels (dB), with higher values indicating less disparities between the original and distorted images, resulting in higher quality. PSNR

doi:10.6342/NTU202404155

values ranging from 30 to 50 dB are usually regarded indicative of adequate image quality. Dai et al. (2020) introduced a Generative Adversarial Network (GAN) that uses the PSNR measure to recover low-resolution insect images, thereby increasing classification accuracy and recall. The testing results indicated an approximate 1.83-fold increase in recall and a 0.12 increase in classification accuracy.

MSE and PSNR are important tools for assessing image quality, but they have severe limitations. Both metrics are derived based solely on pixel values at specified places, interpreting the image as a collection of individual pixels and ignoring local or overall image structure information. To address this, the Structural Similarity Index (SSIM) was created. SSIM assesses not only the changes between an original and processed image, but also other important factors such as brightness and contrast, which may not be visible to the human eye but are critical for enhancing model training. Chodey and Noorullah Shariff (2022) suggested a four-stage model that used various deep learning methods. The first stage used Bayesian theorem for image denoising, the second stage applied LightenNet for image enhancement, the third stage utilized a Residual Network (ResNet) for image classification and recognition, and the final stage included the classification results in a fully connected layer. This method's effectiveness was validated on various insect datasets, achieving an SSIM measure of 0.99, making it suitable for real-time pest monitoring and detection.

doi:10.6342/NTU202404155

2.3 Data quality assessment

The proliferation of big data has had a significant impact on all research disciplines, including computer science, business, and medicine (Boyd and Marwick, 2011). Big data has also been used in agricultural techniques, including environmental data monitoring and pest management (Gevrey & Worner, 2006). These data are no longer simply technical, but have been collectively given by individuals and organizations from other sectors (Boyd & Crawford, 2012).

However, big data is only meaningful when it is valuable (Batty, 2012). Big data also brings challenges with data quality and usage, which can reduce its usefulness. Incorrect data-driven research does not fulfill the requirements of acceptable scientific practice in terms of validity or accuracy. Without a thorough understanding of data quality and related difficulties, this type of research may yield biased or inaccurate results (Liu et al., 2016).

2.3.1 Applications of enhancing data quality

Data quality is the process of characterizing and quantifying data's quantitative and qualitative qualities. Li et al. (2022) suggested a transfer learning-based data quality enhancement technique that applies machine learning systems to semantic analysis on social media, revealing a strong association between dataset quality and machine learning model performance. Therefore, assessing data quality is essential for improving model performance. Xin and Wang (2021) introduced a DCNN-G model that combines deep learning with Google Analytics, categorizing image quality into nine levels by

doi:10.6342/NTU202404155

adjusting image denoising parameters, and then using YOLOv4 to test and validate the categorized images. This approach achieved a recognition accuracy of 0.95, surpassing the original DCNN model's accuracy of 0.84. Lathey and Atrey (2015) proposed a method based on the Shamir Secret Sharing (SSS) algorithm to enhance the quality of encrypted image data in the cloud. This technology enables several image improvement operations, such as denoising, anti-aliasing, edge enhancement, contrast adjustment, and dehazing, to be done within the encrypted domain with minimum accuracy loss and decreased processing resources.

Fine-tuning is the process of changing a machine learning model that has been trained on a specific dataset to fit a new dataset. Tan et al. (2018) trained machine learning models using the AskAPatient and TwADR-L datasets, transforming text strings into vectors for model input. They chose pre-trained models, such as AWD LSTM in ULMFit or BERT, added layers specific to the final experimental objectives, and trained the model using the provided word vectors.

2.3.2 Indexes of data quality assessment

Determining the quality of data requires the use of applicable data quality assessment metrics. Currently, most data quality measurement standards are problem-specific and lack the fundamental principles necessary for practically applicable metrics (Pipino et al., 2002). Data quality is a multidimensional concept, encompassing various aspects of the data itself and objective measurements based on specific datasets (Puljak, 2016). When conducting data evaluation, a set of principles should be followed to develop assessment metrics tailored to specific needs (Cai & Zhu, 2015).

doi:10.6342/NTU202404155

The Simple Ratio Principle (SRP) evaluates data quality by calculating the ratio of predicted results to overall outcomes, with 1 being the most ideal result and 0 representing the least ideal result (de Aquino et al., 2018). The Minimum or Maximum Operation Principle can be used to evaluate various data assessment indicators simultaneously. The Minimum Operation Principle is a cautious technique that ensures that the total value of data dimensions does not exceed that of the weakest data assessment measure (Chen et al., 2021).

Ballou et al. (1998) proposed a general metric for evaluating the timeliness dimension, applicable to the assessment of volatility in monetary terms over time. When each variable has its own importance in evaluating a particular dimension of data, Weighted Averages are appropriate. Each weight should range between 0 and 1, and the sum of all weights should be equal to 1 (Karkouch et al., 2016).

2.4 Application of voting and spatiotemporal analysis

2.4.1 Feature extraction in machine learning

In the field of machine learning, feature extraction is the process of extracting relevant features or qualities from raw data and using them to train and analyze machine learning models. These qualities can take several forms, including numerical data, structured data, images, and audio. The basic purpose of feature extraction is to convert raw data into a format that machine learning models can interpret. Machine learning ensemble classification is a widely utilized method for pest identification and classification. This strategy often mixes numerous machine learning models and bases the final categorization decision on their combined predictions.

doi:10.6342/NTU202404155

Atallah and Al-Mousa (2019) proposed a majority vote ensemble method for predicting the chance of people developing heart disease. This study trained on real-life data from healthy and ill patients and classified them using the majority vote of various machine learning models. This method proved to be more accurate than utilizing a single model, with an accuracy rate of 0.9. Kasinathan and Uyyala (2021) used machine vision and several feature extraction approaches, such as texture, color, and shape, to classify crop pests. They used a variety of machine learning approaches, including basic and ensemble classifiers, and assessed the classification results using majority voting. Finally, they had an accuracy rate of around 0.92 for 24 different pest species.

2.4.2 Ensemble voting mechanism

Ensemble Voting Classification is a machine learning technique that improves the accuracy and performance of prediction models. It integrates numerous machine learning models, each of which makes its own predictions, to determine the final forecast using a majority voting technique.

Kumari et al. (2021) recommended that the ensemble soft voting classifier be used for binary classification, specifically for early detection and prediction of diabetes incidence. The ensemble uses three machine learning methods for classification: random forest, logistic regression, and support vector machine (SVM). The method is also contrasted and examined using breast cancer datasets, with the suggested ensemble soft voting classifier reaching an accuracy of around 0.97.

doi:10.6342/NTU202404155

2.4.3 Spatiotemporal analysis

For time series data analysis, instability or misclassification in the system often leads to missing or incorrect data. In sections 2.4.2 and 2.4.3 discussed two methods for image curation and data quality enhancement, using a voting mechanism to eliminate the misjudged parts. Nasri et al. (2023) proposed the spatiotemporal proximity distribution, observing the number of students at various locations on a campus over time through wearable devices. They recorded the movements of students within the campus using a spatiotemporal matrix, observing and analyzing students' social networks and their social behaviors.

Throughout the literature review in this chapter, some algorithms and strategies were used to improve the overall quality of the pest recognition system. This involves enhancing the quality of collected images by image curation, optimizing deep learning models to increase recognition accuracy, and using voting and spatiotemporal techniques to improve statistical data quality. These approaches eliminate errors and produce analysis data that closely resembles real-world settings.

doi:10.6342/NTU202404155

doi:10.6342/NTU202404155

CHAPTER 3

Materials and Methods

3.1 System overview

The objective of this study is to enhance model performance and overall data quality of the current I^{2}PDM system, providing environmental data and accurate counts of insect pests that closely approximate real-world conditions. The framework of research is illustrated in Fig. 3-1, which contains three key components: image curation, model training, and data analysis.

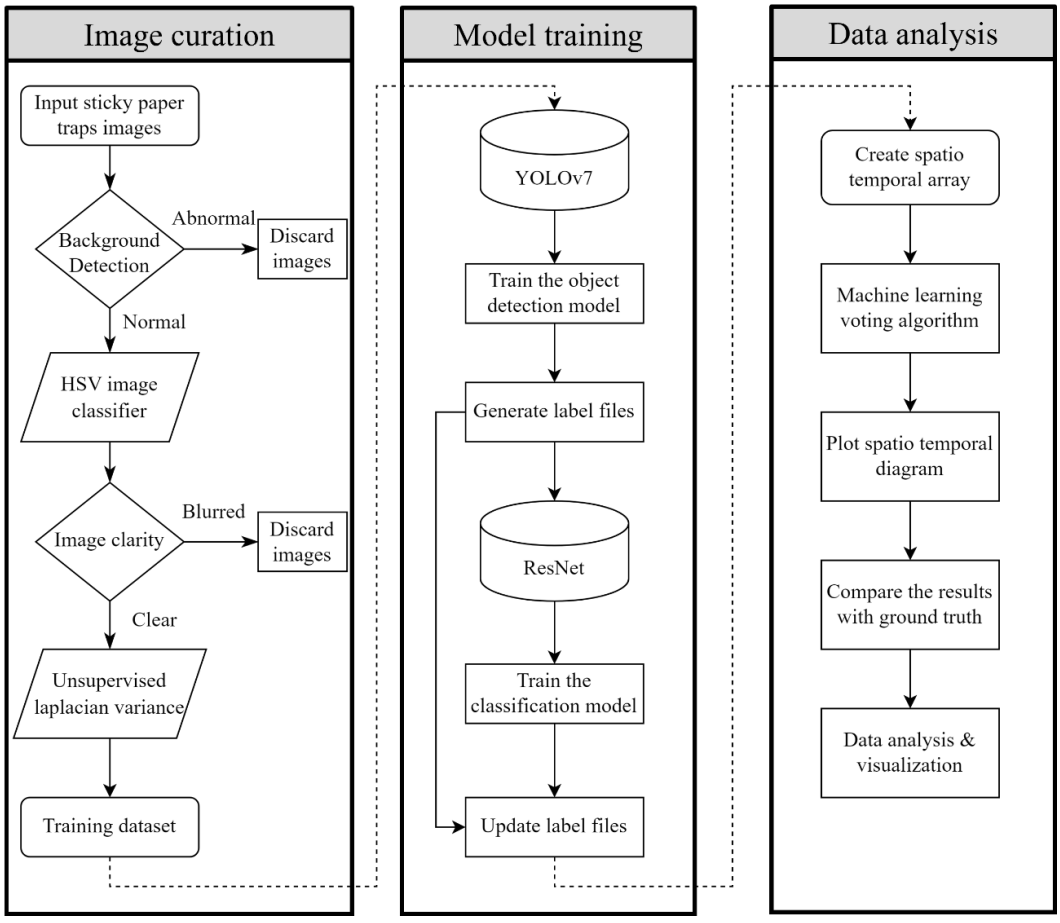


Fig. 3-1. Illustrations of research framework.

doi:10.6342/NTU202404155

3.2 System architecture

In this chapter, we will introduce the optimization of the entire system architecture, including three key components: image curation, model training, and data analysis.

3.2.1 Image curation

In the image curation process, images of sticky paper traps were recorded with wireless sensor network devices and sent to a laboratory server for model training and data analysis. Weather, illumination, and environmental conditions like as dust, water droplets, and glare can all have an impact on the camera module and result in low-quality images when collecting continuous data.

This work offers a two-step image curation approach, with a focus on background identification of sticky paper traps and image deblurring. The first step in the image preparation process is to remove low-quality images. These images often feature more than two backdrop colors. To identify them, we transform the RGB color space to the HSV color space and determine the standard deviation of each color channel, as shown in Eq. 3-1:

$$||C_1 - C_2|| = \sqrt{(C_{1,H} + C_{2,H})^2 + (C_{1,S} + C_{2,S})^2 + (C_{1,V} + C_{2,V})^2} \quad (3\text{-}1)$$

Where $C_1 - C_2$ represents the difference in color distance, by calculating the hue (H) component, saturation (S) component, and value (V) component in the HSV color space to see if the background colors differ too much, hence determining whether the background color of sticky paper traps is a single color. Combining these deviations

doi:10.6342/NTU202404155

yields an overall measure of color variability, allowing us to distinguish between high-quality and low-quality images.

The next step involves checking the clarity of the images using Laplacian variance to calculate an indicator of overall image clarity, as shown in Eq. 3-2. In this study, the unsupervised K-means method was employed to automatically classify images into normal and abnormal categories. The model determined thresholds autonomously, eliminating the need for empirical observations.

$$Laplacian\ Variance = \frac{1}{N}\sum_{x=1}^{N}[L(x) - \bar{L}]^2 \quad (3\text{-}2)$$

Where Laplacian variance is a measure of the variance of the image, which is used to determine the sharpness or blur of an image. N is the number of pixels in the image, $L(x)$ is the Laplacian value of the x-th pixel, and $\bar{L}$ is the mean of the Laplacian values of all the pixels. A higher value of Laplacian variance indicates a sharper image, whereas a lower value of Laplacian variance indicates a blurrier image.

3.2.2 Model training

After completing the image curation, we obtained high-quality image for model training. Initially, we employed YOLOv7 object detection model to identify and mark all objects present on the sticky paper traps, the operation as shown in Fig. 3-2.

doi:10.6342/NTU202404155

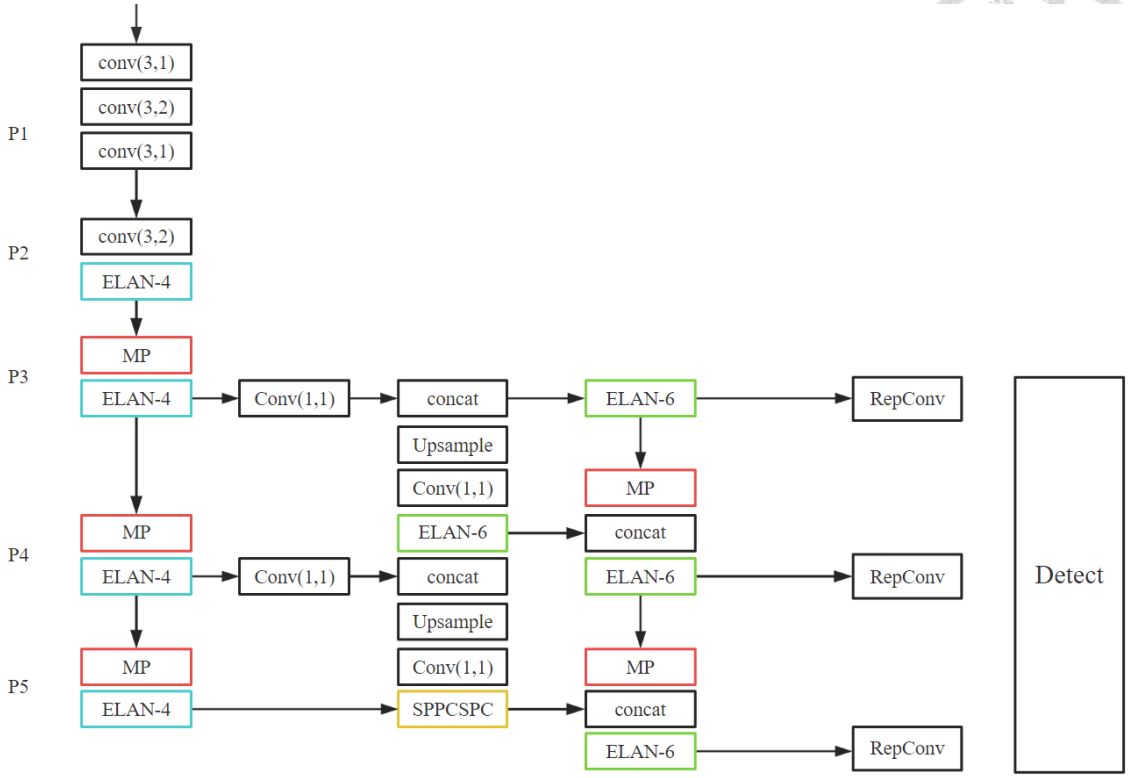


Fig. 3-2. YOLOv7 object detection model architecture (Wang et al., 2023).

First, the original 3280 × 2464 RGB image captured by the Raspberry Pi camera module was scaled to 3200 × 2400 using bicubic interpolation to ensure equal division. The resized image was then tiled into 5 images of length and 4 images of width, for a total of 20 images at 640 × 640 resolution. Tiling is an object detection approach that improves detection accuracy by breaking a high-resolution image into smaller sections, lowering the size of the object detector's input image. This method is especially successful in detecting little items. Tiling is an object identification approach that improves detection accuracy by breaking down a high-resolution image into several smaller parts. This method reduces the dimensions of the input image for the object detector, resulting in better detection of small things. The image tiling resolution of

doi:10.6342/NTU202404155

640 × 640 was chosen since it corresponds to the YOLOv7 input size, removing the need for extra resizing and serving as individual inputs to the object detection model.

We next used a residual network to classify insect pests on the identified objects. The authors of ResNet developed the concept of residual modules, which use shortcut connections to add the output of a prior layer to subsequent stacked layers, as illustrated in Fig. 3-3. By integrating residual modules, the model may more quickly learn the residual between input and output, allowing for smoother deep network training. This effectively tackles the vanishing gradient problem that arises while deepening models.

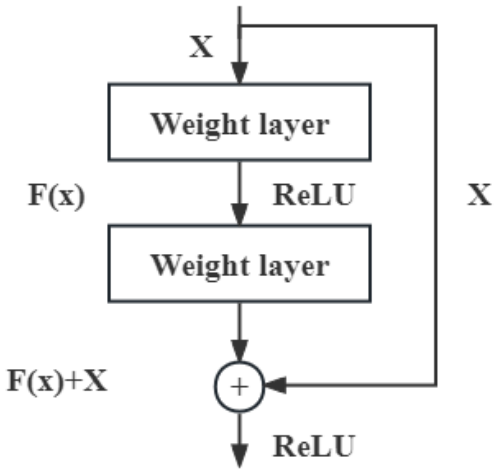


Fig. 3-3. The shortcut connection of ResNet.

In this research, we select ResNet-18 model, a lightweight model within the ResNet series, having 18 stacked layers that retain key ResNet properties. It maintains a balance between model correctness and computing efficiency. ResNet-18 consists of an input layer, convolutional layers, residual blocks, a global average pooling layer, fully

doi:10.6342/NTU202404155

connected layers, and an output layer. The input layer accepts images as input to the model, then performs convolution operations with a stride of 2 to minimize image size. Outputs are then processed using stacked residual blocks, with each block consisting of two convolutional layers. Each convolutional layer is followed by batch normalization and Rectified Linear Unit (ReLU) activation functions, and residuals can be propagated effectively using skip connections.

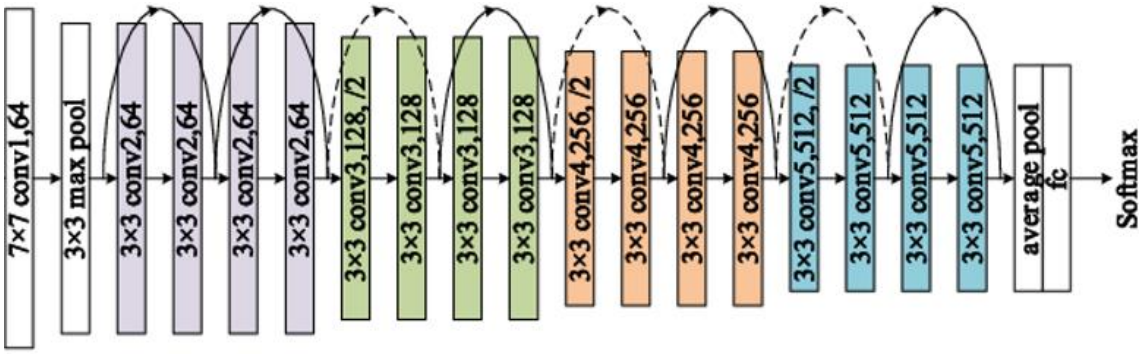


Fig. 3-4. ResNet-18 model architecture diagram (He et al., 2016).

In this study, following the architecture of current I^{2}PDM systems, insect images are scaled to 128 x 128 pixels as input. This ensures compatibility of the new model's input with the existing system, allowing seamless integration into the I^{2}PDM system without requiring additional adjustments.

The advantage of using a two-stage approach over a one-stage approach lies in the practical challenges presented by insect pests on sticky paper traps—they are often too small, densely populated, and may overlap. Therefore, compared to combining object detection and insect pest classification into a single step using YOLOv7, employing a

doi:10.6342/NTU202404155

two-stage approach simplifies the complexity of a single model, thereby enhancing the model's generalization capability. During training with YOLOv7, label files are generated to record the spatial coordinates of detected objects. Following classification with the residual network, the classifications are mapped onto the coordinates of the YOLOv7-detected objects. Finally, this process produces a label file containing the types and populations of insect pests.

3.2.3 Data analysis

After training the object detection and insect pest recognition models, we used the resulting label file to analyze the data and calculate the types and numbers of insect pests. However, the analysis results derived directly from raw data contained major mistakes. For instance, in Fig. 3-5, On days 1, 2, and 4, the system correctly identified the insect species and provided precise coordinate information. On Day 3, however, one misidentification occurred, and one insect was not detected. In this paper, we present a voting system and a spatiotemporal algorithm to handle the problem of model misclassification and missing data, as shown in Fig. 3-6.

doi:10.6342/NTU202404155

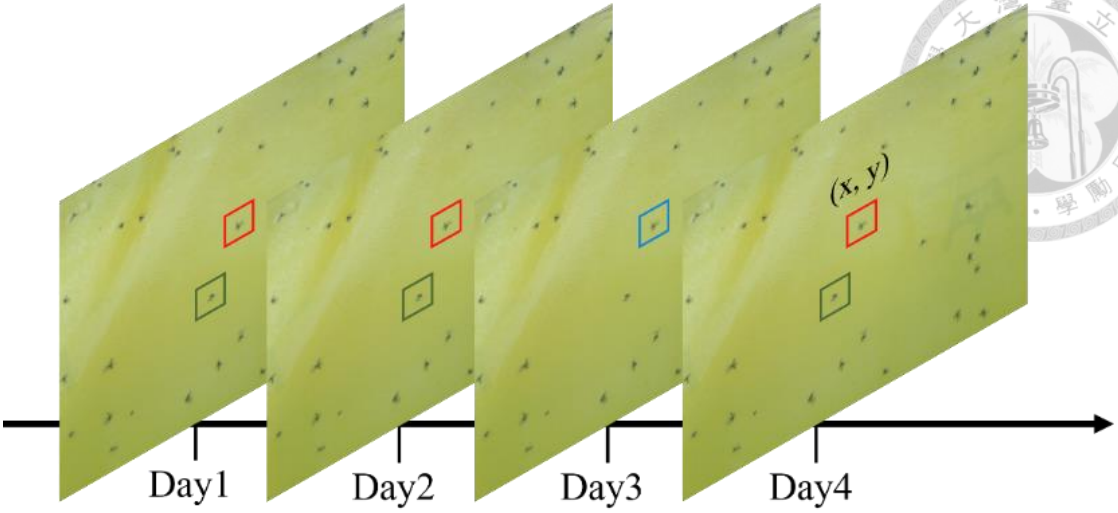


Fig. 3-5. Illustrations of misclassification and missing data.

This method pseudocode has two important components: majority voting and spatial temporal correction. First, compute the occurrences of each insect pest during a certain time interval, recording the starting time point and coordinates at which each insect pest is first identified. Use these coordinates to see if an insect pest remains on the sticky paper trap. If an insect pest is observed at the same coordinates for more than half of the time points, it is considered to be continuously present. If it does not match this requirement, the model considers it a misclassification.

Second, after using the voting mechanism to reduce misclassifications, the number of insect pests may still be inaccurate. It is important to correct these misclassifications to the correct insect pest species in order to avoid an excessive amount of missing data, which could lead to incorrect statistical results. By comparing coordinates at previous and subsequent time points, misclassifications and missing data can be adjusted to the correct insect pest species, producing more accurate statistical results.

doi:10.6342/NTU202404155

Algorithm 1 Spatiotemporal Correction Algorithm

Input: Raw data containing labels generated by the object detection and insect pest recognition models
Output:: Corrected data with classifications of insect pests

procedure MAJORITY_VOTING(predictions)
    counted_predictions ← Count(*predictions*)
    max_occurrence ← Max(*counted_predictions*)
    if *max_occurrence* $> \frac{1}{2} \times$ TotalCount(*predictions*) **then**
        majority_class ← Class with highest occurrence
    else
        majority_class ← Undetermined
    end if
    return *majority_class*
end procedure
procedure SPATIAL_TEMPORAL_CORRECTION(data, label_file)
    corrected_data ← EmptyList
    for each entry **in** label_file **do**
        current_species ← entry.species
        current_time ← entry.time
        current_coordinates ← entry.coordinates
        neighbor_species ← EmptyList
        for each neighbor_entry **in** label_file **do**
            if |neighbor_time − current_time| ≤ Time Intervals **and** Distance(neighbor_coordinates, current_coordinates) ≤ Threshold **then**
                neighbor_species.Append(*neighbor_entry.species*)
            end if
        end for
        corrected_species ← Majority_Voting(*neighbor_species*)
        if corrected_species ≠ current_species **then**
            corrected_data.Append({*species* : corrected_species})
        else
            corrected_data.Append({*species* : current_species})
        end if
    end for
    return *corrected_data*
end procedure

Fig. 3-6. Spatiotemporal algorithm pseudocode

doi:10.6342/NTU202404155

Next, we employ Mean Squared Error (MSE) and Mean Absolute Percentage Error (MAPE) to analyze the differences between raw data, spatiotemporal analysis, and ground truth. Using these statistical quantitative criteria, we can assess if the voting mechanism and spatiotemporal algorithm enhances the overall data quality. This technique highlights how our proposed analytical methodology improves the accuracy of statistical data in this study, allowing it to better approximate real-world data.

3.3 Experimental sites

The wireless sensor network devices were installed in greenhouses in Taiwan. The information for each site is presented in Table 3-1, and the geographical locations of the installation sites are illustrated in Fig. 3-7. Three of the sites were established by our team previously, while the remaining three were set up recently by this work (Rustia et al., 2022). At each site, the number of installed sensor nodes was determined based on recommendations from the farm managers. The sensor nodes at these sites continuously collects environmental data and images of sticky paper traps, and then transmitted data back to the laboratory server.

doi:10.6342/NTU202404155

Table 3-1. Basic information of each installation site.

Field name	Geographical location	Crops	Number of nodes
T1	Yunlin, Taiwan	Tomato Seedlings	10
T2	Chiayi, Taiwan	Tomato Seedlings	7
S1	Chiayi, Taiwan	Strawberry	8
O1	Chiayi, Taiwan	Orchid	6
O2	Chiayi, Taiwan	Orchid	4
M1	Taipei, Taiwan	Melon	6

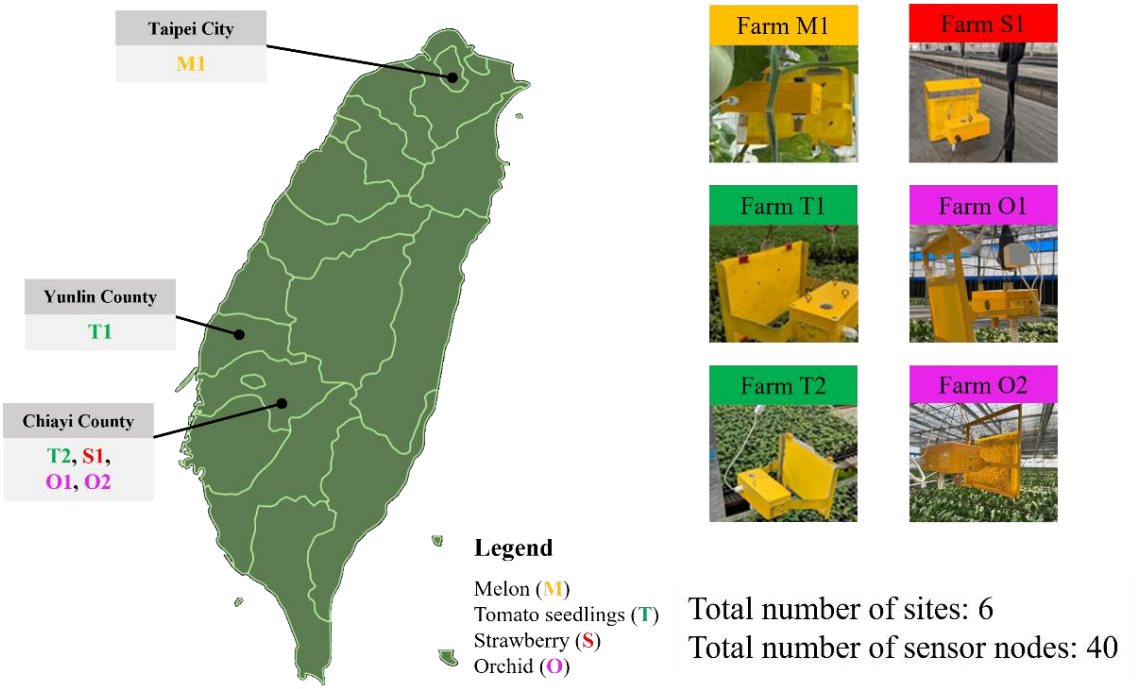


Fig. 3-7. Geographical location of each installation site.

doi:10.6342/NTU202404155

The number of installed devices in each site is related to the greenhouse area, the farmers' management strategies, and the pest hotspot areas. The experimental sites are independent of each other in terms of crop cultivation, management strategies, and weather factors. The crops grown are mostly major economic crops in Taiwan, including tomato seedlings, orchids, melons and strawberries. The installation setup maps for each site are shown in Fig. 3-8 to 3-13.

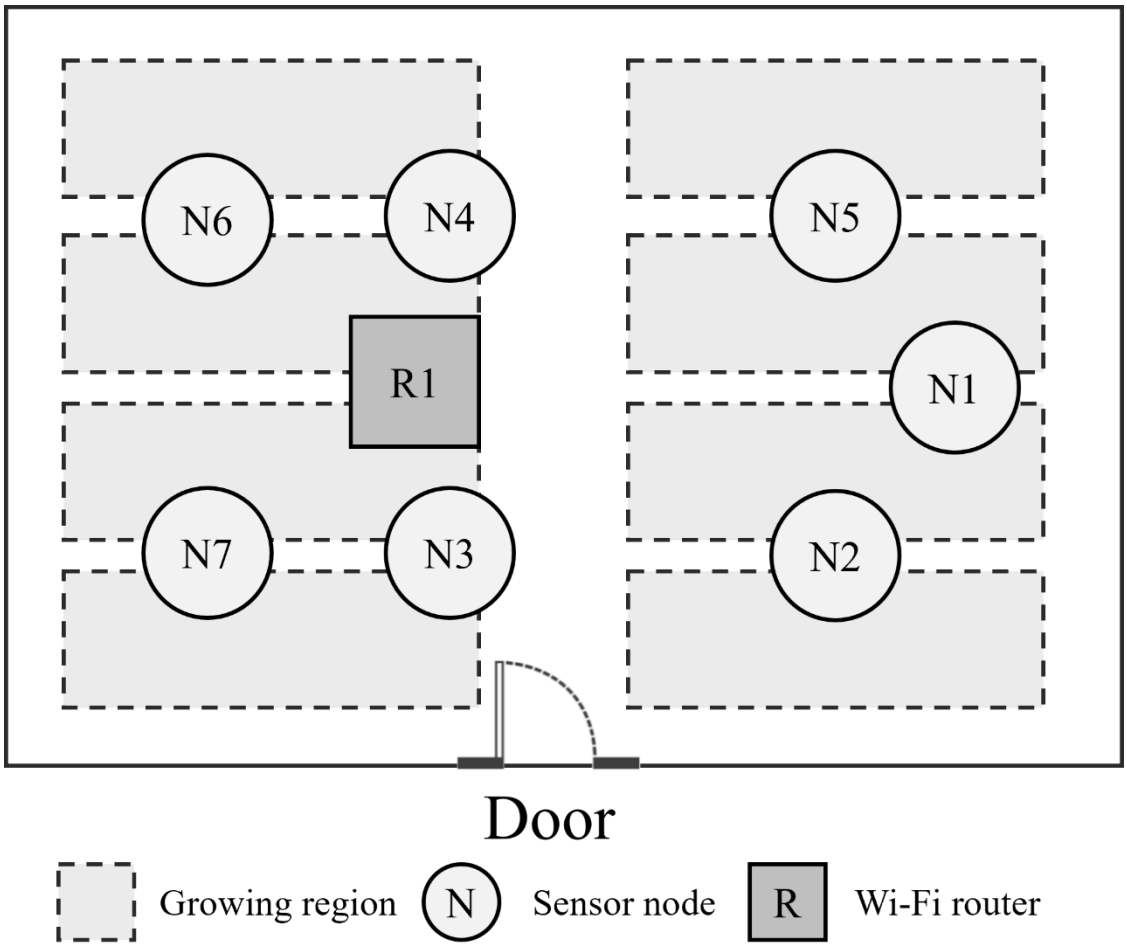


Fig. 3-8. Field map of Farm T1.

doi:10.6342/NTU202404155

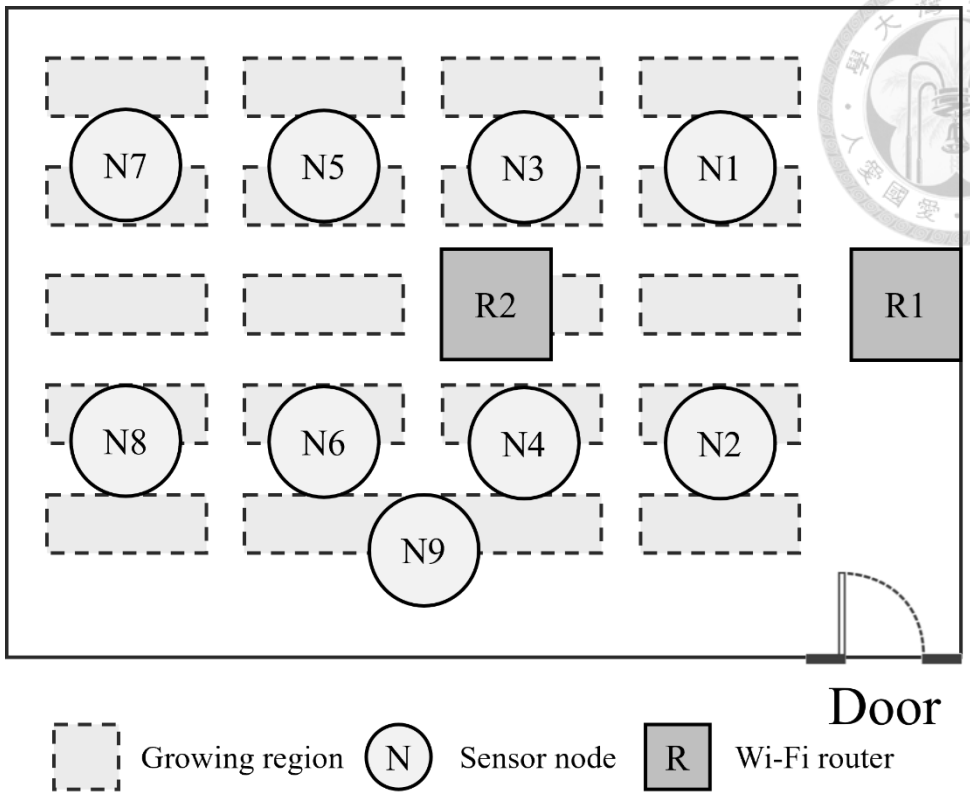


Fig. 3-9. Field map of Farm T2.

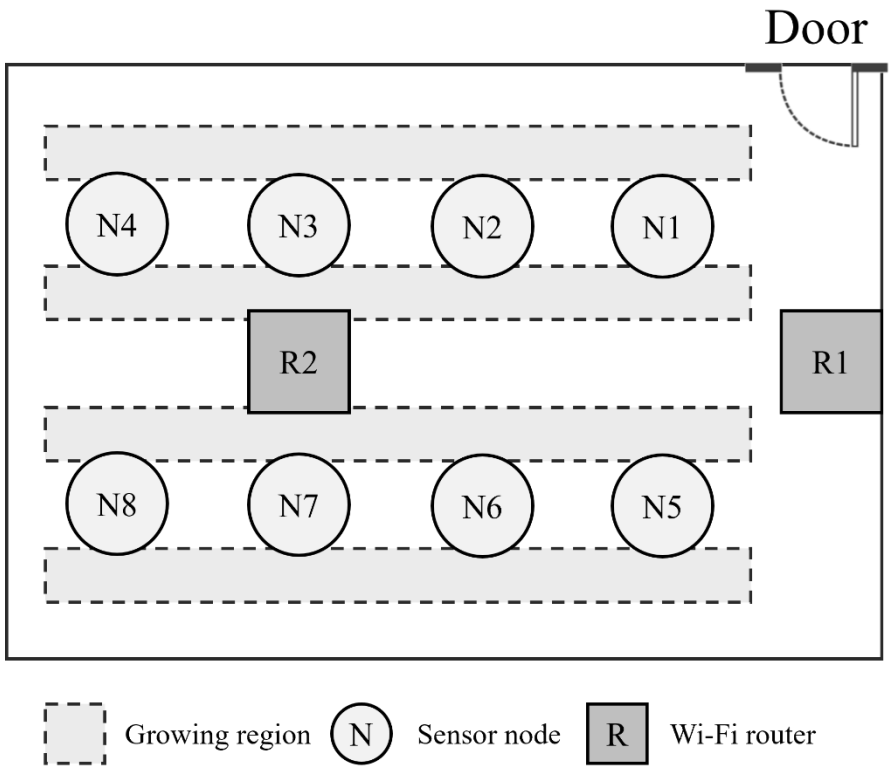


Fig. 3-10. Field map of Farm S1.

doi:10.6342/NTU202404155

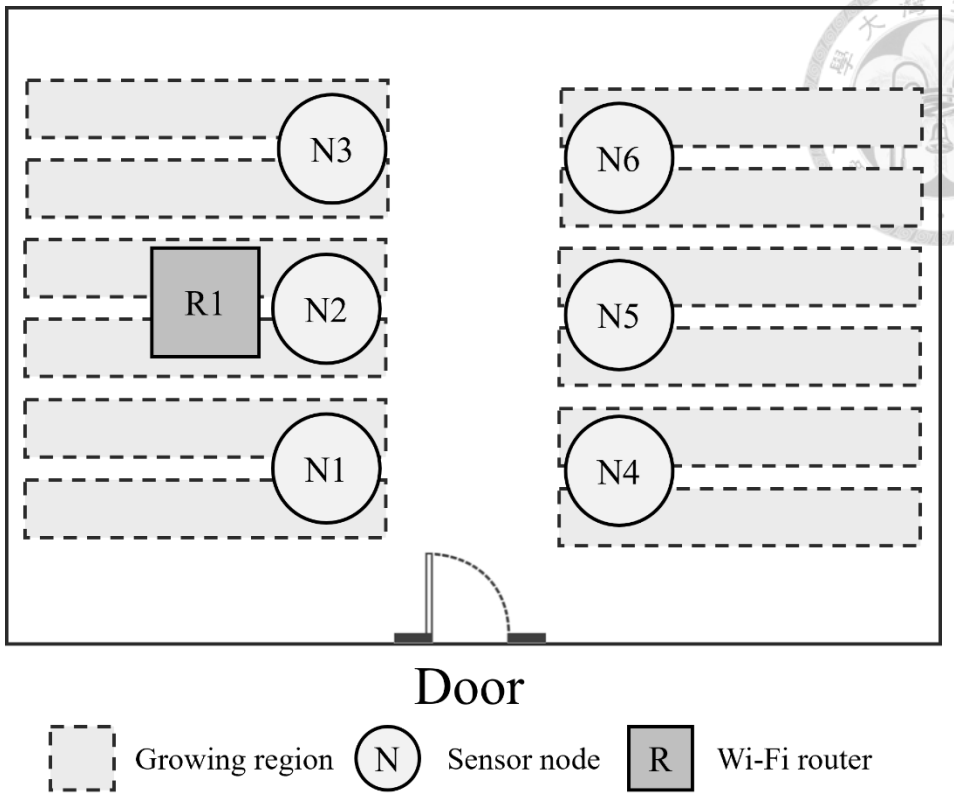


Fig. 3-11. Field map of Farm O1.

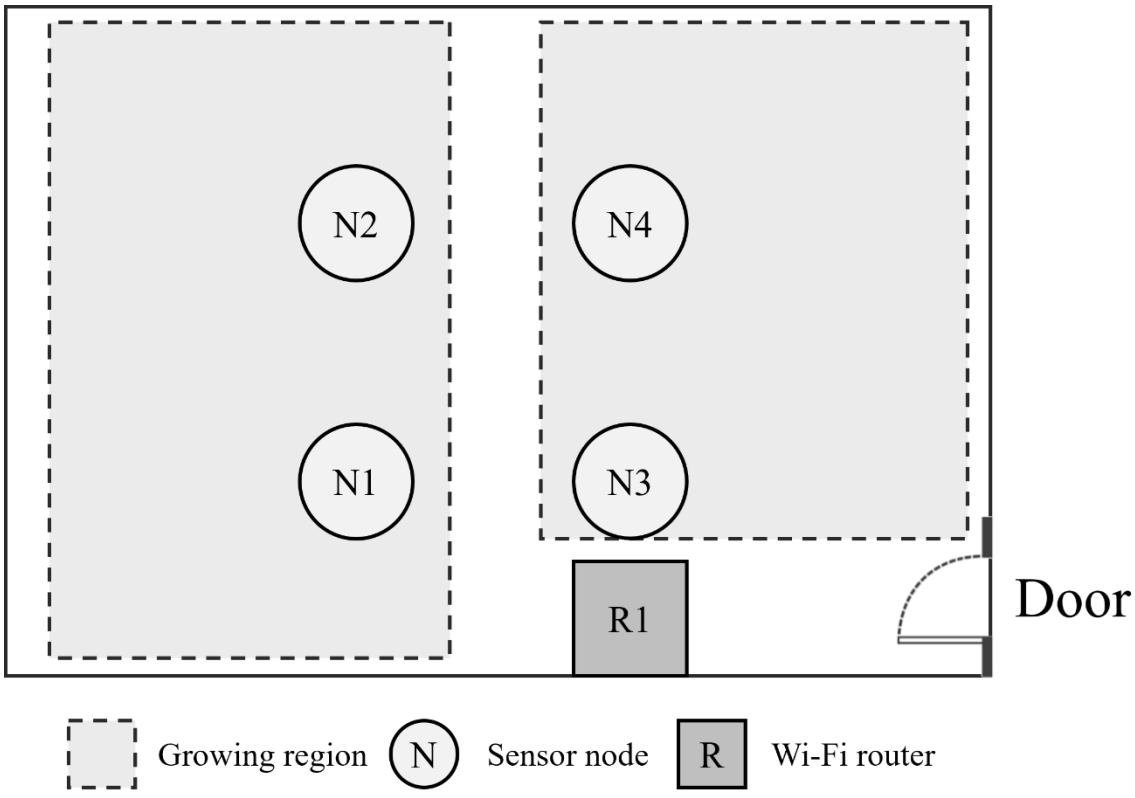


Fig. 3-12. Field map of Farm O2.

doi:10.6342/NTU202404155

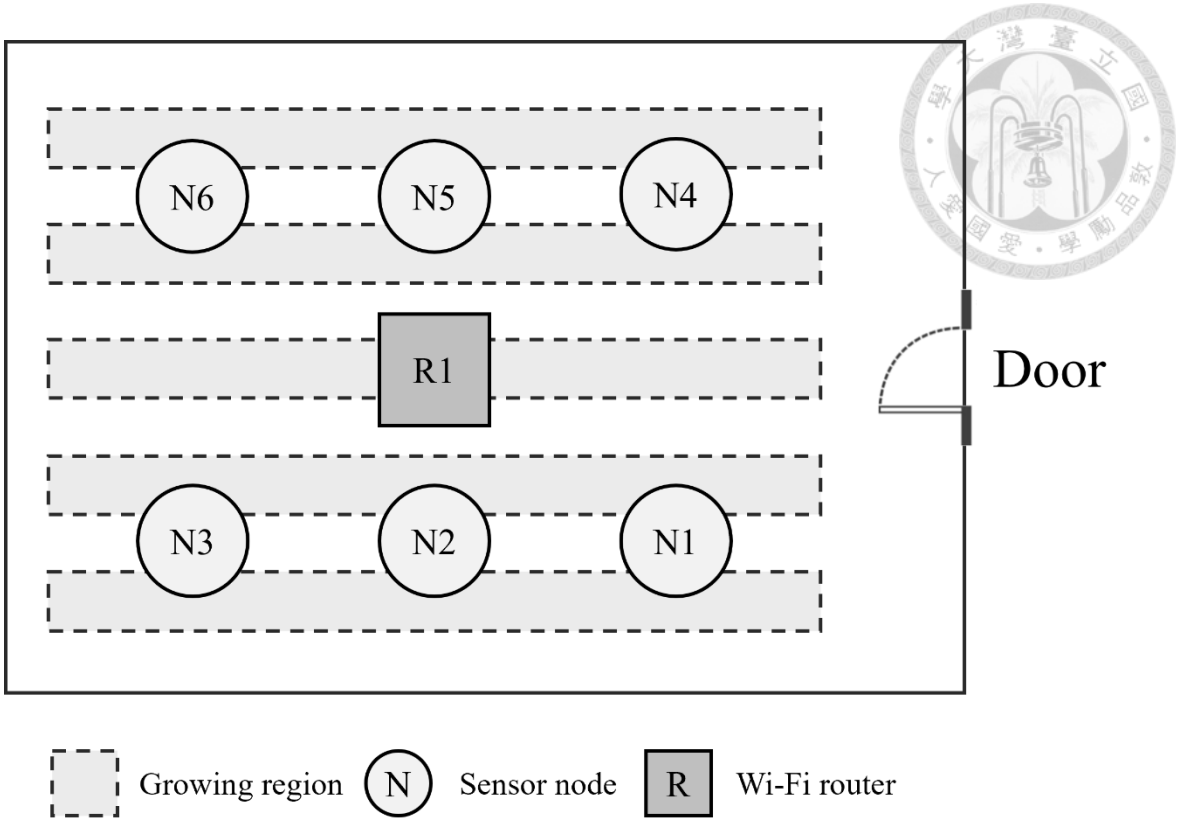


Fig. 3-13. Field map of Farm M1.

The devices were installed in the field following discussions with the farm manager, with an average distribution across the area. The devices were intentionally placed as close to the growth zone as acceptable. A Wi-Fi router was utilized to offer network access to the sensor nodes, allowing the acquired environmental and image data to be sent back to the server.

doi:10.6342/NTU202404155

3.4 Data collection

3.4.1 Wireless sensor network module

The insect pest image dataset was collected by previously developed a wireless sensor network module named Intelligent and Integrated Pest and Disease Management (I^2PDM) in our laboratory (Rustia et al., 2020), as shown in Fig. 3-14. This module is based on a Raspberry Pi embedded system, with an aluminum stand mounted in front of Raspberry Pi Camera Module V2 and sticky paper traps placed to attract pests. The module was installed in greenhouses at National Taiwan University, Yunlin, and Chiayi. It automatically monitors environmental data and insect pest population in greenhouses growing melons, tomatoes, strawberries, and orchids, and then transmits the data back to the laboratory server using Wi-Fi and MQTT communication protocols.

Fig. 3-14. Illustrations of a wireless sensor network device.

doi:10.6342/NTU202404155

3.4.2 Target insect pests

In this study, we selected eight common greenhouse insect pests that are commonly detected on sticky paper traps, including fly, gnat, moth fly, midge, thrips, mosquito, cranefly, and whitefly, as seen in Fig. 3-15. Cranefly and gnat larvae feed on plant roots, which can cause root system damage. Fly and midge larvae lay eggs on plant tissues and feed on them, potentially causing tissue rot and destruction. Mosquito and moth fly larvae eat plant roots and organic materials, resulting in root injury and degradation. Thrips feed on plant cell sap, causing tissue damage and often serving as vectors for viral transmission. Whiteflies feed on plant sap, with large infestations causing poor plant growth and spreading viruses. These pests can severely impact crops, leading to stunted growth, reduced yield, and increased susceptibility to other diseases.

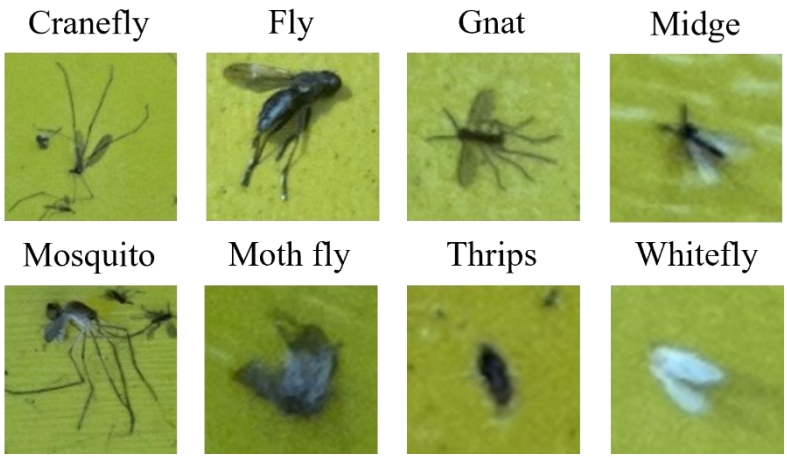


Fig. 3-15. Eight common greenhouse insect pests in greenhouse.

doi:10.6342/NTU202404155

3.4.3 Data augmentation

The dataset used in this study is based on the eight-category insect pest dataset used in earlier studies. However, due to the limited number of samples available for specific categories, such as cranefly and mosquito, as well as the issue of class imbalance, data augmentation techniques were used to expand and balance sample sizes across all categories. Furthermore, the “Others” category was included to avoid the model from incorrectly putting non-insect pest samples into one of the existing insect pest categories, as illustrated in Fig. 3-16.

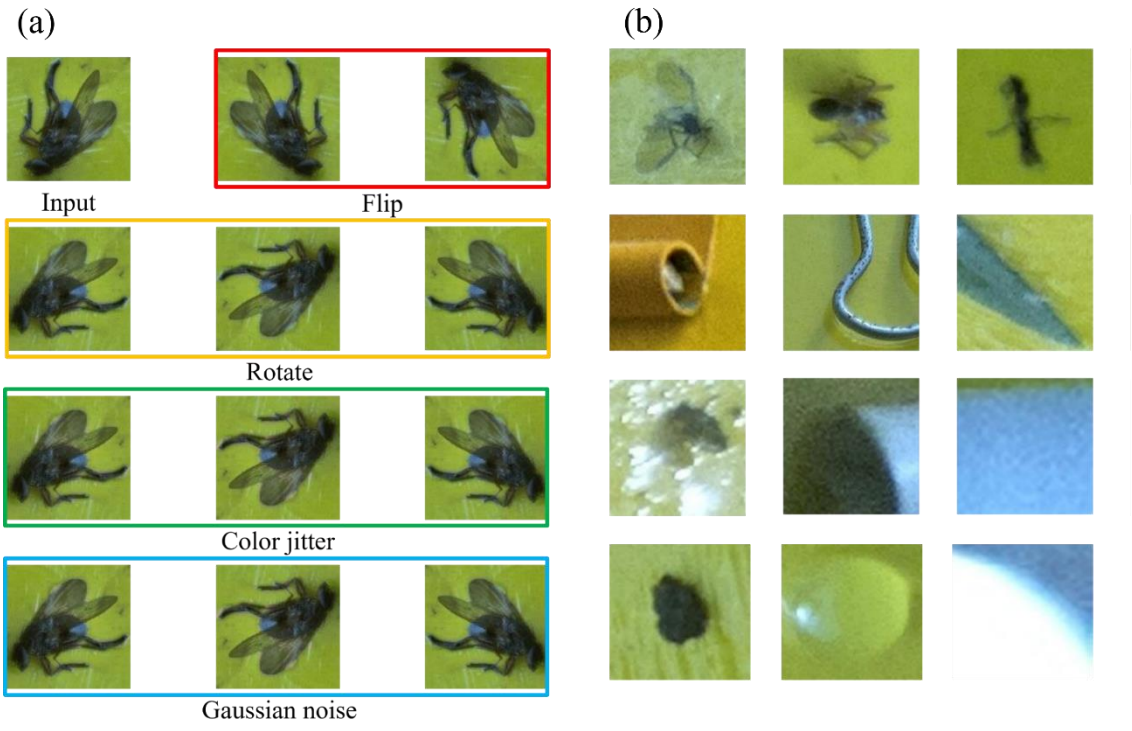


Fig. 3-16. (a) Data augmentation techniques were used to randomly increase the dataset; (b) Illustration of non-target insect pest species, including aphid, spider, leaf, water droplet, glare and some obstacles.

doi:10.6342/NTU202404155

3.5 Size feature extraction

Based on previous experiences with the I^{2}PDM system, convolutional neural networks (CNNs) have shown insensitivity to insect size due to the fixed-size input constraint. This leads to misclassification when insect pests of different sizes but similar morphology are present. For example, fly and gnat both have two wings, and despite significant size differences, their morphological similarities at certain angles cause the model to often misidentify flies as gnats. To address this issue, we decided to incorporate insect size as an additional feature in the model. By doing so, the model can take size information into account when identifying insect pests, thereby improving its ability to distinguish between different species.

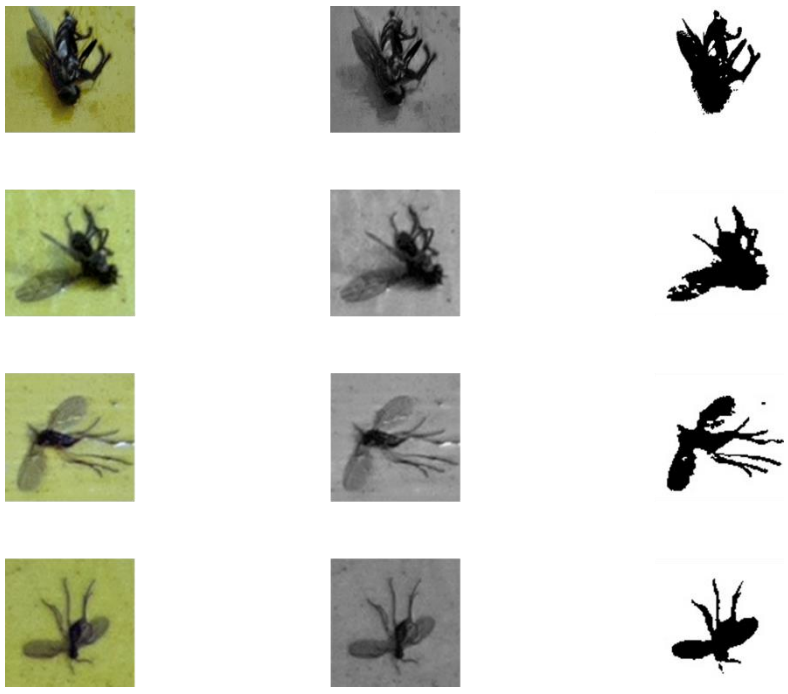


Fig. 3-17. Manual extraction process of size features.

doi:10.6342/NTU202404155

Fig. 3-17 illustrates the process of manually extracting insect image size features. First, the input RGB image is converted to a grayscale image based on Equation 3.3. This formula weights each pixel's RGB values according to the human eye's sensitivity to different colors, producing a new grayscale value. This method extracts grayscale information from a color image.

$$Y = 0.299 \times R + 0.587 \times G + 0.114 \times B \tag{3-3}$$

Where Y represents a new grayscale value, and R, G, B represent the original image's color channels. This method extracts grayscale information from a color image.

Next, we apply the Otsu algorithm to binarize the grayscale image. Otsu's method is an automatic threshold selection technique based on the image's grayscale histogram (Otsu, 1979). For images with significant contrast, such as sticky paper traps where the background color contrasts sharply with the insect body color, Otsu's method effectively binarizes the image, as shown in Fig. 3-18. This allows us to mark the background as white and the insect bodies as black, thereby clearly distinguishing the insect regions.

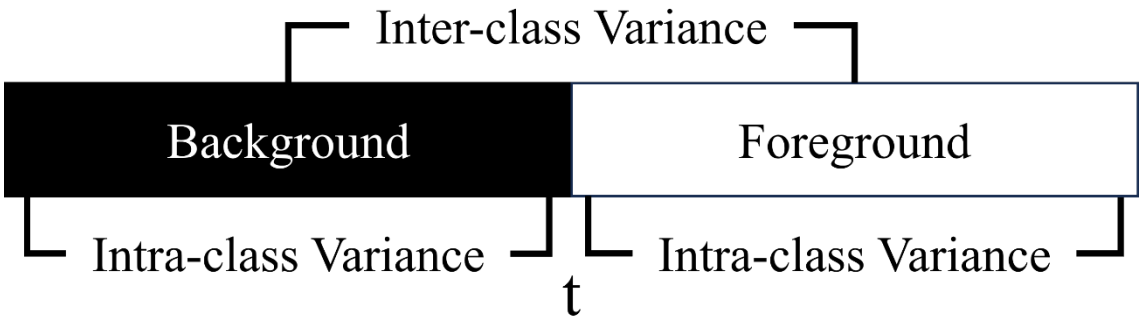


Fig. 3-18. OTSU Thresholding method

doi:10.6342/NTU202404155

Finally, we calculate the number of black pixels using Equation 3.4 and divide it by the cropping area of the object detection model to obtain the proportion of the insect on the sticky paper traps. This provides the size feature of the insect.

$$Size\ ratio = \frac{Number\ of\ black\ pixels}{Cropping\ area\ of\ object\ detection\ model} \quad (3\text{-}4)$$

Additionally, to prevent the size feature from being diluted in the high-dimensional feature vector, we added an extra fully connected layer after the fully connected layer of the original ResNet-18 model. The purpose of this additional fully connected layer is to reduce the original 512-dimensional feature vector to 31 dimensions. In this fully connected layer, the size feature is injected, resulting in a final feature vector consisting of 32 dimensions that include the size feature.

3.6 Model evaluation

After completing the model training, we conducted a comprehensive evaluation on the testing dataset. The evaluation of the object detection model included calculating the mean Average Precision (mAP). Specifically, mAP:.5 refers to the mAP calculated at an Intersection over Union (IoU) threshold of 0.5. For the evaluation of the insect pest classification model, the F1-score was computed. The F1-score is a commonly used classification performance metric that combines precision and recall to evaluate the

doi:10.6342/NTU202404155

model's capability in classification tasks. The formula for calculating the F1-score is shown in Equation 3-5 to 3-7.

$$Precision = \frac{TP}{TP + FP} \tag{3-5}$$

$$Recall = \frac{TP}{TP + FN} \tag{3-6}$$

$$\text{F1 score} = 2 \times \frac{precision \times recall}{precision + recall} \tag{3-7}$$

Where TP (true positive) represents the number of samples correctly predicted as positive, FP (false positive) represents the number of samples incorrectly predicted as positive, and FN (false negative) represents the number of samples incorrectly predicted as negative. The F1-score ranges from 0 to 1, with a value closer to 1 indicating better classification performance.

The F1-score comprehensively considers both precision and recall, preventing the oversight of the model's performance across different classes that might occur when using a single metric. For multi-class classification models, the formula is adjusted as shown in Equation 3-8. The F1-score for each class is calculated and then averaged by the number of classes N to obtain the macro average F1-score.

$$Macro\ F_1\ \text{score} = \frac{\sum_{n=1}^{N} F_1 - score_n}{N} \tag{3-8}$$

doi:10.6342/NTU202404155

In addition to evaluating the performance of the models themselves, the object detection model was evaluated using mAP@.5, while the insect pest classification model used F1-score. This study utilized McNemar's statistical test to examine whether the proposed models and methods significantly improved upon the original models (Wang et al., 2022). This method compares the predictions of two classifiers on the same samples, establishing a 2x2 contingency table, as shown in Table 3-2.

Table 3-2. Contingency table of McNemar's test.

	Model 1 correct	Model 1 incorrect
Model 2 correct	A	B
Model 2 incorrect	C	D

Where A represents the number of samples where both models predicted correctly; B represents the number of samples where the model 1 predicted incorrectly but the model 2 predicted correctly; C represents the number of samples where the model 1 predicted correctly but the model 2 predicted incorrectly; and D represents the number of samples where both models predicted incorrectly.

According to the contingency table above, these results can be divided into two categories of pairings: consistent (A, D) and inconsistent (B, C). When the null hypothesis H_0 is true (i.e., there is no significant difference between the two models),

doi:10.6342/NTU202404155

we expect $A + B = A + C$ or $B + D = C + D$, which is equivalent to $B = C$. As a result, assuming H_0, the numbers in B and C should be evenly distributed between the two categories. A chi-square test can be performed to determine whether the numbers in B and C are representative of the predicted distribution. This statistical procedure is referred to as the McNemar test. When doing the McNemar test, the chi-square test statistic is computed using Equation 3-9:

$$\chi^2 = \frac{(B - C)^2}{B + C} \tag{3-9}$$

After calculating the chi-square test statistic, utilize it with one degree of freedom to get the P-value. Compare this P-value to the significance level of 0.05. If the P-value is more than 0.05, we cannot reject the null hypothesis H_0, implying that there is no significant difference between the two models. On the other hand, if the P-value is less than 0.05, we reject the null hypothesis H_0, indicating a substantial difference between the models. We use the McNemar test to assess the performance differences between models and determine whether they are statistically significant, which provides a basis for selecting whether to switch models.

doi:10.6342/NTU202404155

3.7 System status front-end

To facilitate easier monitoring of equipment anomalies in each field and increase equipment availability, we utilized an MQTT broker as an intermediary between field devices and the server. Additionally, we designed a website to display the operational status of the equipment. The user interfaces allow for quick and clear visualization of equipment performance, eliminating the need to wait for the entire model training process to complete. By designing a log file, we greatly decreased the loading time, avoiding the need to load all the data from the server. This section discusses the design and development of the system status front-end.

3.7.1 MQTT broker

In the original setup, each device transmitted data directly to the server. However, since the Raspberry Pi's IP address is randomly assigned upon reboot, packet loss issues arose. Continuous listening also increased the server's operational load. To address these issues, we introduced an MQTT broker in the communication protocol, acting as a bridge between field devices and the server, as shown in Fig. 3-19.

doi:10.6342/NTU202404155

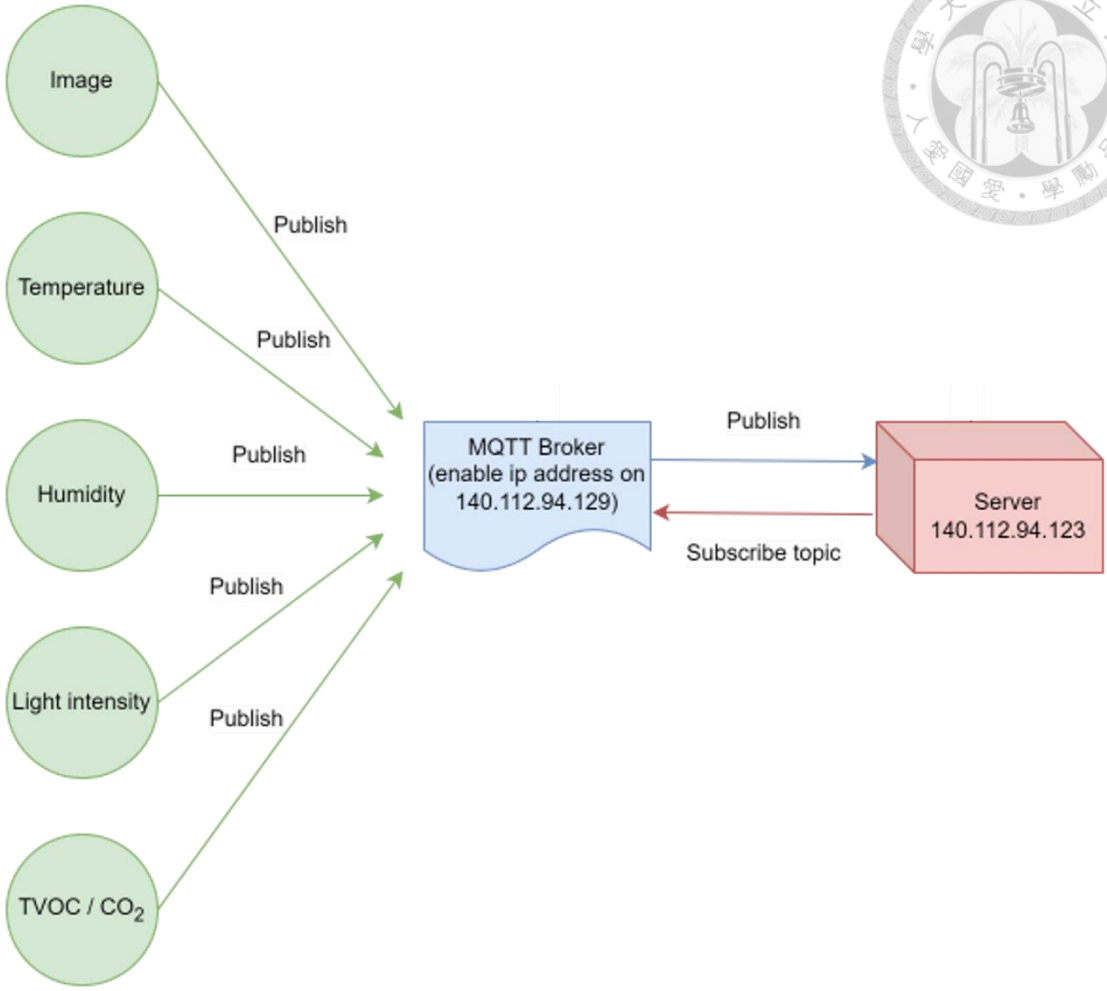


Fig. 3-19. **Illustration of MQTT broker setup.** This figure demonstrates the configuration of the MQTT broker, detailing its role as an intermediary between field devices and the server, and highlighting the data flow and communication pathways.

The broker's functionality is divided into two parts:

1. Subscriber Functionality: The broker acts as a subscriber, subscribing to the devices in different locations to obtain raw data. The field devices act as publishers, responsible for publishing information to the subscriber. This approach reduces the server's burden by delegating data collection and transmission tasks to the broker.

doi:10.6342/NTU202404155

2. Publisher Functionality: The broker acts as a publisher, and the server acts as a subscriber, subscribing to the broker to obtain the collected data. The advantage of this setup is that the server only needs to provide its IP address to the broker. By fixing the broker's IP address, the issue of devices being randomly assigned IP addresses upon reboot is eliminated.

3.7.2 Website

To obtain real-time operational status of the devices, we implemented an additional log file within the broker. This log file records the field, device number, and IP address reported by each device, effectively functioning as a roll call system. This setup facilitates monitoring device health, ensuring proper operation, and checking for any IP address conflicts. Additionally, we utilized AWS Quicksight to design a webpage that displays the roll call system, providing a clear and immediate overview of device status. The specific workflow is illustrated in Fig. 3-20, and the interface of AWS Quicksight is shown in Fig. 3-21.

Fig. 3-20. **Workflow of MQTT Broker and AWS Website.** This figure illustrates the integration of the MQTT broker with the AWS website.

doi:10.6342/NTU202404155

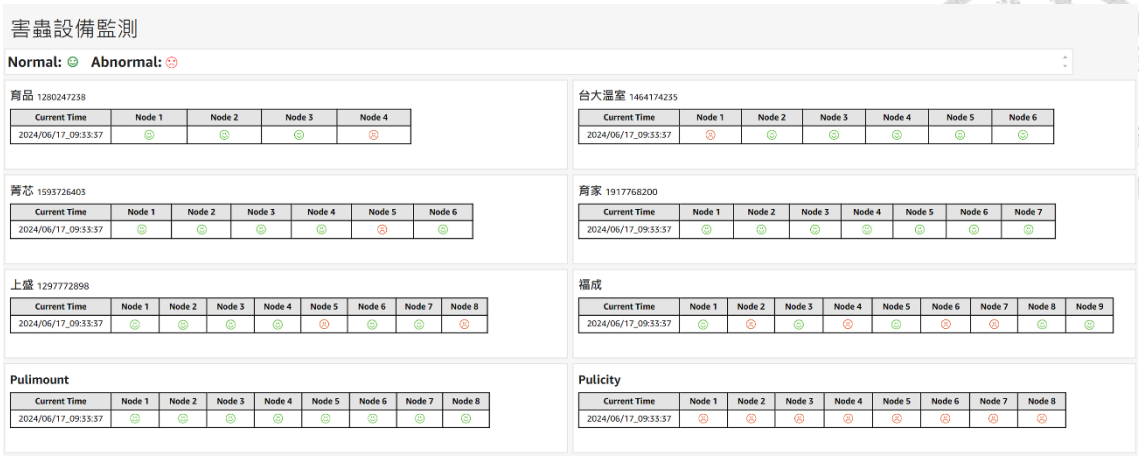


Fig. 3-21. **Real-time Device Monitoring AWS Webpage.** This figure shows the AWS webpage for real-time device monitoring, utilizing AWS IoT Core to collect data sent from servers and display it using Quicksight on the webpage.

doi:10.6342/NTU202404155

CHAPTER 4

Results and discussion

4.1 Sticky paper image curation method

Each wireless sensor network device deployed in the field captures images every day between 7 a.m. and 5 p.m., generating two images per hour at 00 and 05 minutes, resulting in a total of 22 images per day. However, environmental factors such as dust, water droplets, glare, or equipment instability, such as loose camera cables or device malfunctions, may occasionally affect image quality, resulting in poor-quality images. Failure to filter out these problematic images can adversely affect the performance of subsequently trained models. Therefore, the initial step involves image curation obtained from the devices. This work proposes two image curation methods: filtering based on the background of captured images and evaluating the degree of blurriness. According to the preliminary work, we observed that the captured images can be roughly categorized into normal images with a single background color and abnormal images with multiple background colors, as shown in Fig. 4.1.

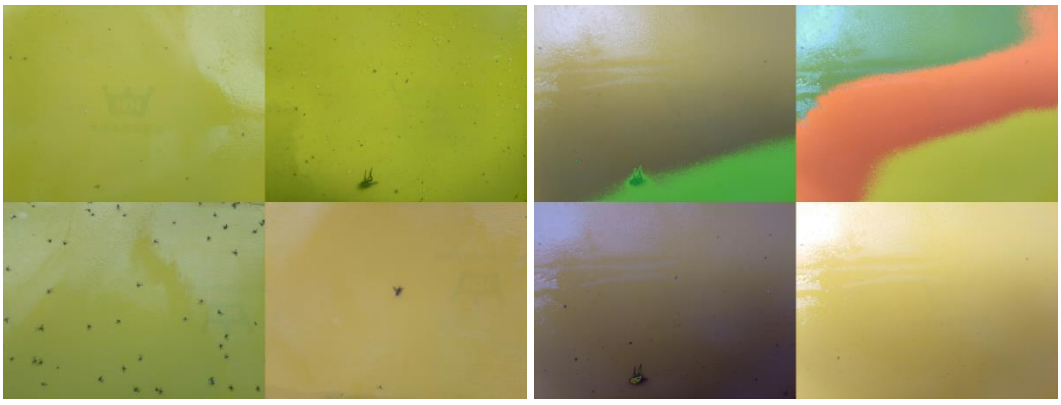


Fig. 4-1. Examples of normal and abnormal image patterns.

doi:10.6342/NTU202404155

4.1.1 Unsupervised k-means clustering

After understanding the patterns of normal and abnormal images, we preliminarily classify the images into single-color backgrounds and multi-color backgrounds by calculating the standard deviation of the HSV color channels. We then use unsupervised k-means clustering to automatically find the classification threshold without human intervention, as shown in Fig. 4-2. This method effectively divides the images into two categories: those with lower values on the left are normal images, while those with higher values on the right are abnormal images. Similarly, for image sharpness, we use this method to determine the Laplace variance threshold, which allows us to distinguish between clear and blurry images. In each experimental field, approximately 10% to 20% of the abnormal images are filtered out at this stage.

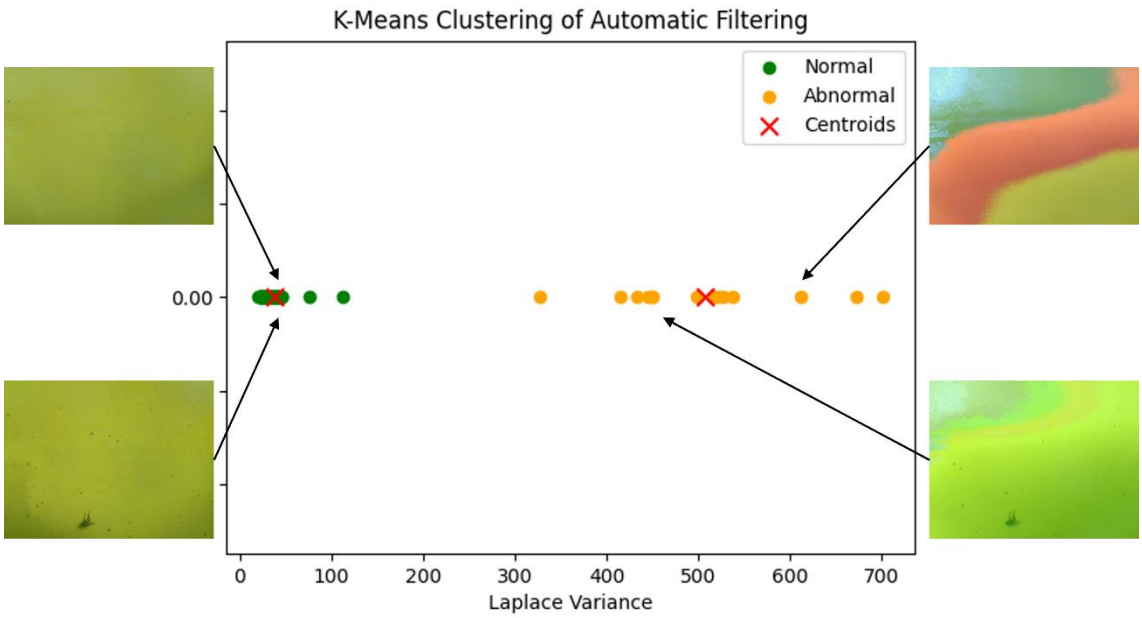


Fig. 4-2. Result of unsupervised k-means clustering.

doi:10.6342/NTU202404155

We applied the experimental results from the two image curation methods to all experimental sites to ensure robustness and generalizability. Fig. 4-3 depicts the results of six independent experimental sites. Each site employed unsupervised k-means clustering to categorize the images as normal or abnormal. The unsupervised k-means clustering algorithm successfully discriminated between the two categories using the Laplace variance. Lower values correspond to regular images with a single background color, showing consistent and predictable situations. Higher numbers, on the other hand, correspond to anomalous images with variable backdrop colors, which may represent anomalies or impediments. This consistent differentiation across multiple sites demonstrates the effectiveness of the curation and clustering method for detecting and categorizing image anomalies.

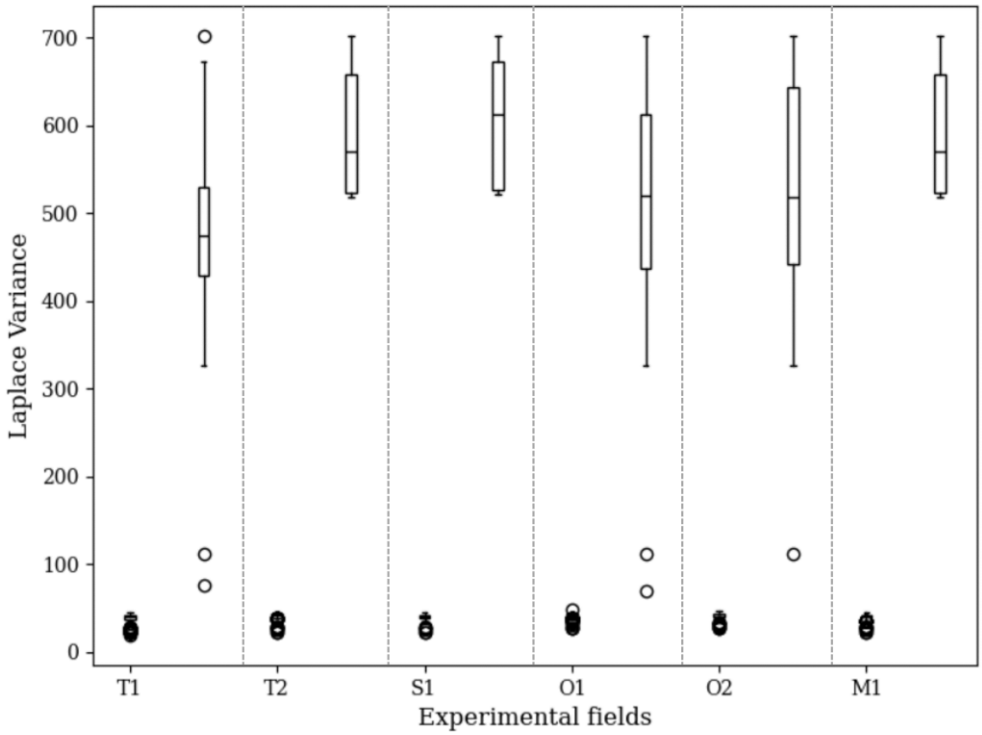


Fig. 4-3. Results obtained from k-means clustering in six experimental fields.

doi:10.6342/NTU202404155

4.1.2 Image curation results on six experimental sites

After filtering out regular and aberrant images, we tested the object detection model to evaluate its overall performance. Table 4-1 and 4-2 present the results, with the model quality assessed using the mAP@.5 metric. Both methods used the same number of images for comparison. The mAP@.5 improved from below 0.6 to over 0.95 after applying image curation methods, demonstrating a significant enhancement in the object detection model's performance.

Table 4-1. YOLOv7 training results of six experimental fields without image curation.

Field	Precision	Recall	mAP@.5	mAP@.5:.95
T1	0.594	0.475	0.465	0.189
T2	0.629	0.480	0.523	0.208
S1	0.584	0.498	0.462	0.187
O1	0.720	0.612	0.597	0.263
O2	0.692	0.590	0.557	0.247
M1	0.812	0.535	0.594	0.262

doi:10.6342/NTU202404155

Table 4-2. YOLOv7 training results of six experimental fields with image curation.

Field	Precision	Recall	mAP@.5	mAP@.5:.95
T1	0.973	0.983	0.984	0.560
T2	0.951	0.915	0.949	0.578
S1	0.963	0.953	0.971	0.608
O1	0.957	0.975	0.986	0.651
O2	0.986	0.982	0.985	0.667
M1	0.949	0.912	0.950	0.582

4.2 Object detection model development

After image curation, we obtained a high-quality image dataset for subsequent model training. The previous object detection model was trained several years ago, and in recent years, we have added several new experimental sites. Considering that new crops have different insect pests and the potential for future development of new experimental sites, we decided to update the object detection model. We employed a semi-supervised learning approach, allowing the model to retrain periodically. This continuous updating process aims to improve the model's accuracy and generalization capabilities over time.

doi:10.6342/NTU202404155

4.2.1 YOLOv3-Tiny vs. YOLOv7

First, we evaluated the performance of the existing and revised models on preprocessed images. The previous model used the YOLOv3-Tiny architecture, which is a lightweight object detection model. In this study, we replaced it with YOLOv7, which has additional layers and includes multi-head attention layers. These modifications aid in distinguishing between foreground and background items, allowing the model to focus on the areas of greatest interest. Training the YOLOv7 object identification model takes about 3 hours, which is a reasonable time frame. Table 4-3 and 4-4 show the results of both models for the same dataset. The findings show that YOLOv3-Tiny is no longer appropriate for training on new experimental sites, resulting in poor outcomes in further training. In comparison, YOLOv7 consistently achieved a mAP@.5 score of over 0.95 across the board. For McNemar's statistical test, the corresponding P-value $= 5.76 \times 10^{-5}$, reject the null hypothesis H_0, there is a significant difference between the YOLOv3-Tiny and YOLOv7 models.

Table 4-3. YOLOv3-Tiny training results of six experimental fields.

Field	Precision	Recall	mAP@.5	mAP@.5:.95
T1	0.440	0.137	0.115	0.037
T2	0.074	0.032	0.011	0.003
S1	0.484	0.200	0.127	0.026
O1	0.720	0.744	0.635	0.208
O2	0.592	0.490	0.357	0.087
M1	0.712	0.135	0.149	0.051

doi:10.6342/NTU202404155

Table 4-4. YOLOv7 training results of six experimental fields.

Field	Precision	Recall	mAP@.5	mAP@.5:.95
T1	0.973	0.983	0.984	0.560
T2	0.951	0.915	0.949	0.578
S1	0.963	0.953	0.971	0.608
O1	0.957	0.975	0.986	0.651
O2	0.986	0.982	0.985	0.667
M1	0.949	0.912	0.950	0.582

4.2.2 YOLOv7 training results of six experimental sites

Following the completion of image curation, a set of high-quality images was obtained for subsequent training of the object detection model. In real-time monitoring applications, factors such as model training duration and the ability to achieve robust performance with limited data samples are equally critical. To address these problems, we first collected image datasets of different sizes from six experimental domains, totaling 250, 500, 750, and 1000 images, respectively. These datasets provided the foundation for training the object detection model. Table 4-5 presents and summarizes the training process's results for a full assessment. At this stage of the model development, the emphasis is placed on a single object class. The evaluation of the model performance is conducted using mAP metric, which provides a comprehensive measure of the model's accuracy in detecting objects within the images.

doi:10.6342/NTU202404155

Table 4-5. YOLOv7 training results across six fields with different number of images.

Field	Number of images	Precision	Recall	mAP@.5	mAP@.5:.95
T1	250	0.810	0.718	0.711	0.234
	500	**0.951**	**0.915**	**0.949**	**0.578**
	750	0.957	0.912	0.943	0.582
	1000	0.960	0.928	0.958	0.588
T2	250	0.957	0.739	0.816	0.376
	500	**0.973**	**0.983**	**0.984**	**0.560**
	750	0.975	0.986	0.985	0.554
	1000	0.976	0.982	0.985	0.558
S1	250	0.901	0.837	0.849	0.390
	500	**0.963**	**0.953**	**0.971**	**0.608**
	750	0.969	0.957	0.972	0.613
	1000	0.971	0.964	0.979	0.626
O1	250	0.928	0.963	0.965	0.531
	500	**0.957**	**0.975**	**0.986**	**0.651**
	750	0.962	0.979	0.987	0.663
	1000	0.961	0.985	0.991	0.698
O2	250	0.950	0.935	0.925	0.433
	500	**0.986**	**0.982**	**0.985**	**0.667**
	750	0.995	0.994	0.995	0.693
	1000	0.997	0.995	0.996	0.698
M1	250	0.859	0.832	0.830	0.378
	500	**0.949**	**0.912**	**0.950**	**0.582**
	750	0.957	0.915	0.951	0.575
	1000	0.960	0.928	0.963	0.588

doi:10.6342/NTU202404155

4.2.3 YOLOv7 training results of sites combination

After training the object detection model, it became clear that using 500 images per field resulted in mAP greater than 0.95 at an IoU threshold of 0.5. Despite efforts to increase the dataset size to 750 and 1000 images, there was no significant improvement in the model's accuracy. Subsequently, to improve the model's accuracy and efficiency, we combined the 500 images from each field into a bigger dataset of 3000 images. The additional dataset was then used to retrain the YOLOv7 model. However, despite changes to several model hyperparameters, such as batch sizes of 8, 16, 24, and 32, no improvement in model performance was detected when compared to the performance on each experimental field. Detailed results are provided in Table 4-6.

Table 4-6. YOLOv7 training results for the 3000-image dataset.

Batch size	Precision	Recall	mAP@.5	mAP@.5:.95
8	0.703	0.666	0.661	0.256
16	0.739	0.684	0.709	0.294
24	0.752	0.679	0.707	0.289
32	0.808	0.724	0.775	0.358

4.2.4 YOLOv7 training results of different crop types

The substantial variability in field conditions resulting from different crops leads to poor overall performance. Therefore, we divided the six fields into three categories based on crop types: two fields for tomato seedlings, two for orchids, and two for fruits,

doi:10.6342/NTU202404155

respectively. Each category underwent individual model training, resulting in all categories achieving mAP exceeding 0.95 at an IoU threshold of 0.5. To validate these findings, we also tested the model with images from tomato seedlings and orchid. However, the model's performance significantly declined, achieving mAP less than 0.8 at an IoU threshold of 0.5. Detailed training results are shown in Table 4-7.

Table 4-7. YOLOv7 training results based on crop types.

Crop types	Precision	Recall	mAP@.5	mAP@.5:.95
T1+T2	0.957	0.946	0.952	0.576
O1+O2	0.960	0.951	0.976	0.654
S1+M1	0.962	0.928	0.963	0.588
T1+O1	0.794	0.734	0.744	0.301
T2+O2	0.804	0.771	0.782	0.336

4.3 Insect pest classification model development

After annotating the target objects in the YOLOv7 object detection model, we proceeded to train the insect pest classification model. This study compared three multi-class models: ResNet-18, Inception-V3, and Vision Transformer (ViT). Each of these models was trained on the same dataset consisting of eight common greenhouse micro-insect pest species. The dataset comprising these eight species, was obtained from the YOLOv7 model's output, where the images were cropped around the detected objects for further classification and analysis. The distribution of the dataset into training,

doi:10.6342/NTU202404155

validation, and testing sets at a ratio of 7:2:1 is illustrated in Table 4-8, and the data augmentation techniques includes flipping, rotating, color jittering, and Gaussian noise.

Table 4-8. Dataset for training the insect pests classification model.

Class	Dataset	Training	Validation	Testing
Cranefly	2000 (200) *	1400	400	200
Fly	2000 (1000) *	1400	400	200
Gnat	2000	1400	400	200
Midge	2000 (1000) *	1400	400	200
Mosquito	2000 (400) *	1400	400	200
Moth fly	2000 (1000) *	1400	400	200
Thrips	2000	1400	400	200
Whitefly	2000	1400	400	200
Others	2000 (1000) *	1400	400	200

* Utilize data augmentation techniques to address the limited quantity of the dataset.

4.3.1 Classification model comparison

Fig. 4-4 depicts the training and validation accuracies, as well as the associated losses, for these models. ResNet-18 stands out among the three cutting-edge models because it has fewer layers and a lighter design, making it perfect for the multi-class classification task addressed in this study. In contrast, Inception-V3 sets a size constraint on the input image, requiring it to be 299x299 pixels. However, given that the average size of micro-insects is only a few dozen pixels, scaling up the images by such

doi:10.6342/NTU202404155

a large amount can hinder the model's ability to learn image features efficiently. As a result, Inception-V3 is less accurate and has a higher loss than the other two models. So ResNet-18's performance in this multi-class task is sufficient.

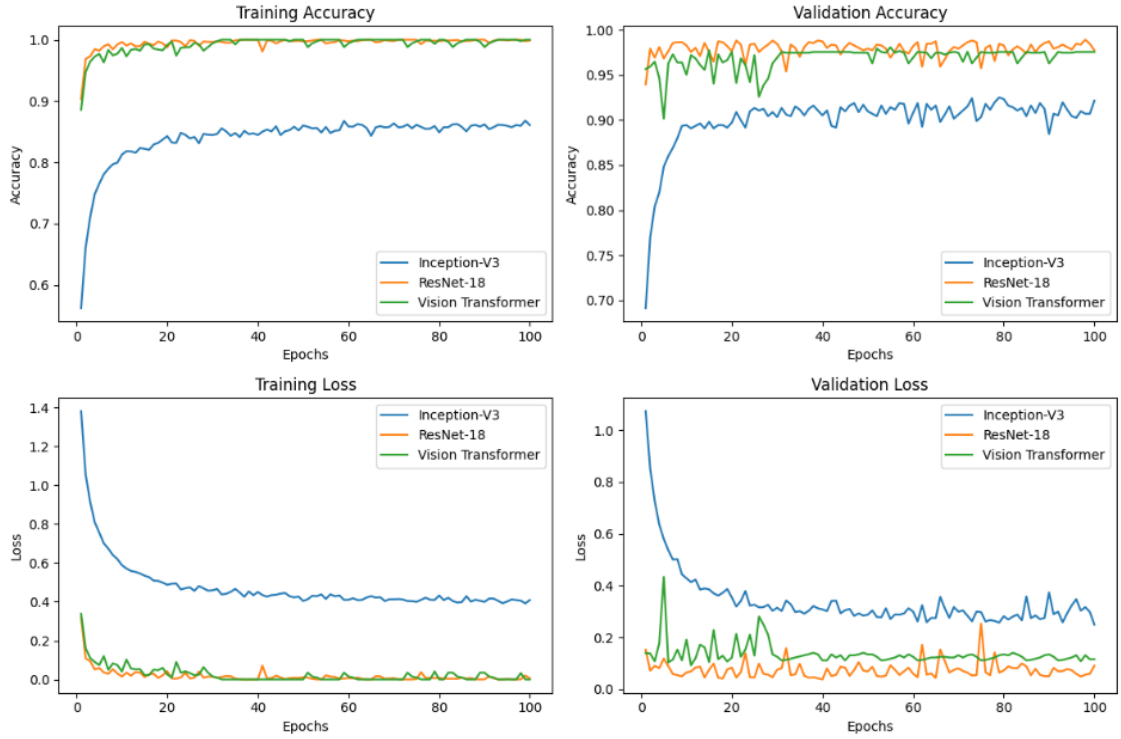


Fig. 4-4. Training and validation accuracy and loss of insect pest classification models.

The F1-score evaluation of the models used a testing dataset separate from the training dataset. From Fig. 4-5 to 4-7, it is obvious that ResNet-18 and Vision Transformer achieved macro F1-scores of 0.988 and 0.983, respectively. However, as previously indicated, Inception-V3's overall performance did not meet expectations due to difficulties with feature extraction. Furthermore, ResNet-18 completed training in under 30 minutes, whereas ViT and Inception-V3 took up to 180 minutes. ResNet-18 was chosen as the micro-insect pest classification model due to its low training time and high performance.

doi:10.6342/NTU202404155

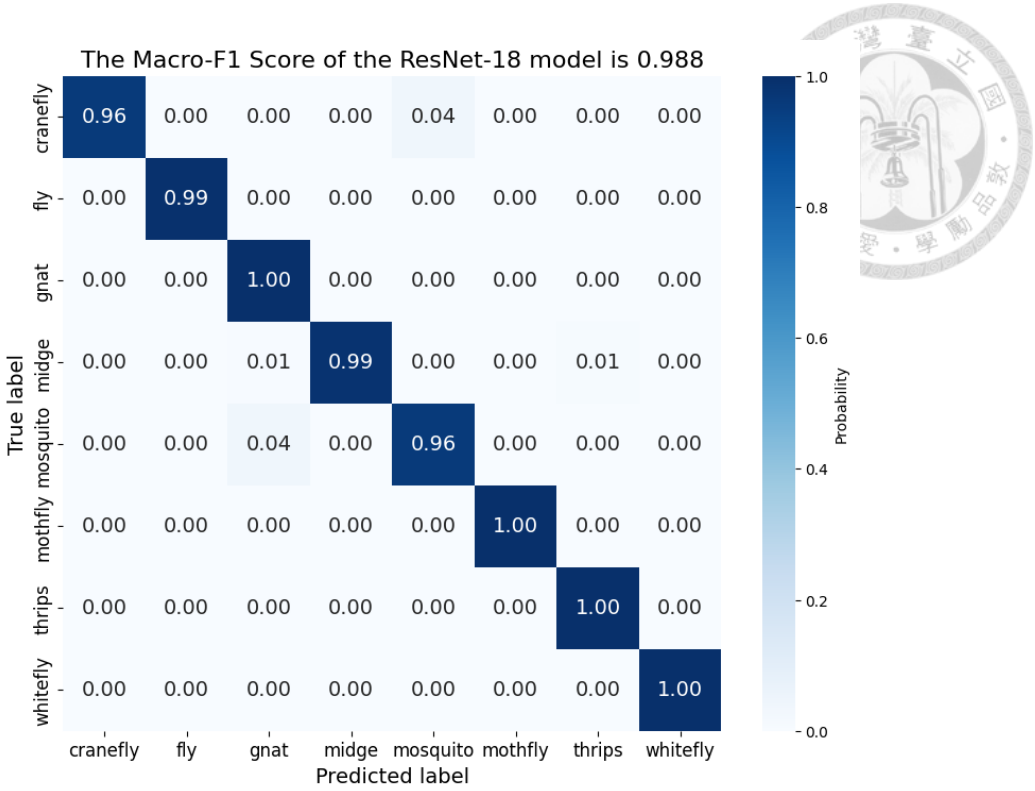


Fig. 4-5. Confusion matrix for the testing results of the ResNet-18 model.

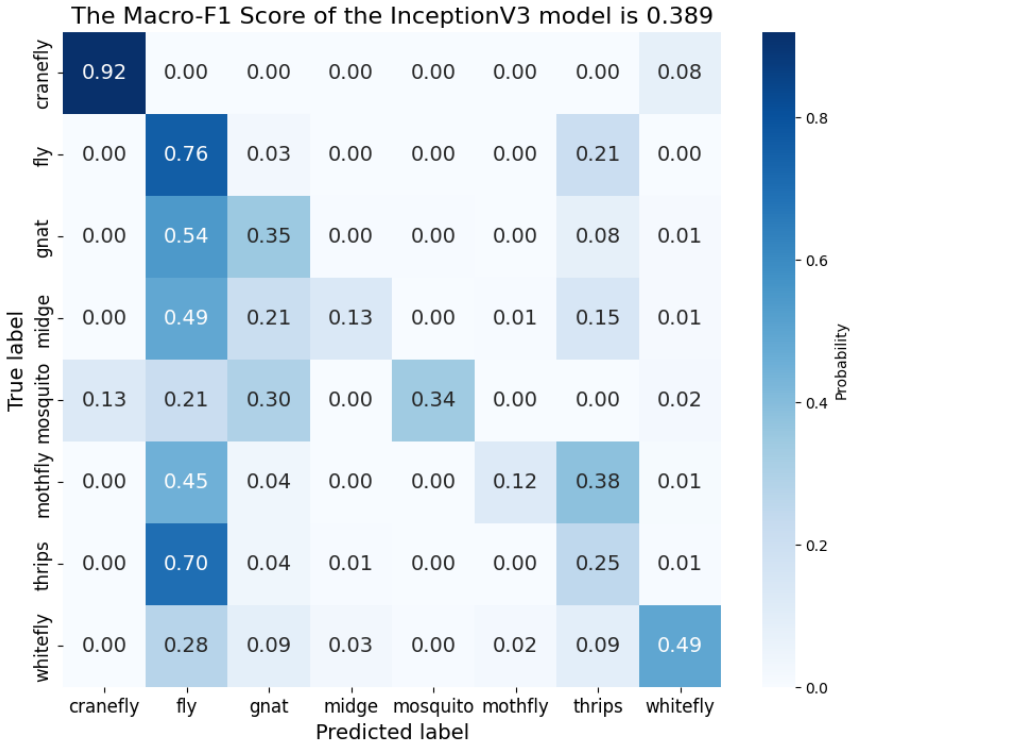


Fig. 4-6. Confusion matrix for the testing results of the Inception-V3 model.

doi:10.6342/NTU202404155

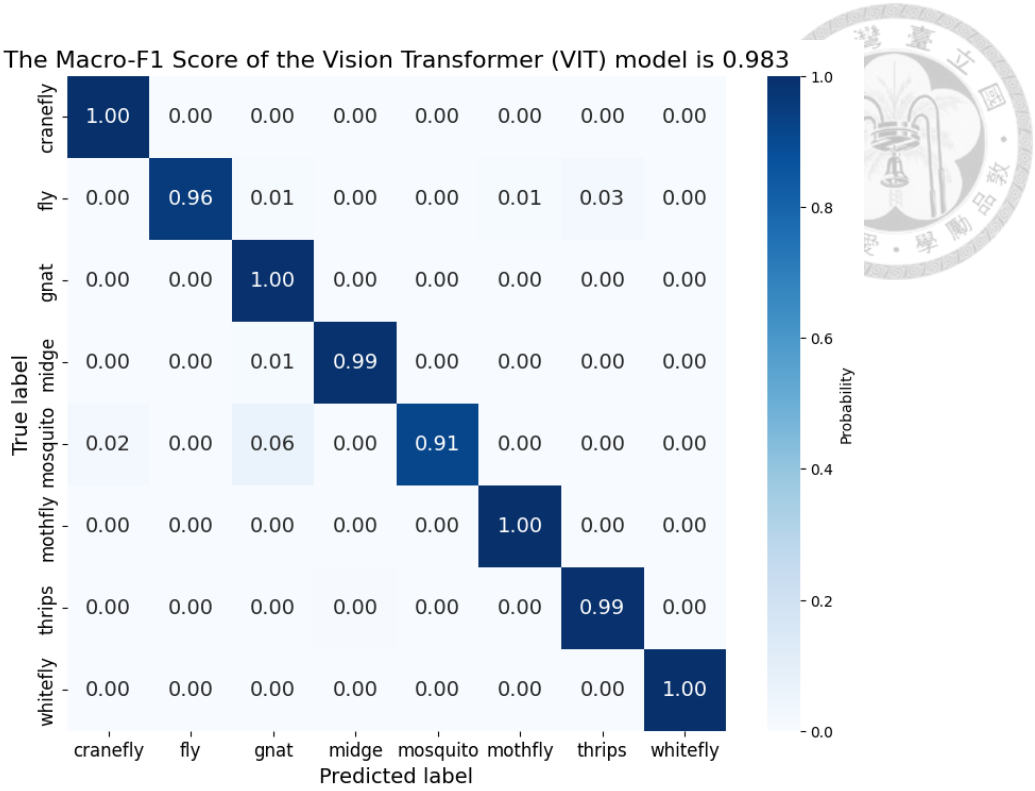


Fig. 4-7. Confusion matrix for the testing results of the Vision Transformer model.

4.3.2 Results of data augmentation

After completing the object identification model training, we discovered that the prior training dataset was too small and had an imbalanced distribution, resulting in poor performance in some categories during actual use. Fig. 4-8 show the confusion matrices for the initial model and the model after data augmentation. The left matrix indicates that the original model performed poorly for fly, mosquito, and thrips, frequently misclassifying them as gnats and midges. In comparison, the right matrix shows that, following data augmentation, all categories performed much better, with the exception of probable misclassifications between fly and gnat. The training and validation details of the model are illustrated in Fig. 4-9. Overall, there were no signs of

doi:10.6342/NTU202404155

overfitting. For McNemar's statistical test, the corresponding P-value $= 7.8 \times 10^{-3}$, reject the null hypothesis H_0, there is a significant difference of data augmentation.

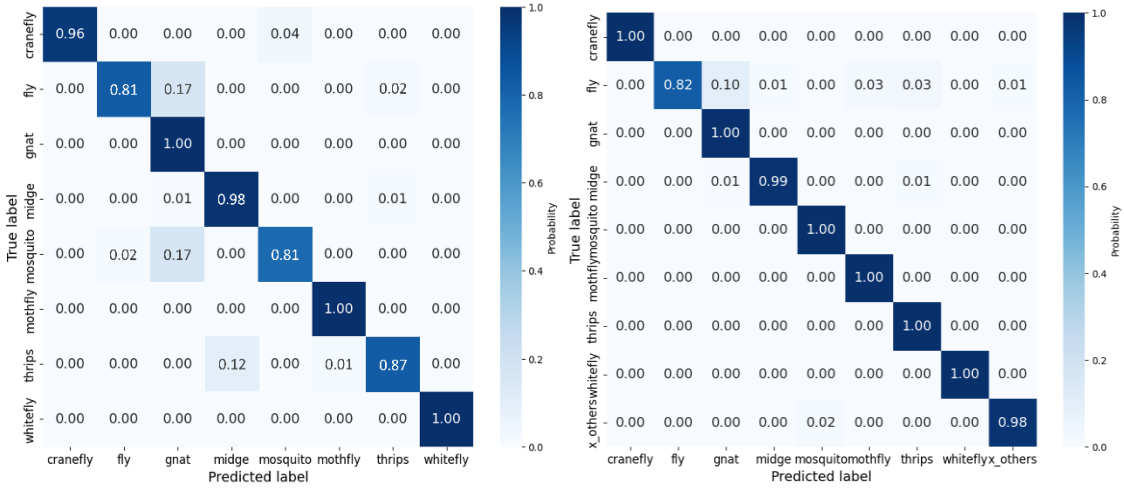


Fig. 4-8. Confusion matrix for the testing results of (a) without data augmentation and (b) with data augmentation and others category.

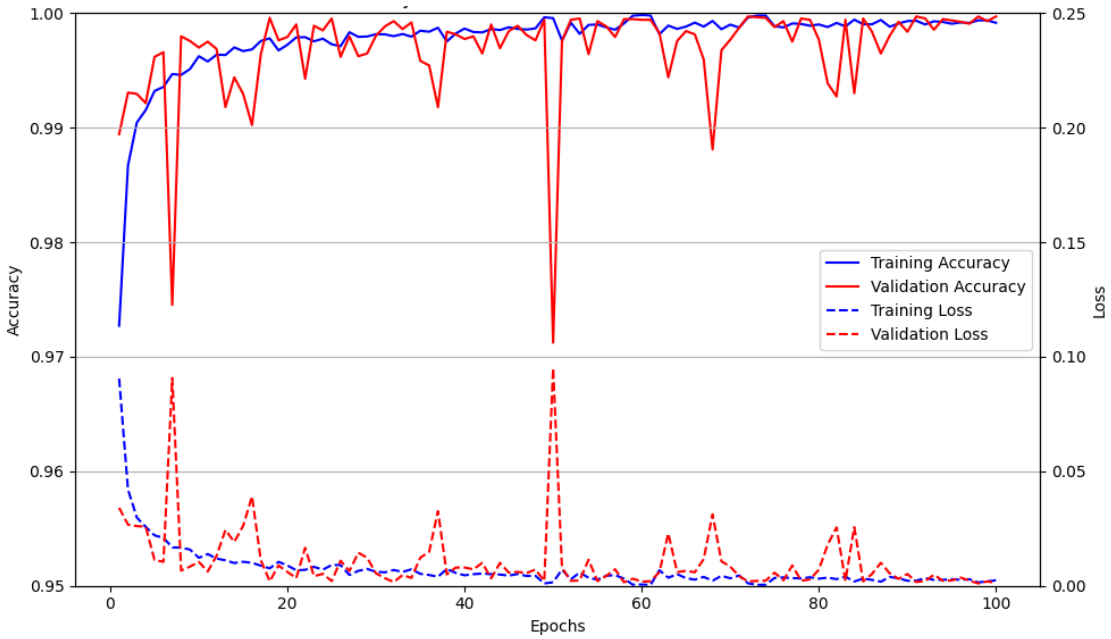


Fig. 4-9. Training and validation accuracy and loss of data augmentation model.

doi:10.6342/NTU202404155

4.3.3 Results of size feature extraction

In the previous part, we enhanced the model's overall performance by augmenting the data and adding the "others" category. The model, however, struggled with the fly and gnat categories because to their physical similarities at some angles, despite large size differences. This frequently resulted in flies being misdiagnosed as gnats. To solve this issue, we developed an OTSU classifier tailored specifically for these two species. Fig. 4-10 compares the results without and with size feature extraction. It is clear that all categories obtained good accuracy with size feature extraction. Furthermore, as illustrated in Fig. 4-11, the training procedure did not display overfitting. For McNemar's statistical test, the corresponding P-value $= 1.2 \times 10^{-3}$, reject the null hypothesis H_0, there is a significant difference of size feature extraction.

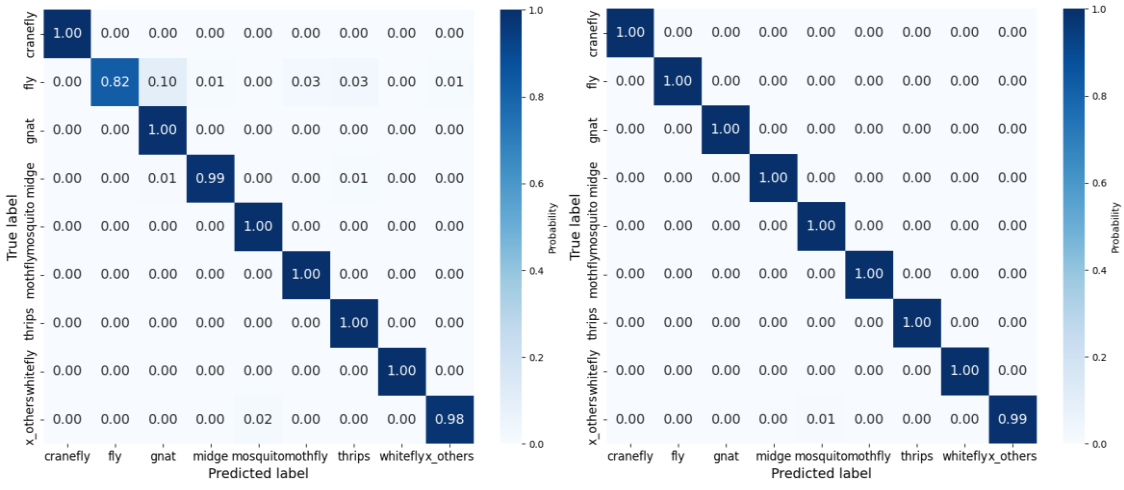


Fig. 4-10. Confusion matrix for the testing results of (a) without size feature extraction and (b) with size feature extraction.

doi:10.6342/NTU202404155

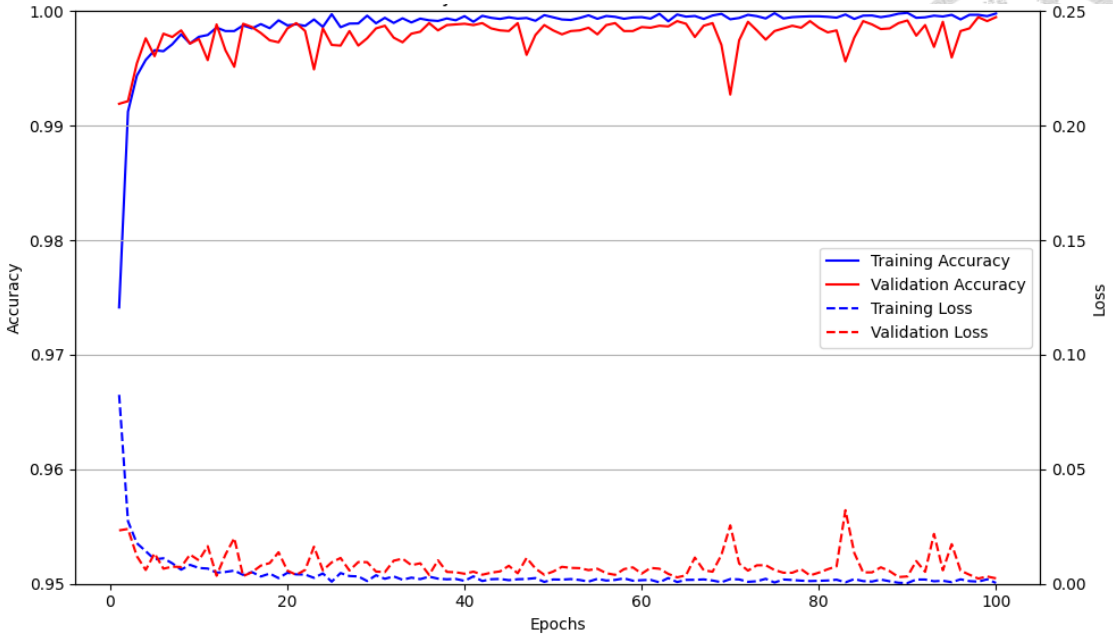


Fig. 4-11. Training and validation accuracy and loss of size feature model.

4.4 Spatiotemporal algorithm analysis

Following the training of the object detection and insect pest classification model, a testing phase was initiated by processing a dataset over a specific time period. The results obtained from the model training were observed and used to correct the types and quantities of insect pests. Fig. 4-12 to 4-17 illustrate the raw data visualization across six experimental fields, highlighting how misclassifications and missing data lead to inaccurate statistical results. Subsequently, we conducted a detailed analysis of insect pest species and counts specifically for field T1.

doi:10.6342/NTU202404155

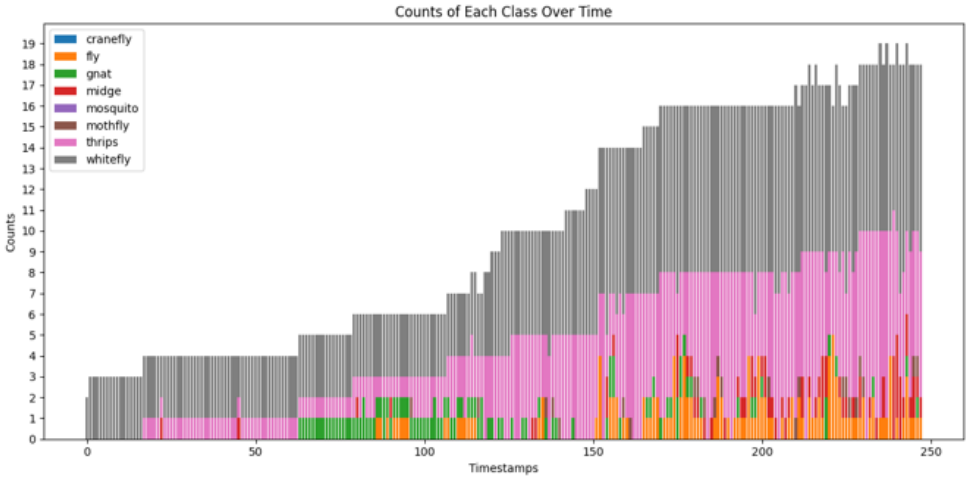


Fig. 4-12. Illustrations of raw data visualization on field T1.

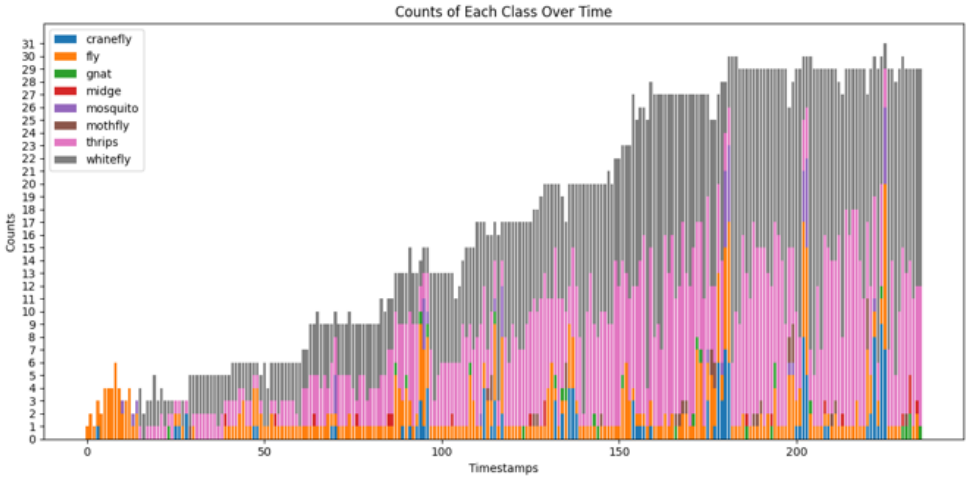


Fig. 4-13. Illustrations of raw data visualization on field T2.

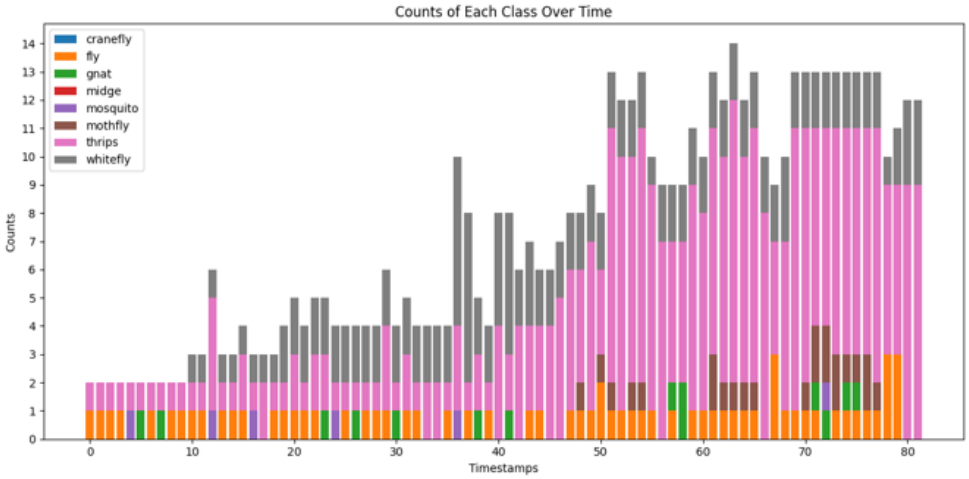


Fig. 4-14. Illustrations of raw data visualization on field S1.

doi:10.6342/NTU202404155

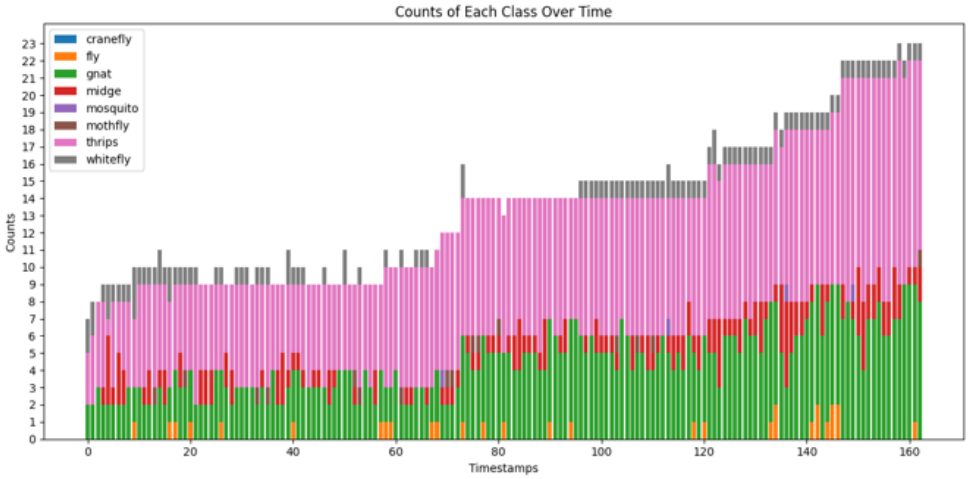


Fig. 4-15. Illustrations of raw data visualization on field O1.

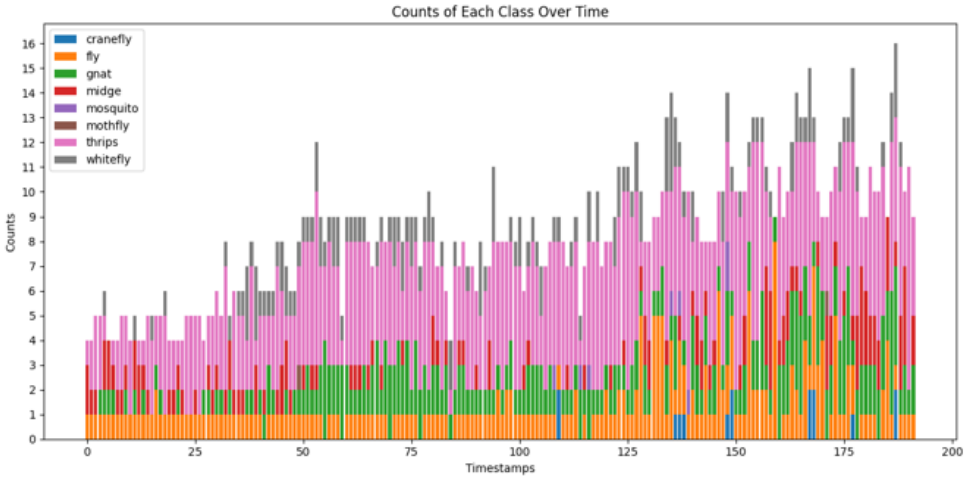


Fig. 4-16. Illustrations of raw data visualization on field O2.

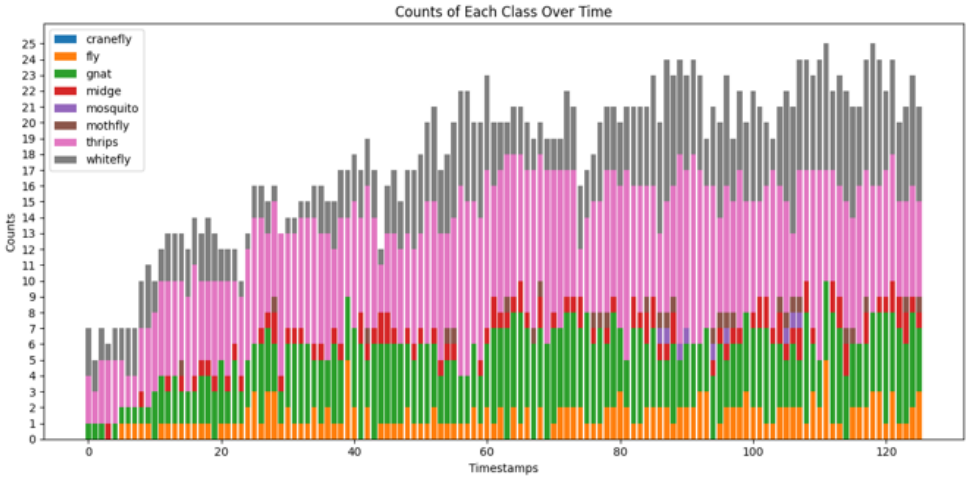


Fig. 4-17. Illustrations of raw data visualization on field M1.

doi:10.6342/NTU202404155

4.4.1 Raw data analysis and ground truth

Where timestamps along the x-axis represent hourly intervals, and the y-axis denotes the quantity of insect pests. Fig. 4-18 depicts the model's inference results from raw data on field T1, revealing significant noise due to partial misclassifications. In contrast, Fig. 4-19 presents the ground truth data obtained through manual verification of insect pest quantities on sticky paper traps.

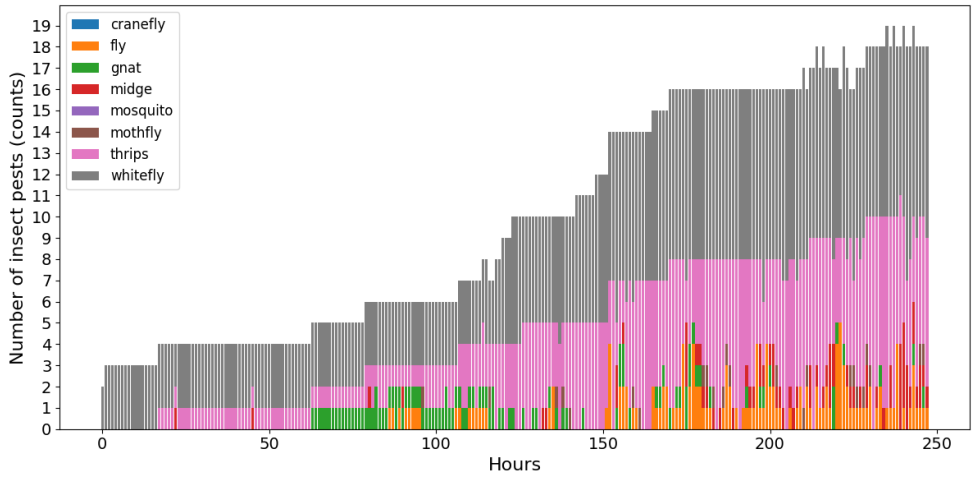


Fig. 4-18. Illustrations of raw data visualization on field T1. There are some misclassifications leads to incorrect statistical data.

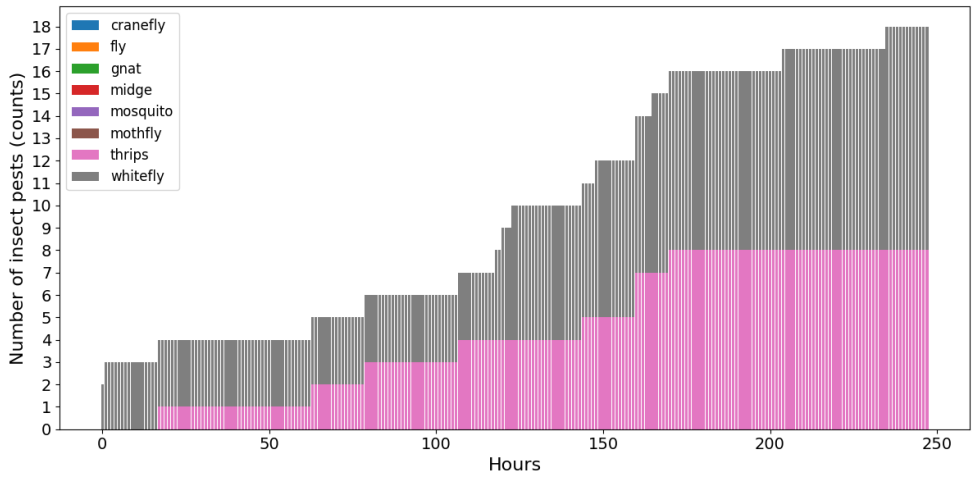


Fig. 4-19. Illustrations of ground truth representation on field T1, which is manually counting the insect pests on the sticky paper traps.

doi:10.6342/NTU202404155

4.4.2 Spatiotemporal algorithm

To align the raw data with the ground truth, we used a voting mechanism combined with spatiotemporal distribution for inference optimization. Initially, misclassifications were addressed by a voting method, in which uncovered insect pests were regarded misjudgments and were eventually excluded. This exclusion method, as shown in Fig. 4-20, retained only thrips and whitefly categories while eliminating other misclassified insect pests. However, these misclassifications originally referred to other insect infestations. As a result, it is critical to reverse these errors and classify items correctly. Furthermore, insect pest populations were found to be lower than at the prior time point. As a result, it is critical to restore both the types and amounts of insect pests that were misclassified. We next used the recorded spatial coordinates to infer temporal sequences before and after, as shown in Fig. 4-21. This successive mapping approach tries to return misclassified insect pests to their original proper categories.

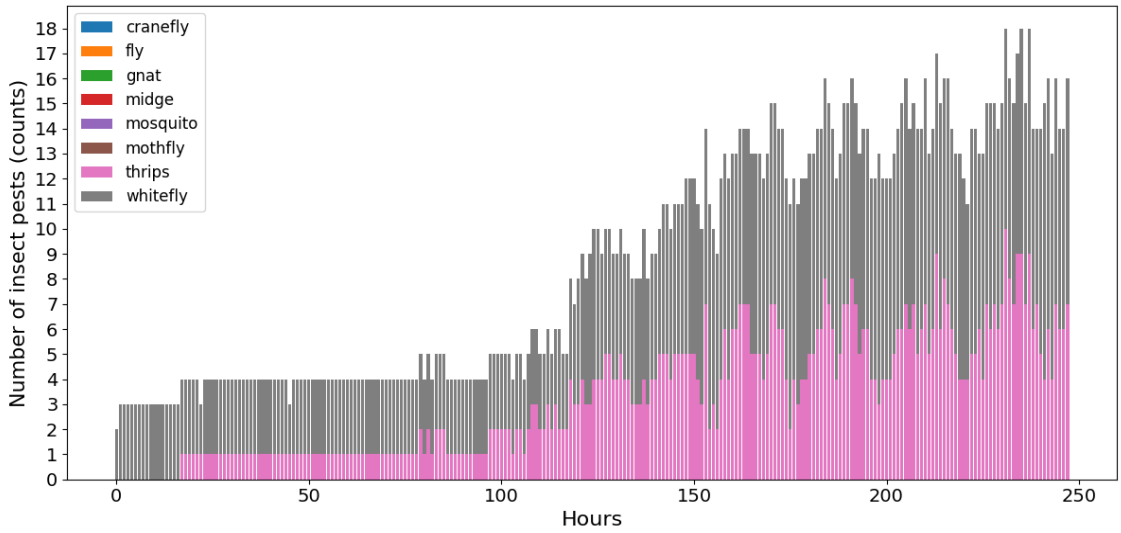


Fig. 4-20. Illustrations of voting method, resulting in several missing data on field T1.

doi:10.6342/NTU202404155

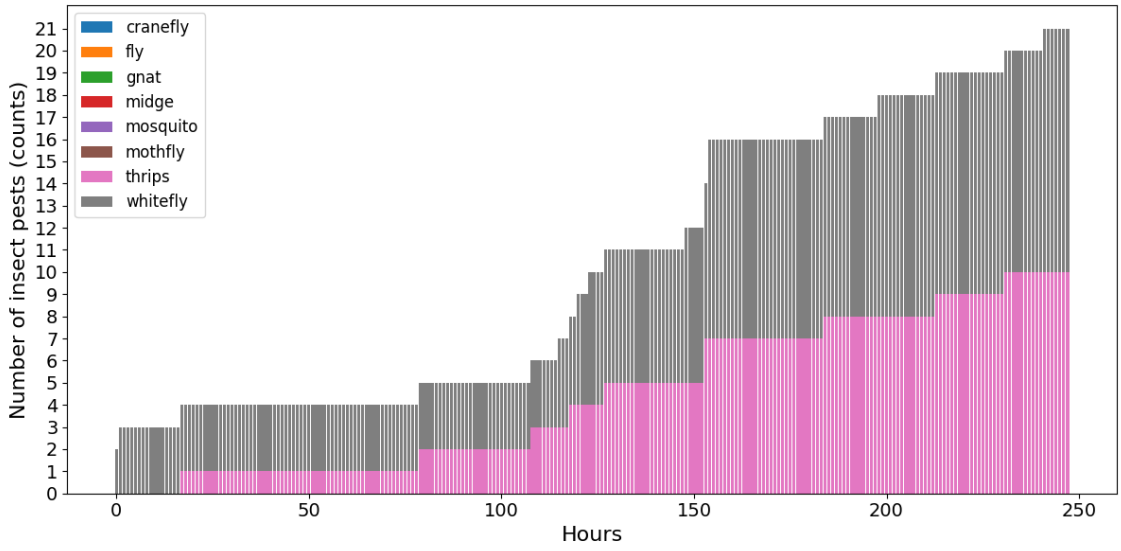


Fig. 4-21. Illustrations utilizing spatiotemporal methods to solve misclassification and missing data on field T1.

4.4.3 Model evaluation and data analysis

The evaluation process included comparisons between individual field runs, with an emphasis on common insect pest types and cumulative line charts. These comparisons were carried out iteratively across numerous occurrences, with the resulting averages calculated. Furthermore, a thorough analysis was conducted to investigate the differences between raw data, spatiotemporal data, and ground truth. MSE and MAPE were used as measures to quantify the average mistakes found in these various comparisons. Table 4-9 and 4-10 offer a complete summary of the data, explaining the efficacy and accuracy of the recommended approaches.

doi:10.6342/NTU202404155

Table 4-9. MSE and MAPE metrics were utilized to evaluate the performance of the whitefly across four cycles and cumulative counts on field T1.

Whitefly	Mean Square Error (MSE)	Mean Absolute Percentage Error (MAPE)
Cycle 1	12.047 (raw data) / 1.095 (spatiotemporal)	31.917% (raw data) / 6.615% (spatiotemporal)
Cycle 2	31.809 (raw data) / 0.476 (spatiotemporal)	33.446% (raw data) / 4.233% (spatiotemporal)
Cycle 3	14.238 (raw data) / 0.563 (spatiotemporal)	21.776% (raw data) / 5.001% (spatiotemporal)
Cycle 4	8.048 (raw data) / 0.332 (spatiotemporal)	18.399% (raw data) / 2.698% (spatiotemporal)
Cumulative counts	148.940 (raw data) / 8.607 (spatiotemporal)	26.182% (raw data) / 6.252% (spatiotemporal)

Table 4-10. MSE and MAPE metrics were utilized to evaluate the performance of the thrips across four cycles and cumulative counts on field T1.

Thrips	Mean Square Error (MSE)	Mean Absolute Percentage Error (MAPE)
Cycle 1	8.571 (raw data) / 0.238 (spatiotemporal)	13.140% (raw data) / 3.437% (spatiotemporal)
Cycle 2	14.286 (raw data) / 0.191 (spatiotemporal)	18.630% (raw data) / 3.241% (spatiotemporal)
Cycle 3	3.574 (raw data) / 0.476 (spatiotemporal)	25.221% (raw data) / 7.189% (spatiotemporal)
Cycle 4	1.905 (raw data) / 0.331 (spatiotemporal)	11.786% (raw data) / 6.264% (spatiotemporal)
Cumulative counts	5.036 (raw data) / 2.309 (spatiotemporal)	8.565% (raw data) / 4.262% (spatiotemporal)

doi:10.6342/NTU202404155

To conduct a thorough analysis, we calculated the MSE and MAPE for four cycles, focusing on whitefly and thrips. This analysis ran from October 1st to December 31st, for a total of three months. To identify differences, raw data and spatiotemporal analysis results were compared to the ground truth. Notably, whitefly cumulative MAPE dropped from 26.182% to 6.252%, while thrips fell from 8.565% to 4.262%. As seen in the table above, both the individual cycle and cumulative counts charts demonstrate a significant improvement in the MSE and MAPE metrics produced by spatiotemporal analysis when compared to raw data. Fig. 4-22 to 4-25 depict the data for whitefly and thrips, organized by cycle and cumulative counts, to provide a visual depiction of the observed trends and improvements over time.

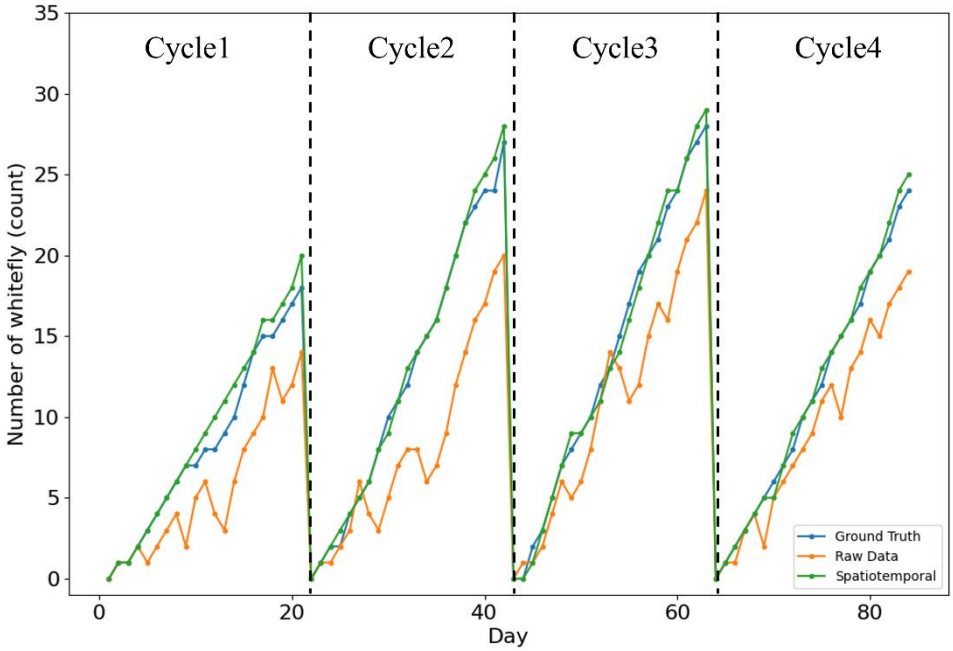


Fig. 4-22. Illustrations of whitefly counts compared to the ground truth for raw data and spatiotemporal methods on field T1.

doi:10.6342/NTU202404155

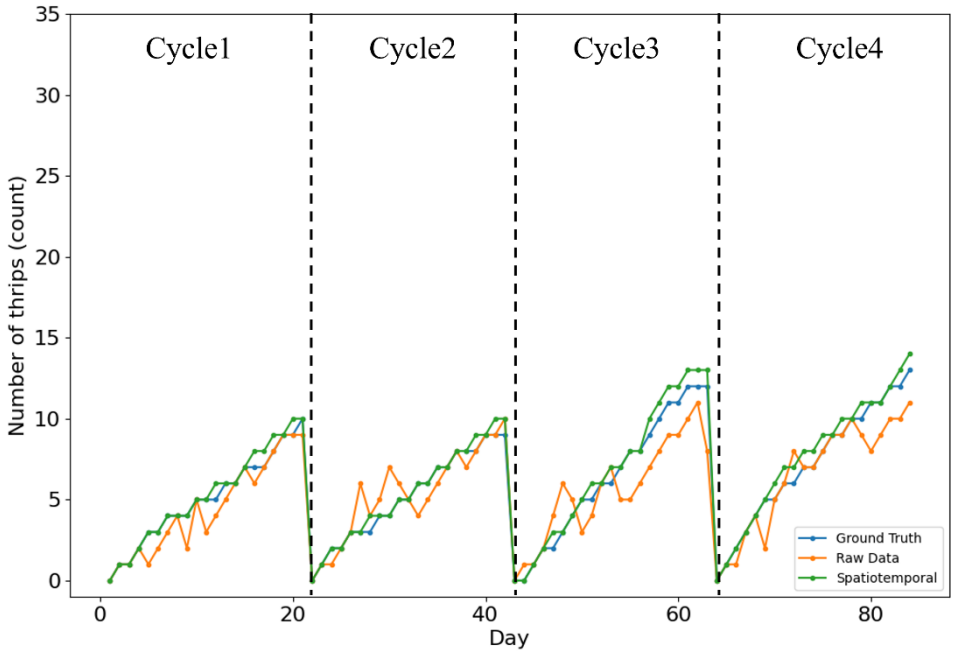


Fig. 4-23. Illustrations of thrips counts compared to the ground truth for raw data and spatiotemporal methods on field T1.

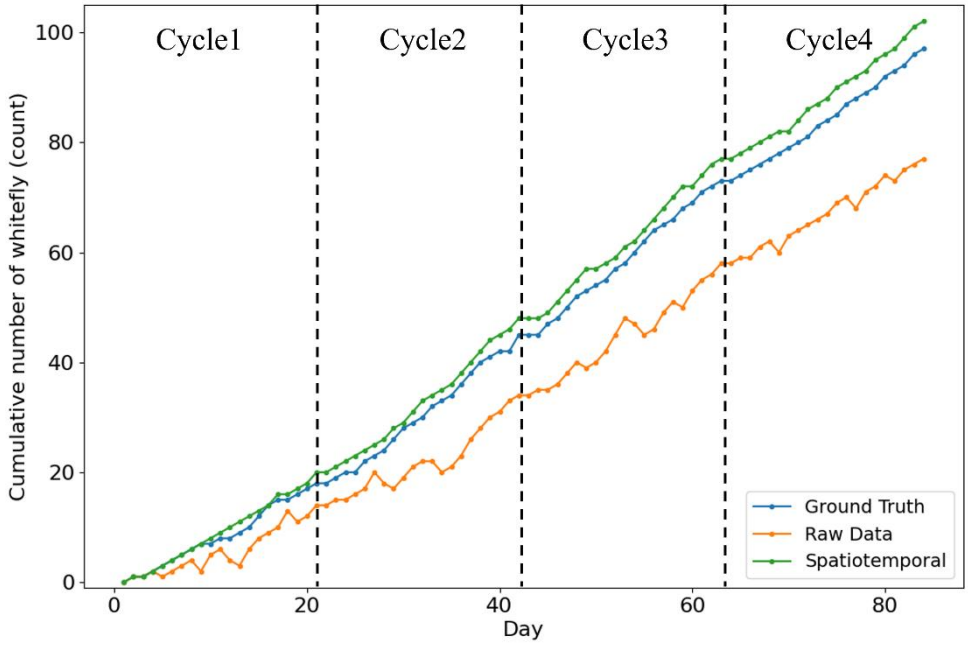


Fig. 4-24. Illustrations of cumulative whitefly counts compared to the ground truth for raw data and spatiotemporal methods on field T1.

doi:10.6342/NTU202404155

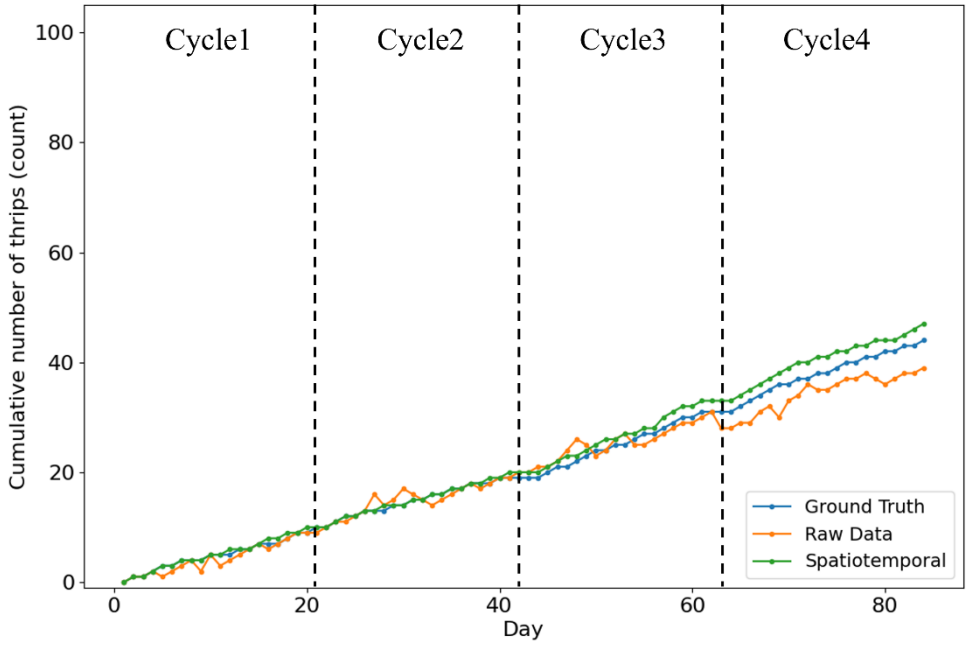


Fig. 4-25. Illustrations of cumulative thrips counts compared to the ground truth for raw data and spatiotemporal methods on field T1.

To expand from a single field's data over three months to all experimental fields, we used MSE and MAPE as indicators of overall data quality improvement. Table 7 shows the average MSE and MAPE for six experimental fields. Results show that the error percentages range from 4% to 5%, which is much lower than raw data without the voting system and spatiotemporal analysis. This demonstrates the success of the proposed approach in improving data quality, lowering misclassification rates, and obtaining data that is closer to the real field observations.

doi:10.6342/NTU202404155

Table 4-11. Average MSE and MAPE with six experimental fields.

Experimental fields	Mean Square Error (MSE)	Mean Absolute Percentage Error (MAPE)
T1	14.285 (raw data) / 1.988 (spatiotemporal)	17.374% (raw data) / 4.936% (spatiotemporal)
T2	14.238 (raw data) / 1.713 (spatiotemporal)	15.126% (raw data) / 4.563% (spatiotemporal)
S1	31.809 (raw data) / 2.306 (spatiotemporal)	25.221% (raw data) / 5.286% (spatiotemporal)
O1	12.047 (raw data) / 1.523 (spatiotemporal)	13.140% (raw data) / 4.347% (spatiotemporal)
O2	11.786 (raw data) / 1.416 (spatiotemporal)	12.654% (raw data) / 4.258% (spatiotemporal)
M1	25.243 (raw data) / 2.014 (spatiotemporal)	21.776% (raw data) / 5.014% (spatiotemporal)

4.5 Reduction of missing data

As mentioned in the final section of Chapter 3. To enhance overall data quality, this study designed a roll call system using an MQTT broker, which was then visualized using AWS for efficient device management. The AWS Quicksight webpage results are shown in Fig. 4-26. Each device sends a system log to the MQTT broker every minute. Upon receipt, the data is immediately synchronized and uploaded to AWS IoT Core, which is then displayed on AWS Quicksight. A green indicator shows that a specific field device's system log was received within the last 30 minutes. If no log is received for over 30 minutes, the device is considered abnormal. This method allows for rapid management of all field devices, thereby improving the overall quality of data collection.

doi:10.6342/NTU202404155

Fig. 4-26. The AWS Quicksight webpage results of six experimental fields.

doi:10.6342/NTU202404155

CHAPTER 5

Conclusions and Recommendations

5.1 Conclusions

This work introduces a novel approach based on deep learning frameworks and spatiotemporal analysis to enhance the data quality of an AIoT insect pest monitoring system. The specific results and achieved goals are listed as follows:

1. Image curation:

This work used two image curation methods to filter poor quality images. The mAP@.5 improved from below 0.6 to over 0.95 after applying two image curation methods, demonstrating a significant enhancement.

2. Model training:

This work used advanced object detection and classification models, including YOLOv7 with an average mAP of 0.95 across several crop fields and ResNet-18 for insect pest classification with an overall F1-score of 0.988, resulted in a 10.0% improvement compared to the base model.

3. Data analysis:

This work provided a spatiotemporal analysis and voting mechanism with a MAPE error of around 5% compared to ground truth. The new method outperformed previous approaches and was evaluated with datasets collected over three months across six experimental sites.

doi:10.6342/NTU202404155

5.2 Recommendations

The optimization of the insect pest recognition model and the improvement of the overall data quality framework in this study have been successfully completed. However, there are some points for improvement and recommendations for future research, which are mentioned as follows:

1. The current object detection model relies on mAP@.5 and McNemar's statistical tests. While rejecting the null hypothesis and confirming significant changes in the model, the focus is still on enhancing mAP@.5 performance. It is recommended to investigate various measures and their relationship to the final outcomes. For example, incorporating the Missing rate or identifying the proportion of samples that must be filtered out to produce better results could yield further insights.
2. The present insect categorization methodology relies solely on the F1-score and McNemar's statistical test. While rejecting the null hypothesis and confirming significant differences in the model, the focus is still on enhancing F1-score performance. However, the "others" category only includes non-target insect pests, which could lead to mistakes in some fields or scenarios. It is advised that the "others" category be further examined to determine whether it should be divided into more specific categories such as water droplets, glare, dust, or other insect pests.
3. The present spatiotemporal analysis method analyzes data from six experimental sites over three months, with four cycles of data. It is advised that the data analysis be extended to six months or a year to cover the entire crop cycle, which would improve the reliability of this analysis method.

doi:10.6342/NTU202404155

References

Abd El-Ghany, N. M., Abd El-Aziz, S. E., & Marei, S. S. (2020). A review: application of remote sensing as a promising strategy for insect pests and diseases management. *Environmental Science and Pollution Research, 27*(27), 33503-33515. https://doi.org/10.1007/s11356-020-09517-2

Arsenovic, M., Karanovic, M., Sladojevic, S., Anderla, A., & Stefanovic, D. (2019). Solving current limitations of deep learning based approaches for plant disease detection. *Symmetry, 11*(7), 939. https://doi.org/10.3390/sym11070939

Atallah, R., & Al-Mousa, A. (2019). Heart disease detection using machine learning majority voting ensemble method. In *2019 2nd international conference on new trends in computing sciences.* https://doi.org/10.1109/ICTCS.2019.8923053

Ballou, D., Wang, R., Pazer, H., & Tayi, G. K. (1998). Modeling information manufacturing systems to determine information product quality. *Management Science*, *44*(4), 462-484. https://doi.org/10.1287/mnsc.44.4.462

Barzman, M., Bàrberi, P., Birch, A. N. E., Boonekamp, P., Dachbrodt-Saaydeh, S., Graf, B., Hommel, B., Jensen, J. E., Kiss, J., Kudsk, P., Lamichhane, J. R., Messéan, A., Moonen, A.-C., Ratnadass, A., Ricci, P., Sarah, J.-L., & 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

Batty, M. (2012). Smart cities, big data. *Environment and Planning B: Planning and Design*, *39*(2), 191-193.

Boyd, D., & Crawford, K. (2012). Critical questions for big data: Provocations for a cultural, technological, and scholarly phenomenon. *Information, Communication & Society*, *15*(5), 662-679. https://doi.org/10.1080/1369118X.2012.678878

doi:10.6342/NTU202404155

Boyd, D., & Marwick, A. E. (2011). Social privacy in networked publics: Teens' attitudes, practices, and strategies. In *A decade in internet time: Symposium on the dynamics of the internet and society.*

Cai, L., & Zhu, Y. (2015). The challenges of data quality and data quality assessment in the big data era. *Data Science Journal*, *14*, 2-2. https://doi.org/10.5334/dsj-2015-002

Chen, H., Chen, J., & Ding, J. (2021). Data evaluation and enhancement for quality improvement of machine learning. *IEEE Transactions on Reliability*, *70*(2), 831-847. https://doi.org/10.1109/TR.2021.3070863

Chodey, M. D., & Noorullah Shariff, C. (2022). Hybrid deep learning model for in-field pest detection on real-time field monitoring. *Journal of Plant Diseases and Protection*, *129*(3), 635-650. https://doi.org/10.1007/s41348-022-00584-w

Dai, Q., Cheng, X., Qiao, Y., & Zhang, Y. (2020). Agricultural pest super-resolution and identification with attention enhanced residual and dense fusion generative and adversarial network. *IEEE Access*, *8*, 81943-81959. https://doi.org/10.1109/ACCESS.2020.2991552

Dawei, W., Limiao, D., Jiangong, N., Jiyue, G., Hongfei, Z., & Zhongzhi, H. (2019). Recognition pest by image-based transfer learning. *Journal of the Science of Food and Agriculture*, *99*(10), 4524-4531. https://doi.org/10.1002/jsfa.9689

de Aquino, G. R. C., de Farias, C. M., & Pirmez, L. (2018). Data Quality Assessment and Enhancement on Social and Sensor Data. In *BiDu-Posters@ VLDB.*

Deng, L., Wang, Z., Wang, C., He, Y., Huang, T., Dong, Y., & Zhang, X. (2020). Application of agricultural insect pest detection and control map based on image processing analysis. *Journal of Intelligent & Fuzzy Systems*, *38*(1), 379-389. https://doi.org/10.3233/JIFS-179413

doi:10.6342/NTU202404155

Deng, L., & Yu, D. (2014). Deep learning: methods and applications. *Foundations and trends® in signal processing*, *7*(3–4), 197-387. https://doi.org/10.1561/2000000039

Deswal, M., & Sharma, N. (2014). A simplified review on fast HSV image color and texture detection and image conversion algorithm. *International Journal of Computer Science and Mobile Computing*, *3*(5), 1216-1222.

Dey, A., Bhoumik, D., & Dey, K. N. (2016). Automatic detection of whitefly pest using statistical feature extraction and image classification methods. *Int. Res. J. Eng. Technol*, *3*(06), 1-10.

Ebrahimi, M., Khoshtaghaza, M. H., Minaei, S., & Jamshidi, B. (2017). Vision-based pest detection based on SVM classification method. *Computers and Electronics in Agriculture*, *137*, 52-58. https://doi.org/10.1016/j.compag.2017.03.016

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*, 495-505. https://doi.org/doi:10.3390/agriculture10050161

Gao, D., Sun, Q., Hu, B., & Zhang, S. (2020). A framework for agricultural pest and disease monitoring based on internet-of-things and unmanned aerial vehicles. *Sensors*, *20*(5), 1487. https://doi.org/10.3390/s20051487

Gevrey, M., & Worner, S. (2006). Prediction of global distribution of insect pest species in relation to climate by using an ecological informatics method. *Journal of Economic Entomology*, *99*(3), 979-986. https://doi.org/10.1603/0022-0493-99.3.979

doi:10.6342/NTU202404155

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. https://doi.org/10.1109/CVPR.2016.90

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*. https://doi.org/10.48550/arXiv.1704.04861

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. https://doi.org/10.1109/CVPR.2017.243

Huddar, S. R., Gowri, S., Keerthana, K., Vasanthi, S., & Rupanagudi, S. R. (2012). Novel algorithm for segmentation and automatic identification of pests on plants using image processing. In *2012 Third International Conference on Computing, Communication and Networking Technologies (ICCCNT'12)*. https://doi.org/10.1109/ICCCNT.2012.6396012

Jiang, C. F., Peng, H. Y., & Yang, Y. K. (2016). Tail variance of portfolio under generalized Laplace distribution. *Applied Mathematics and Computation, 282*, 187-203. https://doi.org/10.1016/j.amc.2016.02.005

Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture, 147*, 70-90. https://doi.org/10.1016/j.compag.2018.02.016

Karkouch, A., Mousannif, H., Al Moatassime, H., & Noel, T. (2016). Data quality in internet of things: A state-of-the-art survey. *Journal of Network and Computer Applications, 73*, 57-81. https://doi.org/10.1016/j.jnca.2016.08.002

doi:10.6342/NTU202404155

Kasinathan, T., & Uyyala, S. R. (2021). Machine learning ensemble with image processing for pest identification and classification in field crops. *Neural Computing and Applications*, *33*, 7491-7504. https://doi.org/10.1007/s00521-020-05497-z

Kaya, Y., & Kayci, L. (2014). Application of artificial neural network for automatic detection of butterfly species using color and texture features. *The visual computer*, *30*, 71-79. https://doi.org/doi.org/10.1007/s00371-013-0782-8

Khan, S. M., Ali, S., Nawaz, A., Bukhari, S. A. H., Ejaz, S., & Ahmad, S. (2019). Integrated pest and disease management for better agronomic crop production. *Agronomic Crops: Volume 2: Management Practices*, 385-428. https://doi.org/10.1007/978-981-32-9783-8_19

Kogan, M. (1998). Integrated pest management: historical perspectives and contemporary developments. *Annual Review of Entomology*, *43*(1), 243-270. https://doi.org/10.1146/annurev.ento.43.1.243

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, *25*. https://doi.org/10.1145/3065386

Kumari, S., Kumar, D., & Mittal, M. (2021). An ensemble approach for classification and prediction of diabetes mellitus using soft voting classifier. *International Journal of Cognitive Computing in Engineering*, *2*, 40-46. https://doi.org/10.1016/j.ijcce.2021.01.001

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

doi:10.6342/NTU202404155

Lathey, A., & Atrey, P. K. (2015). Image enhancement in encrypted domain over cloud. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, *11*(3), 1-24. https://doi.org/10.1145/2656205

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436-444. https://doi.org/10.1038/nature14539

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324. https://doi.org/10.1109/5.726791

Lewis, W. J., Van Lenteren, J., Phatak, S. C., & Tumlinson Iii, J. (1997). A total system approach to sustainable pest management. *Proceedings of the National Academy of Sciences*, *94*(23), 12243-12248. https://doi.org/10.1073/pnas.94.23.12243

Li, F., & Xiong, Y. (2018). Automatic identification of butterfly species based on HoMSC and GLCMoIB. *The visual computer*, *34*(11), 1525-1533. https://doi.org/10.1007/s00371-017-1426-1

Li, W., Huang, R., Li, J., Liao, Y., Chen, Z., He, G., Yan, R., & Gryllias, K. (2022). A perspective survey on deep transfer learning for fault diagnosis in industrial scenarios: Theories, applications and challenges. *Mechanical Systems and Signal Processing*, *167*, 108487. https://doi.org/10.1016/j.ymssp.2021.108487

Lima, M. C. F., de Almeida Leandro, M. E. D., Valero, C., Coronel, L. C. P., & Bazzo, C. O. G. (2020). Automatic detection and monitoring of insect pests—A review. *Agriculture*, *10*(5), 161. https://doi.org/10.3390/agriculture10050161

Lin, M., Chen, Q., & Yan, S. (2013). Network in network. *arXiv preprint arXiv:1312.4400*. https://doi.org/10.48550/arXiv.1312.4400

doi:10.6342/NTU202404155

Liu, B., Hu, Z., Zhao, Y., Bai, Y., & Wang, Y. (2019). Recognition of pyralidae insects using intelligent monitoring autonomous robot vehicle in natural farm scene. *arXiv preprint arXiv:1903.10827*. https://doi.org/10.48550/arXiv.1903.10827

Liu, B., Zhang, Y., He, D., & Li, Y. (2017). Identification of apple leaf diseases based on deep convolutional neural networks. *Symmetry*, *10*(1), 11. https://doi.org/10.3390/sym10010011

Liu, J., Li, J., Li, W., & Wu, J. (2016). Rethinking big data: A review on the data quality and usage issues. *ISPRS journal of photogrammetry and remote sensing*, *115*, 134-142. https://doi.org/10.1016/j.isprsjprs.2015.11.006

Liu, J., & Wang, X. (2021). Plant diseases and pests detection based on deep learning: a review. *Plant Methods*, *17*(1), 22. https://doi.org/10.1186/s13007-021-00722-9

Miranda, J. L., Gerardo, B. D., & Tanguilig III, B. T. (2014). Pest detection and extraction using image processing techniques. *International Journal of Computer and Communication Engineering*, *3*(3), 189. https://doi.org/10.3390/agriculture10050161

Mohapatra, S., Patra, D., & Satpathy, S. (2014). An ensemble classifier system for early diagnosis of acute lymphoblastic leukemia in blood microscopic images. *Neural Computing and Applications*, *24*, 1887-1904. https://doi.org/10.1007/s00521-013-1438-3

Nasri, M., Baratchi, M., Tsou, Y. T., Giest, S., Koutamanis, A., & Rieffe, C. (2023). A novel metric to measure spatio-temporal proximity: a case study analyzing children's social network in schoolyards. *Applied Network Science*, *8*(1), 50. https://doi.org/10.1007/s41109-023-00571-6

Oerke, E.-C. (2006). Crop losses to pests. *The Journal of Agricultural Science*, *144*(1), 31-43. https://doi.org/10.1017/s0021859605005708

doi:10.6342/NTU202404155

Otsu, N. (1979). A threshold selection method from gray-level histograms. *Automatica*, *11*(285-296), 23-27.

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

Pipino, L. L., Lee, Y. W., & Wang, R. Y. (2002). Data quality assessment. *Communications of the ACM*, *45*(4), 211-218.

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

Puljak, L. (2016). Using social media for knowledge translation, promotion of evidence-based medicine and high-quality information on health. *Journal of Evidence-Based Medicine*, *9*(1), 4-7. https://doi.org/10.1111/jebm.12175

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

Rustia, D. J. A., Chiu, L. Y., Lu, C. Y., Wu, Y. F., Chen, S. K., Chung, J. Y., ... & Lin, T. T. (2022). Towards intelligent and integrated pest management through an AIoT-based monitoring system. *Pest Management Science*, *78*(10), 4288-4302. https://doi.org/10.1002/ps.7048

Sara, U., Akter, M., & Uddin, M. S. (2019). Image quality assessment through FSIM, SSIM, MSE and PSNR—a comparative study. *Journal of Computer and Communications*, *7*(3), 8-18. https://doi.org/10.4236/jcc.2019.73002

doi:10.6342/NTU202404155

Silveira, M., & Monteiro, A. (2009). Automatic recognition and measurement of butterfly eyespot patterns. *Biosystems*, *95*(2), 130-136. https://doi.org/10.1016/j.biosystems.2008.09.004

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*. https://doi.org/10.48550/arXiv.1409.1556

Skendžić, S., Zovko, M., Živković, I. P., Lešić, V., & Lemić, D. (2021). The impact of climate change on agricultural insect pests. *Insects*, *12*(5), 440. https://doi.org/10.3390/insects12050440

Susanto, A., Dewantoro, Z. H., Sari, C. A., Setiadi, D. R. I. M., Rachmawanto, E. H., & Mulyono, I. U. W. (2020, July). Shallot quality classification using HSV color models and size identification based on Naive Bayes classifier. In *Journal of Physics: Conference Series (Vol. 1577, No. 1, p. 012020). IOP Publishing.* https://doi.org/10.1088/1742-6596/1577/1/012020

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* https://doi.org/10.1109/CVPR.2015.7298594

Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., & Liu, C. (2018). A survey on deep transfer learning. Artificial Neural Networks and Machine Learning–*ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part III 27.*

Tan, M., & Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning.*

doi:10.6342/NTU202404155

Thenmozhi, K., & Reddy, U. S. (2019). Crop pest classification based on deep convolutional neural network and transfer learning. *Computers and Electronics in Agriculture, 164*, 104906. https://doi.org/10.1016/j.compag.2019.104906

Torrey, L., & Shavlik, J. (2010). Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques* (pp. 242-264). IGI global. https://doi.org/10.4018/978-1-60566-766-9.ch011

Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience, 2018*. https://doi.org/10.1155/2018/7068349

Wang, J., Lin, C., Ji, L., & Liang, A. (2012). A new automatic identification system of insect images at the order level. *Knowledge-Based Systems*, *33*, 102-110. https://doi.org/10.1016/j.knosys.2012.03.014

Wang, X., Wang, C., Yao, J., Fan, H., Wang, Q., Ren, Y., & Gao, Q. (2022). Comparisons of deep learning and machine learning while using text mining methods to identify suicide attempts of patients with mood disorders. *Journal of affective disorders, 317*, 107-113. https://doi.org/10.1016/j.jad.2022.08.054

Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M. (2023). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 7464-7475). https://doi.org/10.48550/arXiv.2207.02696

Wen, C., Wu, D., Hu, H., & Pan, W. (2015). Pose estimation-dependent identification method for field moth images using deep learning architecture. *biosystems engineering*, *136*, 117-128. https://doi.org/10.1016/j.biosystemseng.2015.06.002

doi:10.6342/NTU202404155

Xin, M., & Wang, Y. (2021). Image recognition of crop diseases and insect pests based on deep learning. *Wireless Communications and Mobile Computing*, *2021*, 1-15. https://doi.org/10.1155/2021/5511676

Zamir, S. W., Arora, A., Khan, S., Hayat, M., Khan, F. S., & Yang, M.-H. (2022). Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. https://doi.org/10.1109/CVPR52688.2022.00564

doi:10.6342/NTU202404155