國立臺灣大學共同教育中心國際學院
全球農業科技與基因體科學碩士學位學程

碩士論文

Master Program in Global Agriculture Technology and Genomic Science
Center of General Education, International College
National Taiwan University

Master's Thesis

應用深度學習和無人載具整合的多模態方法對櫻桃番茄成熟度進行評估

Multi-modal Approach for Cherry Tomato (*Solanum lycopersicum*) Maturity Assessment through Deep Learning and Unmanned Vehicles (UV) Integration

墨艾誠
Aeron Rollon Mojica

指導教授：顏炳郎 博士
Advisor: Ping-Lang Yen, Ph.D.

中華民國 113 年 7 月
July 2024

doi:10.6342/NTU202401594

國立臺灣大學碩士學位論文
口試委員會審定書
MASTER'S THESIS ACCEPTANCE CERTIFICATE
NATIONAL TAIWAN UNIVERSITY

應用深度學習和無人載具整合的多模態方法對櫻桃番茄成熟度進行評估

Multi-modal Approach for Cherry Tomato (Solanum lycopersicum) Maturity Assessment through Deep Learning and Unmanned Vehicles (UV) Integration

The undersigned, appointed by the Department / Institute of The Master Program in Global Agriculture Technology and Genomic Science on 3 (date) July (month) 2024 (year) have examined a Master's thesis entitled above presented by Aeron Rollon Mojica (name) R11H43009 (student ID) candidate and hereby certify that it is worthy of acceptance.

口試委員 Oral examination committee:

(指導教授 Advisor)		

doi:10.6342/NTU202401594

Acknowledgement

"I can do all things through Christ who strengthens me (Philippians 4:13)."

I wish to express my utmost gratitude, deepest respect and appreciation to the following who have selflessly given their encouragement, immense hard work, and support for the accomplishment of this master's thesis.

Dr. Ping-Lang Yen, my adviser, for his unwavering support, remarkable guidance, for sharing his knowledge, and for always believing that I can do things beyond my initial capabilities.

Dr. Shih-Fang Chen, Dr. Hsiao-Mei Wu, and Dr. Gella Patria Abella, for their valuable insights, remarks, suggestion and recommendations, that greatly helped in improving the overall quality of the study.

To the Southeast Asian Regional Center for Graduate Study and Research in Agriculture (SEARCA) and Master Program in Global Agriculture Technology and Genomic Science (Global ATGS), I would like to express my heartfelt appreciation for providing the financial support. Without your help and generosity, studying at NTU would not be possible.

To all NTU International College Professors, classmates and juniors especially Hai Trieu, Nan, and Joyce for being my food buddies, and to my dear friends, especially Lester, Sheena, Cristine, and Mabelle, thank you all for your kindness, for being supportive, and for helping me in every point of this academic journey.

doi:10.6342/NTU202401594

To my colleagues and Lab mates, especially Audric and Felix, thank you for the friendship that we have forged, for all your support, as well as the personally and academically-related advices, that greatly assisted me throughout my academic journey.

To my mentors and colleague-turned family, Dr. Ruel, Ma'am Sally, Sir Andrew, Ma'am Jing, and Kuya Wilson, thank you for all the encouragement, emotional support, and advices which greatly helped me reach where I am now.

To Carla May, my other half, for her support, love, understanding, trust, care, and for serving as an inspiration and motivation especially in times of downcast.

To my dear parents, Danilo and Merlina Mojica, my sister, Abigael, and to all my relatives and loved ones, thank you very much for the care, moral and financial support, for the guidance and wisdom, and for all encouragements and sacrifices that helped me to go through this journey with relief.

Above all, to the **ALMIGHTY GOD**, who made all these things possible, for the continuous channel of blessings, for giving me the strength, courage, knowledge and wisdom, and peace of mind, throughout this journey. Truly, all honor and glory are Yours now and forever.

doi:10.6342/NTU202401594

中文摘要

近代農業中，櫻桃番茄（*Solanum lycopersicum*）是一種經濟重要的農產品。然而，它們的小尺寸和獨特特性，包括短暫的壽命和易受損的脆弱性，給作物監測和收穫帶來了挑戰。迄今為止，大多數先前的研究都集中在開發和優化深度學習模型，用於檢測不同成熟水平的番茄。然而，距離和信心閾值等因素對物體檢測器的檢測性能的影響往往被忽略和未加調查。因此，本研究提出了一種使用無人機（UAV）和YOLOv8自動檢測溫室櫻桃番茄成熟水平的方法。隨後，對上述因素的影響進行了調查。通過使用UDP協議建立了無人機與計算機的通信系統，利用DJI Tello無人機進行了測試，實現了高效的數據傳輸和UAV控制。利用使用DJI Tello、手機和Intel RealSense D435深度相機獲取的櫻桃番茄數據集，對基於YOLOv8n的深度學習模型進行了訓練。微調和消融研究的結果表明，將坐標注意塊和具有動態聚焦機制的邊界框回歸損失（WIoU）損失函數結合起來，可以實現高精度（90.2%）、召回率（88.5%）、F1分數（89.34%）和mAP（93.7%）的成熟檢測模型。開發的模型YOLOv8n + CA + WIoU用於檢測和跟踪，與微調的BoT-SORT跟踪算法一起。結果顯示，BoT-SORT對櫻桃番茄的跟踪效果良好，多目標跟踪精度（MOTA）在74%至87%之間。此外，這些結果進一步證實了低閾值導致更高的敏感性，而高閾值導致更高的特異性和增加的跟踪性能。然而，觀察到YOLOv8n + CA + WIoU算法在40厘米至100厘米的物體至相機距離範圍內開始顯示出檢測確定性下降，特別是在同類番茄之間存在遮擋和接近的情況下。總的來說，這些研究結果突顯了利用UAV系統和先進的深度學習模型在溫室環境中高效準確地監測櫻桃番茄成熟水平的潛力。此外，後續的研究結果還表明，物體至相機距離、信心閾值和遮擋對檢測性能的影響至關重要。解決這些因素對於最大程度地提高UAV

doi:10.6342/NTU202401594

基礎的農業監測系統的準確性和可靠性至關重要，進一步加強了這些技術在實際應用和工業應用中的可行性和有效性。

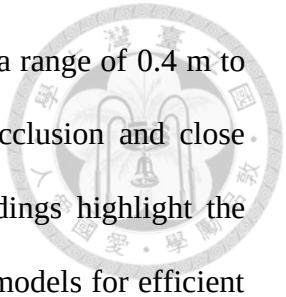
關鍵詞：櫻桃番茄成熟度檢測、UAV、YOLOv8、坐標注意、WIoU、BoT-SORT

doi:10.6342/NTU202401594

ABSTRACT

In modern agriculture, cherry tomato (*Solanum lycopersicum*) is an economically significant commodity. However, their small size and unique characteristics, including their brief lifespan and vulnerability to damage, pose challenges in crop monitoring and harvesting. Up to date, most previous studies focused on developing and optimizing deep learning models for detecting tomatoes at different maturity levels. However, the impact of factors like distance and confidence threshold on the detection performance of object detectors is often ignored and left uninvestigated. Therefore, this study proposed an autonomous method of detecting maturity levels of greenhouse-grown cherry tomatoes using UAV and YOLOv8. Subsequently, the impact of the aforementioned factors was investigated. DJI Tello drone was utilized to setup a UDP-based communication, enabling efficient data transmission and UAV control. For the model training, a cherry tomato dataset comprising images from different modalities were used. The results of fine-tuning and ablation studies demonstrated the effectiveness of incorporating coordinate attention blocks and a bounding box regression loss with dynamic focusing mechanism (WIoU) loss function, achieving high precision (90.2%), recall (88.5%), F1-score (89.34%), and mAP (93.7%) for the ripe detection model. The developed model, YOLOv8n + CA + WIoU, was used for detection and tracking together with a fine-tuned BoT-SORT tracking algorithm. The results revealed the efficacy of BoT-SORT for tracking the cherry tomatoes by achieving Multi-Object Tracking Accuracy (MOTA) ranging from 74 ~ 87% and Counting Accuracy ranging from 60% ~ 85%. Moreover, the results of the study determined the implications that for cherry tomatoes, low threshold (45% ~ 50%) leads to higher sensitivity, while higher threshold (75% ~ 80%) leads to higher specificity and increased tracking performance. However, it was observed that the YOLOv8n + CA +

doi:10.6342/NTU202401594



WIoU algorithm started to display decrease in detection certainty at a range of 0.4 m to 1.0 m object-to-camera distance, particularly in the presence of occlusion and close proximity between tomatoes with the same class. The overall findings highlight the potential of using UAV-based systems and advanced deep learning models for efficient and accurate monitoring of the maturity level of cherry tomatoes in greenhouse settings. Furthermore, subsequent findings also demonstrate the critical impact of object-to-camera distance, confidence threshold, and occlusion on detection performance. Addressing these factors are essential for maximizing the accuracy and reliability of UAV-based agricultural monitoring systems, reinforcing the feasibility and effectiveness of these technologies in real-world and industrial applications.

Keywords: Cherry tomato ripeness detection, UAV, YOLOv8, Coordinate Attention, WIoU, BoT-SORT

doi:10.6342/NTU202401594

Table of Contents

doi:10.6342/NTU202401594

doi:10.6342/NTU202401594

doi:10.6342/NTU202401594

List of Figures

doi:10.6342/NTU202401594

doi:10.6342/NTU202401594

List of Tables

doi:10.6342/NTU202401594

List of Appendices

doi:10.6342/NTU202401594

Abbreviations

AI	Artificial Intelligence
BBR	Bounding Box Regression
CA	Coordinate Attention
CNN	Convolutional Neural Network
FMR	Farm-to-Market Road
FPS	Frames Per Second
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HSV	Hue, Saturation, and Value
IoU	Intersection-over-Union
LWYS	Label What You See
mAP	Mean Average Precision
MOT	Multi-Object Tracking
MOTA	Multi-Object Tracking Accuracy
SDK	Software Development Kit
SGD	Stochastic Gradient Descent
SORT	Simple Online Realtime Tracking
UAV	Unmanned Aerial Vehicle
UDP	User Datagram Protocol
UV	Unmanned Vehicle
YOLO	You Only Look Once

doi:10.6342/NTU202401594

Chapter 1. Introduction

In the domain of precision agriculture, a prominent development involves the replacement of manual labor with robotic counterparts, marking a paradigm shift towards increased accuracy and efficiency. These technological advancements and transitions are particularly crucial in addressing the precise detection of fruit ripeness, as they not only enable timely intervention but also ensure optimal yield quality and resource utilization in modern farming practices. Manual inspection of tomato's maturity is considered to be labor-intensive and susceptible to human errors; hence, automating the tomato maturity detection process using imaging, computer vision, and deep learning techniques may be necessary to provide real-time data for informed decision-making.

As Unmanned Aerial Vehicle (UAV) can easily obtain real-time data, it has great potential for initial ripeness monitoring and surveillance. UAV possesses the ability to quickly cover expansive greenhouse structures from an aerial perspective, providing a comprehensive view of the tomatoes' conditions, especially their maturity. This capability proves beneficial during the critical stages of ripeness assessment, allowing farmers to quickly identify areas requiring immediate harvesting attention. Through the introduction of computer vision and deep learning in the UAV system, automated detection may help to improve operational efficiency.

Prior research has mostly concentrated on developing and optimizing deep learning object detectors to identify tomatoes at varying stages of maturity. However, the determination of factors affecting the overall detection performance is frequently overlooked and left uninvestigated. Therefore, the primary aim of this study is to develop a detection model using UAV and the YOLOv8 algorithm to accurately classify the maturity levels of cherry tomatoes cultivated in greenhouse environments. This model not only focuses on the precise identification and classification of the maturity stages of

doi:10.6342/NTU202401594

the cherry tomatoes but also integrates advanced object tracking mechanisms. By employing the principles of Multi-Object Tracking (MOT), the study seeks to continuously monitor and track individual tomatoes over time, providing dynamic insights into their growth and development. Additionally, the key factors that influence the overall detection performance of the YOLOv8-based object detector were observed and identified, with the overall goal of enhancing the precision and reliability of automated agricultural monitoring systems.

doi:10.6342/NTU202401594

Chapter 2. Literature Review

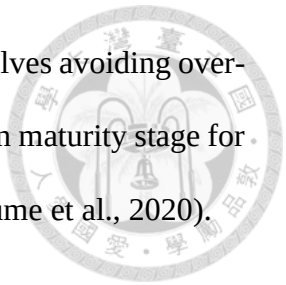
2.1 Cherry Tomato

Tomatoes (*Solanum lycopersicum*) are unequivocally vital in agriculture, serving as an essential cornerstone in farming practices (Tsironis et al., 2020) and considered as an economically significant horticultural commodity (Afonso et al., 2020; de Luna et al., 2019; Kimura & Sinha, 2008; Rodríguez-Ortega et al., 2019). However, harvesting this crop proves to be challenging and time consuming, mainly due to the distinctive planting features of tomatoes, such as their brief lifespan and susceptibility to damage. Achieving accuracy in timing is imperative during the harvesting process, further emphasizing the intricate nature of this agricultural activity (Tsironis et al., 2020).

In general, tomatoes can be classified into six categories based on their maturity stage: (1) green; (2) breaker; (3) turning; (4) pink; (5) light red; and (6) red (El-Bendary et al., 2015; Baek et al., 2020). This commodity possesses a climacteric nature which allows them to be harvested at an earlier maturity stage, because they have the capability to subsequently ripen even after their detachment from the host plant (El-Bendary et al., 2015; Tsouvaltzis et al., 2023). Hence, tomatoes can still reach a full red color even when harvested at the green maturity phase (El-Bendary et al., 2015). Nevertheless, the post-harvest physiological process, performance, and sensory attributes are significantly influenced by the fruit's maturity stages (Tolasa et al., 2021). During the storage period, physiological damage, shrinkage, and degraded quality are common problems for immature fruits, whereas overripe fruits are more vulnerable to physical harm, increased biochemical activity that causes senescence, disease, and insect infestation (Arah et al., 2016; Hamdu et al., 2016). Therefore, harvesting the fruits at proper maturity stage can reduce post-harvest losses (Moneruzzaman et al., 2009). Optimal harvest timing, contingent upon transport logistics, necessitates a particular approach to preserve the

doi:10.6342/NTU202401594

quality of the tomatoes by minimizing mechanical damages. This involves avoiding over-ripeness for long farm-to-market routes and choosing the mature green maturity stage for longer farm-to-market roads (FMR) (Moneruzzaman et al., 2009; Njume et al., 2020).



2.2 Convolutional Neural Network (CNN) for Image-based Object Detection

In the field of computer vision, Convolutional Neural Networks (CNNs) have asserted their dominance among diverse artificial neural networks (Yamashita et al., 2018; Alzubaidi et al., 2021). This neural network strategically employs grid patterns inspired by the structural organization observed in the animal visual cortex, a part of the brain responsible for processing visual information (Hubel & Wiesel, 1968; Yamashita et al., 2018). This inspiration is further implemented through the adaptive learning process of CNNs, which is facilitated by the integration of convolution, pooling, and fully connected layers. Convolution and pooling layers specialize in feature extraction, while the fully connected layer maps these features for the final output, often used in classification tasks. This process begins with the representation of digital images, which involves arranging pixel values in a two-dimensional (2D) grid. Moreover, CNN employs a compact grid of parameters called kernel which serves as a customizable feature extractor utilized at each position across the image (Yamashita et al., 2018).

CNN offers significant advantages in computer vision applications compared to traditional neural networks. One key benefit is the incorporation of weight sharing, which effectively reduces the number of trainable parameters. This not only mitigates the risk of overfitting but also enhances the model's ability to generalize well to new data. By sharing weights across different parts of the input, CNN efficiently captures and recognizes diverse patterns, contributing to the model's overall understanding of the data (Alzubaidi et al., 2021).

doi:10.6342/NTU202401594

2.3 Object Detection Via Deep Learning

Object detection's primary and major goal is to detect objects, perform object localization using rectangular bounding boxes, and conduct sorting and classification of the objects according to various predefined groups or categories (Zhang, X. et al., 2013; Liu et al., 2019; Xioa et al., 2020; Zou et al., 2023; Kaur & Singh, 2023). Prior to deep learning's popularity, a conventional method that involves the utilization of handcrafted features and a sliding window approach for generating bounding boxes was being used for object detection tasks. However, traditional object detection methods were considered computationally inefficient, leading to a shift in focus towards deep learning (Kaur & Singh, 2023). Object detection models can be classified into two different distinctions based on the type of their detectors: (a) two-stage; and (b) one-stage detectors.

Two-stage detectors (see Figure 1), also known as region-based frameworks (Liu, L. et al., 2019), use a sequential two-step methodology. The procedure begins with object localization through the generation of region proposals, followed by the classification process that assigns the object to specific categories. However, despite their high detection accuracy, these detectors are commonly associated with slow detection speeds (Kaur & Singh, 2023). The field of object detection using a deep learning approach has seen remarkable advancements since 2014 when Girshick et al. (2014) made a huge contribution to the advancement of Artificial Intelligence by developing the Region-based Convolutional Neural Network (RCNN) architecture.

RCNN has made significant improvements in object detection tasks; nonetheless, it contends with issues like slow detection speed, the multi-stage training pipeline, and the rigidity associated with the selective search method (Kaur & Singh, 2023). To address some of those drawbacks, the Spatial Pyramid Pooling Network (SPP-Net) layer was introduced and added on top of the final convolutional layer. The addition of the network

doi:10.6342/NTU202401594

layer resulted in a significant improvement in detection speed without compromising the accuracy. Despite achieving higher efficiency, SPP-Net still adapts similar procedure to RCNN such as the process of neural network fine-tuning, feature extraction, and bounding box regression which thereby contribute to time and processing complexities (Zhao et al., 2019; Kaur & Singh, 2023).

Further addressing the drawbacks of RCNN and SPP-Net, Fast RCNN was developed by Girshick (2015). The architecture adopts a unified training approach, simultaneously instructing the detector on a softmax classifier and class-specific bounding box regression. Unlike RCNN and SPPNet, which train these aspects separately, Fast RCNN employs a multitask loss for integrated training. The improved architecture may have shown significant developments in terms of detection speed and accuracy, but the use of a selective search approach for region proposals makes the architecture computationally expensive and time-consuming. Ren et al. (2015) solved these problems by developing the very first end-to-end detector in deep learning called Faster RCNN. Instead of using traditional algorithms for region proposals, this architecture utilizes the Region Proposal Network (RPN) for feature map generation.

Despite that noteworthy advancement, Faster RCNN experiences complexities in training and has trouble identifying small and multiscale objects (Kaur & Singh, 2023). In this regard, the Feature Pyramid Network (FPN) was developed to address the challenge of detecting objects at varying scales in object detection. This feature extractor solves multi-scale problems through pyramid representations (Lin et al., 2017). Other problems which arose in the field of object detection involve issues of instance segmentation. To address the existing problem, one of the latest developments in two-stage object detectors is the Mask RCNN developed by He et al. (2017). Its approach involves a thorough pixel-level inspection to estimate the presence of objects within

doi:10.6342/NTU202401594

selected regions. While it follows the Faster R-CNN architecture, Mask RCNN introduces a notable enhancement by generating three outputs for each proposed object. Despite that notable development, Mask RCNN still faces low detection speed for real-time applications.

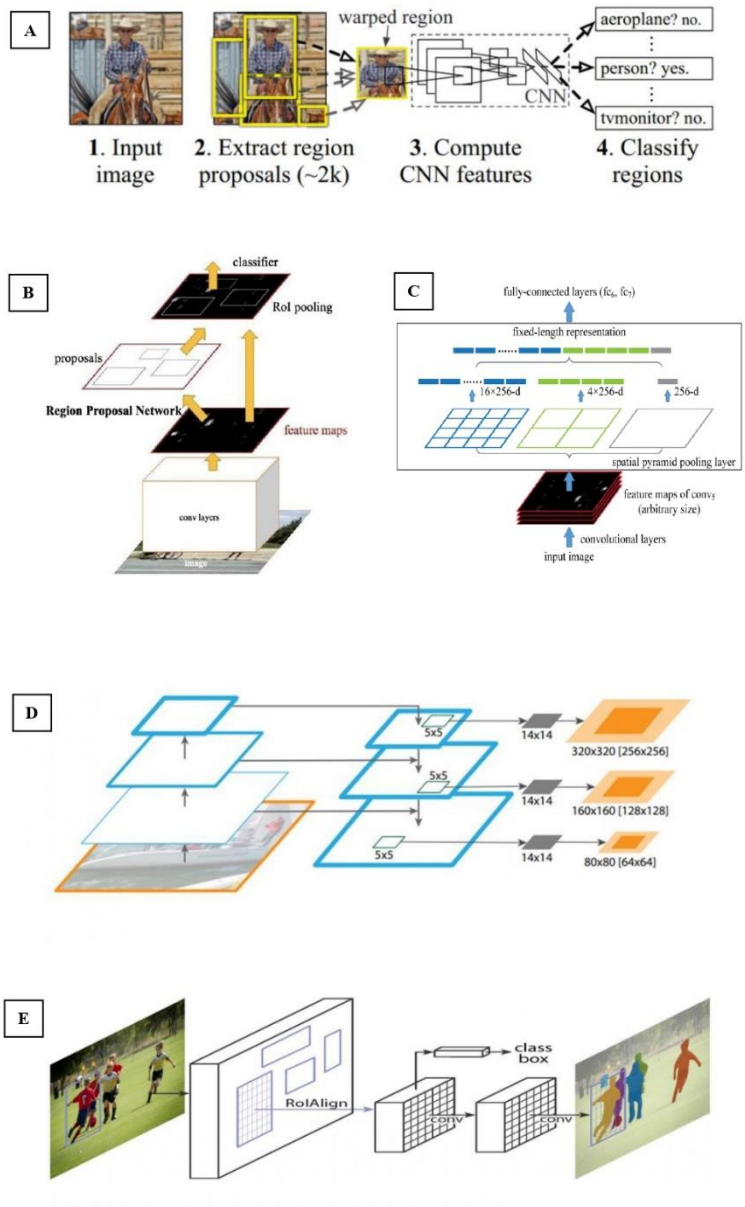


Figure 1. Two-stage Object Detectors: (A) Faster R-CNN (Girshick et al., 2015); (B) Region Proposal Network (Ren et al., 2015); (C) SPP Network Structure (He et al., 2015); (D) FPN for Object Segment Proposals (Lin et al., 2017); (E) Mask R-CNN (He et al., 2017)

doi:10.6342/NTU202401594

Compared to two-stage object detectors, the one-stage object detection model only requires a single pass through a neural network. DetectorNet, which was built upon the AlexNet backbone, treats the problem as a regression task. In this model, the softmax layer was replaced with a regression layer for predicting foreground pixels (Szegedy et al., 2013). Despite its capability to learn features for classification and acquire geometric information, DetectorNet's slow training process, necessitating training for each object and mask type, and its limitation in handling multiple objects of similar classes are considered as the architecture's notable drawbacks. On the other hand, a fully convolutional deep network called OverFeat (Sermanet et al., 2013), takes a unified structure for localization, classification, and detection using a multi-scale sliding window approach. However, the training strategy in OverFeat involves sequential training of classifiers and regressors, setting it apart from DetectorNet. Moreover, OverFeat struggles in dealing with multiple instances of the same class (Kaur & Singh, 2023). Before the significant development and improvement of the YOLO algorithm, SSD was first implemented by Liu, W. et al. (2016). It is a fast single-shot multi-box detector that integrates the concept of YOLO's regression and Faster RCNN's anchor mechanism (Tang et al., 2017; Kaur & Singh, 2023). Even though SSD was observed to be faster than YOLO and on par with Faster RCNN, the detection of small objects with this architecture was still an issue (Kaur & Singh, 2023). Similar to DetectorNet, the YOLO architecture which was developed by Redmon et al. (2016) also carries out object detection as a regression problem. Predicting the coordinates of bounding boxes and determining the likelihood of predefined categories, the model attains end-to-end optimization by only using a single network. It directly predicts detections with a limited set of candidate regions, setting it apart from region-based strategies which depend on features from specific regions. YOLO's distinct approach involves utilizing features comprehensively

doi:10.6342/NTU202401594

across the entire image. Throughout the years, YOLO algorithm was enhanced over and over to address the underlying problem concerning the difficulty with the detection of small objects and real-time applications (Redmon & Farhadi, 2017; Redmon & Farhadi, 2018; Thuan, 2021; Bochkovskiy et al., 2020; Li, C et al., 2022; Wang, C. et al., 2022). Specifically, Kaur & Singh (2023) concluded that the YOLOv5 model effectively addressed the challenge of detecting small objects. Moreover, it emerged as the fastest model compared to other object detectors given that YOLO's 8th version was developed just recently during that period and needs further insights when it comes to their architectural decisions compared to earlier versions of YOLO (Terven et al., 2023a). Further substantiating these findings, a recent study conducted by Camacho & Morocho-Cayamcela (2023) highlighted that YOLOv8 outperformed Mask RCNN in instance segmentation and classification tasks, emphasizing the advantages of one-stage object detection models over their two-stage counterparts.

YOLOv8, along with other advancements in the YOLO framework, became a sophisticated and efficient framework for real-time applications. Through the application of architectural improvements, innovative training protocols, and effective dataset augmentation on the architecture, the performance across the YOLO family was remarkably enhanced. Additionally, the adoption of transfer learning has played an important role in YOLO's adaptability, allowing the framework to excel in diverse object detection tasks (Terven et al., 2023a).

2.4 Phenotyping Using Deep Learning Methods

In plant phenotyping and agriculture, the use of deep learning is becoming increasingly popular (Lawal, 2021; Tsironis et al., 2020), particularly when dealing with tomatoes. Because of their short lifespan and sensitivity, tomatoes' harvesting process is

doi:10.6342/NTU202401594

considered a challenging and delicate procedure that requires precise timing especially when dealing with the identification of their ripening stages (Tsironis et al., 2020). Thus, the development of a system that can help in improving the efficiency of tomato harvesting process is necessary. Maturity detection commonly relies on object detection, a particularly informative aspect of deep learning in the context of fruit detection (Afonso et al., 2020) which demands more intricate training data.

Several studies which were recently published involve the utilization of object detection for tomato maturity recognition. In 2020, Afonso et al. (2020) proposed a fruit detection and counting system for tomatoes using deep learning and depth cameras. In this work, ResNet and ResNext architectures were utilized as the backbone of the MaskRCNN algorithm. They concluded that ResNext 101 architecture is consistently better than ResNet 50- and 101-layer architectures, with results ranging from 93% to 95% for precision, recall, and F1-score. Another study which involves tomato maturity recognition concluded that most state-of-the-art detectors were not yet production-ready when the dataset projects real case scenarios (Tsironis et al., 2020). The author mentioned and highlighted that one-stage detectors, which include YOLO, resulted in poor detection accuracy using their dataset with a mean average precision of 63.92% (YOLOv3). The major challenge which caused the poor detection performance of the detectors was attributed to the small size of tomato in the images. YOLOv3 was then modified by incorporating a dense architecture, spatial pyramid pooling, and the Mish function activation (Lawal, 2021). By incorporating the modifications, the models were able to achieve Average Precision (AP) values as high as 98% to 99.5%. While these techniques can improve the model's performance, they also come with a trade-off of increased computational complexity.

doi:10.6342/NTU202401594

Recently, improved YOLOv5 architecture was utilized to classify tomatoes into four different maturity stages: (a) mature green; (b) breaker; (c) pink; and (d) red (Li, R. et al., 2023). The research achieved a mean Average Precision (mAP) of 97.42%, with Precision and Recall of 95.58% and 90.07%, respectively. The model demonstrated the robustness and capability of detecting the maturity of tomatoes under occlusions with considerable accuracy. Another study also used an improved YOLOv5 architecture (Appe et al., 2023). The authors integrated a Convolutional Block Attention Module (CBAM) into YOLOv5 to improve the model's accuracy. However, its mean Average Precision (mAP) of 88.1% is relatively lower compared to the work of Li et al. (2023).

Latest studies utilized the latest version of YOLO, the YOLOv8, for tomato counting and maturity recognition. YOLOv8's backbone was improved by adopting the MSHA attention mechanism (Li, P. et al., 2023). After initially evaluating the performance of standard YOLO models, they discovered that YOLOv8 has the highest mAP of 85.9%. After improvement, the authors were able to increase the mAP up to 86.4% by incorporating the MSHA attention mechanism. Despite achieving satisfactory results, they still emphasized that occlusion, background virtualization, and light interference are still the major factors that affect a model's performance. Another study which tested YOLOv8 using the Laboro Instance Segmentation dataset (https://github.com/laboroai/LaboroTomato) achieved an overall mAP of 83.7% under an IoU of 0.50 (Camacho & Morocho-Cayamcela, 2023), approximately 2% to 3% lower compared to the mAP obtained by YOLOv8 in other studies.

The application of deep learning, particularly through advanced object detection models such as YOLOv3, YOLOv5, YOLOv8, and their respective modifications, has significantly advanced the field of tomato phenotyping and maturity recognition in agriculture. These studies showcase the continuous efforts to address challenges in tomato

doi:10.6342/NTU202401594

harvesting efficiency, specifically emphasizing the complexities associated with small-sized objects and real-world scenarios. However, it is crucial to acknowledge persistent challenges related to occlusions, background variations, and light interference, emphasizing the ongoing need for fine-tuning hyperparameters and exploring innovative solutions to enhance model robustness.

2.5 Loss Function

Deep learning, which utilizes hierarchical architectures, has been broadly applied in the field of Artificial Intelligence, most commonly in transfer learning and computer vision (Guo, Y. et al. 2016). When dealing with complex problems which involves images, this subfield of machine learning has been recognized as a significant tool (Terven et al., 2023b). In the context of object detection, localization of the objects plays a crucial role. This aspect critically relies on the evaluation and optimization of the loss function known as bounding box regression (BBR) (Zheng, Z. et al., 2020; Zheng, Z. et al., 2022; Liu, C. et al., 2024). Loss function in general, may serve as a metric to determine how well a model can estimate the desired result by calculating the difference between the predicted output and the ground truth (Terven et al., 2023b). According to Rezatofighi, H. et al. (2019), regression losses can be improved through the replacement of some of its specific components in combination with a metric loss calculated based on Intersection over Union (IoU); however, it is often set aside and ignored. These information highlights the significance of BBR in the domain of 2D and 3D computer vision as it affects object detectors' overall performance (Rezatofighi, H. et al., 2019; Liu, C. et al., 2024).

Loss functions are used during training in order to optimize the parameter settings of the deep learning models; hence, proper selection is necessary to achieve satisfactory and acceptable performance (Tian et al., 2022; Terven et al., 2023b). However, selecting the most suitable protocol is deemed to be a difficult process given the availability of

doi:10.6342/NTU202401594

wide range of choices. In the field of deep learning, Terven et al. (2023b) emphasized the importance of evaluating loss functions based on several critical properties. These properties encompass: (a) convexity; (b) differentiability; (c) robustness; (d) smoothness; (e) sparsity; (f) multi-modality; (g) monotonicity; and (h) invariance. For object detection tasks, they pointed out the prevalent use of four key loss functions: (1) Smooth L1; (2) IoU Loss; (3) Focal Loss; and (4) YOLO Loss. However, the efficiency of IoU metrics faces a principal problem involving infeasible optimization, especially when dealing with non-overlapping bounding boxes.

Nowadays, several loss functions for object detection tasks which utilize CNN are gaining popularity especially to address faster convergence and better performance (Zheng, Z., 2020). One of the commonly used loss functions is the Complete-IoU (CIoU). CIoU includes a term related to the aspect ratio consistency to provide a more comprehensive evaluation of the bounding box regression task instead of focusing solely on the distance between the centers of the boxes without taking into account the aspect ratio (Tian et al., 2022). Despite those developments throughout the years, training data populated with a portion of poor-quality samples tend to impede the generalization capability of object detection models. Majority of the previous studies focused on how to enhance the capacity of BBR loss to fit high-quality samples in the training set. However, enhancing the BBR loss resulted to a compromised localization performance. This drawback is primarily caused by the build of existing detection models, as they are limited to static focusing mechanism (FM).

2.6 The WIoU Loss Function

When the anchor box closely aligns with the target box, an effective loss function should reduce penalties for geometric discrepancies, enhancing the model's generalization ability (Tong et al., 2023). Most current approaches use a static focusing

doi:10.6342/NTU202401594

mechanism (FM) that does not fully utilize non-monotonic FM and focuses on improving the bounding box regression (BBR) loss, assuming high-quality training data. However, excessively strengthening BBR on low-quality examples can harm localization performance. In object detection, efforts have been made to improve algorithmic self-learning for better recognition of small objects (Ni et al., 2024). To address this, Tong et al. (2023) proposed a dynamic non-monotonic FM, resulting in the Wise-IoU (WIoU) loss function. WIoU, a dynamic non-monotonic focal loss function based on IoU, improves gradient gain allocation by assessing anchor box quality through outlier degree instead of IoU. This reduces the negative impact of low-quality examples and competition among high-quality anchor boxes, focusing on producing anchor boxes of average quality. This dynamic adjustment enhances the model's ability to accurately detect objects, especially for small objects (Ni et al., 2024). Implementing WIoU leads to better overall detection performance by balancing the learning of low-quality and high-quality examples.

With this particular loss function, high-quality anchor boxes become less competitive and low-quality samples bears less noticeable impact. The study demonstrated an improved overall detection performance by weighing the learning of low-quality examples and high-quality examples (Tong et al., 2023).

2.7 Attention Mechanisms

In the context of neural networks, attention mechanisms play an essential role in allowing the model to focus on pertinent information. By integrating attention mechanisms, the model gains the capability to weight the importance of different parts of the input sequence based on the current state of the decoding procedure. This further allows the model to direct its attention more towards relevant parts of the input when

doi:10.6342/NTU202401594

generating each part of the output sequence. Wang, C. et al. (2023) highlighted in their results that modifying the structure of the YOLO algorithm by incorporating the Coordinate Attention (CA) Mechanism further improved the overall performance of the model. Results of their experiments shown that incorporating CA module to YOLOv5 exhibited significant advantages by improving the detection accuracy by a range of 0.6% to 1.7%.

2.8 Object Tracking Using Deep Learning

In the field of agriculture, computer vision technologies based on Artificial Intelligence and deep learning has become a more viable option in comparison with classic image processing methods. Deep learning's capacity to extract crucial features has unlocked additional opportunities for scientific applications in agriculture, empowering researchers to tackle complex tasks with newfound precision and efficiency (Wang, Z. et al., 2023). There is a growing need for manual labor to perform specific tasks like counting and monitoring tomato yields; however, there is a continuous decline in the overall workforce available for such agricultural work. Thus, automating operations such as counting and harvesting using robots bears extreme significance and urgency (Saleem et al., 2021; Rong et al., 2023). However, the only difference tomatoes have, given their similar color and morphological features, lies on their pixel area variation (Rong et al., 2023). To address this knowledge gap, various researchers utilized the deep learning-based object detection and frame-to-frame tracking. With the development of deep learning algorithms, the detection-based moving-object-tracking paradigm has drawn a significant amount of attention (Wojke et al., 2017; Yu, C. et al., 2023), particularly in the domain of multi-object tracking (MOT). MOT is dedicated to detecting and tracking objects in a given scene while preserving unique identifiers for each object, enabling the estimation of their spatial-temporal trajectories within a video stream (Aharon et al.,

doi:10.6342/NTU202401594

2022). In the context of MOT, the tracking-by-detection paradigm stands out as the most efficient method, characterized by two key phases: initial object detection followed by subsequent object tracking.

Over the years, a remarkable increase in the utilization of various MOT algorithms across diverse applications was observed, with ByteTrack, BoT-SORT, and DeepSORT emerging as some of the foremost tracking techniques renowned for their effectiveness and widespread adoption. With the introduction of the Byte-Track technology in 2022, Zhang, Y. et al. (2022) transformed the field of tracking algorithms with their novel Byte association strategy, which significantly improved tracking accuracy. Aharon et al. (2022) followed suit in the same year with the release of the BOT-SORT algorithm, which exhibited remarkable speed and accuracy by developing a basic algorithm structure which utilizes Hungarian algorithm and Kalman filter. This groundbreaking algorithm achieved top performance on the MOT-Challenge dataset, which includes the challenging MOT17 and MOT20 dense pedestrian test sets. It did so with the aid of camera motion compensation, an advanced Kalman filter state vector, and the smooth integration of motion and appearance data (Aharon et al., 2022; Host et al., 2023). Alongside BoT-SORT, DeepSORT also utilizes the Hungarian algorithm to address the problem with global assignment. These algorithms play crucial roles in efficiently linking the detected object to their respective tracker across consecutive frames. (Host et al., 2023).

Deep learning techniques were already claimed to be beneficial in video surveillance systems since a decade ago (Sacchi et al., 2013). The assertion was emphasized by Hnoohom et al. (2024) with the introduction of a video-oriented safety approach using pedestrian and vehicle monitoring techniques to determine time-to-collision (TTC) and post-encroachment time (PET), with the principal objective of enhancing safety measures in crosswalk areas. For agriculture-specific applications, Rong

doi:10.6342/NTU202401594

et al. (2023) utilized ByteTrack algorithm to track and count tomato clusters in a greenhouse environment. The results of their field experiments enabled a yield estimation approach with real-time capability and stability as it reached a 95.1% average counting accuracy. Furthermore, the algorithm was also utilized in the domain of aquaculture. Wang, Z. et al. (2023) highlighted that among object tracking solutions, Bytetrack stands out for its ability to precisely address problems related to the classification of objects with identical properties, such those found in fish tracking scenarios. By consistently delivering exceptional performance metrics such as MOTA and IDF1, surpassing the 95% threshold with only detection data, Bytetrack showcases its superior performance. Furthermore, the algorithm's ability to operate at high frame rates ensures efficient real-time monitoring.

Among the aforementioned state-of-the-art algorithms, the BoT-SORT algorithm has emerged as a versatile tool for tracking various entities, including pedestrians and vehicles (Hnoohom et al., 2024), and non-traditional objects like ducks (Duan et al., 2023). BoT-SORT operates within the tracking-by-detection (TBD) paradigm, utilizing bounding boxes derived from object detection to monitor trajectories (Duan et al., 2023). This method integrates motion and appearance information, reinforcing the robustness of detection outcomes. The study involves duck tracking and monitoring at different time intervals through the use of BoT-SORT algorithm in conjunction with YOLOv8. Based on their findings, the BoT-SORT model has demonstrated remarkable accuracy, achieving a multi-object tracking accuracy of 85.1%. Furthermore, the comparative analysis of tracking algorithms for player tracking highlights the efficacy of BoT-SORT alongside Deep SORT, both reliant on external object detectors like Mask R-CNN and YOLO (Host et al., 2023). Through empirical validation and theoretical underpinnings, the utility and effectiveness of BoT-SORT and its variants in diverse tracking scenarios

doi:10.6342/NTU202401594

are observed, offering valuable insights in the field of computer vision and object tracking.

2.9 Tello EDU Drone

Within the academic community, researchers are particularly interested in low-cost, and compact-size flying robots because of the increased safety they provide for studies carried out in indoor environments where GPS are inaccurate and unreliable (Giernacki et al., 2022). The Tello EDU drone (Ryze Technology), is a compact UAV explicitly designed for students and research enthusiasts delving into UAV technologies (Boonsongsrikul & Eamsaard, 2023). This drone is designed for educational applications (Bhujbal & Barahate, 2022), given its programmability and wide availability in the market (De Silva et al., 2022). It is the first commercially available drone designed for learning programming in Scratch, Python, and Swift languages with a great potential for high-level control layers and machine learning (Giernacki et al., 2022). Equipped with a sophisticated Python SDK, users can unlock the full potential of the drone by programming it to execute an array of functions (Bhujbal & Barahate, 2022; Boonsongsrikul & Eamsaard, 2023). Adding to its versatility, the drone is easily manageable and controlled through a smartphone or PC, through the 2.4 GHz communication channel (Boonsongsrikul & Eamsaard, 2023). This control is further enhanced by the integration of DJI's flight control command system, transforming control commands into rotor motions. To enable autonomous navigation, an autonomous system can transmit control commands to the drone's flight controller (Bhujbal & Barahate, 2022).

Tello EDU was used in several researches which involves computer vision. Boonsongsrikul & Eamsaard (2023) designed and implemented a human motion tracking method using the aforementioned UAV which displayed an accuracy rate ranging from

doi:10.6342/NTU202401594

96.67% to 100% for the detection of human movement. Another study conducted by Bhujbal & Barahate (2022) utilized the Tello drone for computer vision applications. The authors utilized YOLO v3 algorithm for the detection of objects in a real-time manner.

2.10 Drawbacks and Solutions: Prolonged UAV Operations

One of the major drawbacks when it comes to the utilization of UAV is their flight time. Battery power of most drones is insufficient in providing power throughout longer-length tasks such as surveillance and monitoring, as well as remote sensing and GIS-related applications (Radiansyah et al., 2017; De Silva et al., 2022). The overall performance and efficiency of a UAV mission can be affected directly if the drone's batteries need to be charged after a brief period of usage (depending on the drone's battery capacity). Nowadays, various researchers are exploring automated solutions to replace the manual process of changing batteries with the primary goal of eliminating the need for human intervention in the battery exchange process. Although the method involving switching out batteries can relatively save time, larger quantity of fully charged batteries are necessary apart from the need for manual operation. However, depending on the charging speed, conventional approach of charging could be less complicated but may take longer time (De Silva et al., 2022). Several methods were proposed to address the battery power issues of UAVs. Some researchers proposed the use of wireless charging systems (Junaid et al., 2016; Mostafa et al., 2017; Junaid et al., 2017; Yang et al., 2019), while some proposed the use of wired charging docks (Kemper et al., 2011; Song et al., 2013; Cocchioni et al., 2014; Mulgaonkar, 2014; Leahy et al., 2015; Mourgelas et al., 2020). As researchers continue to venture and explore solutions for overcoming the limitations of UAV flight time, the prospect of automated, efficient, and prolonged aerial

doi:10.6342/NTU202401594

missions holds significant promise for the future of agricultural applications, particularly in the field of plant phenotyping.

2.11 Geometric Calibration

In computer vision, camera calibration is a crucial step (Ntouskos et al., 2007) that is necessary to be performed in order to retrieve metric information from two-dimensional images (Zhang, Z., 2000). For quantitative image analysis (IA), the calibration and alignment of cameras and images are operations of essential relevance. In essence, the identification of information such as image primitives, characteristics, and objects, in image space as well as its transfer and representation in object space are the main goals of quantitative information analysis. On the other hand, qualitative image analysis encompasses the understanding of what is depicted in one or more images, and the process of finding and recognizing specific attributes within those images (Gruen, 2001).

Camera calibration methods encompass a range of techniques, from utilizing different camera models to employing both linear and non-linear algorithms, and targeting specific 3D or 2D test-fields (Ntouskos et al., 2007). Moreover, Grammatikopoulos et al. (2007) investigated the use of points or lines as image features, contributing to the diversity of approaches in this field. However, camera calibration is also possible without any knowledge about the a priori locations of the objects (Ntouskos et al., 2007). In the process of UAV calibration, visual distortions can occur due to the inaccuracy of light and compact versions of digital cameras, which are frequently utilized for UAV imagery due to the payload limitations of compact UAVs. With the images captured by these cameras, the straight lines in the scene are translated into curved lines. Hence, the distortion should be eliminated if the data are to be used for mapping or photogrammetry (Douterloigne et al., 2009). The analysis of a set of characteristic points projected onto images, which are inherent characteristics of a known object, is a common

doi:10.6342/NTU202401594

step in calibration methods. Sequences of images with necessary calibration pattern needs to be acquired to identify and detect unique features. After these points and their two-dimensional coordinates have been extracted from the sequence of images, the relationship between each point in the pattern for the series in all the sequences necessarily needs to be discovered, to obtain both intrinsic and extrinsic parameters of the camera (de la Escalera & Armingol, 2010).

Many different types of calibration patterns have been used for camera calibration over the years. The very first patterns employed for this purpose were three-dimensional (3D) objects, rather than flat, two-dimensional (2D) patterns (de la Escalera & Armingol, 2010). Nevertheless, despite the fact that several algorithms have been developed to recognize one-dimensional patterns like numerous points on the same line (Ronda et al., 2008) or vanishing lines (Grammatikopoulos et al., 2007), majority of patterns are typically multi-dimensional due to the fact that they are economically effective and easy to construct through the use of precise printing tools such as laser printers (Zhang, Z., 2000). This is considered beneficial and practical for the precision requirement of vast majority of image analysis applications (de la Escalera & Armingol, 2010). Nowadays, patterns with similar appearance to a chess board are commonly used because the alternating clear and dark zones favor the detection of the corners of elements on the board (de la Escalera & Armingol, 2010).

2.12 Factors Affecting the Detection Accuracy

In the field of object detection, according to literatures, the factor of distance is often ignored because the mean accuracy for all objects, regardless of the distance, is typically used to quantify object detection accuracy (Yu, H. et al., 2023). A particular problem that is commonly encountered within the object detection process involves the

doi:10.6342/NTU202401594

distance between the object of interest and the camera and the light intensity (Giap et al., 2023; Yu, H. et al., 2023). Bhusnoor et al. (2023) also stated and claimed the same idea that decreased light intensity, lower-grade camera specifications, and long detection ranges may all detrimentally influence the object detection process. Giap et al. (2023) analyzed the accuracy of detection under varying distance and lighting conditions. The results of their study, through the use of methods adapted from the idea of Hendrawan & Istiono (2023), emphasized that farther distances decrease the level of accuracy; hence, distance is a crucial factor in producing good levels of accuracy. At the same year, Bhusnoor et al. (2023) and Yu, H. et al. (2023) explored the same research area. Bhusnoor et al. (2023) investigated how the accuracy of object detection, represented by the confidence level, changes as the distance between UAV and the target object increases. A person was utilized as the main subject in this research, and varying distances between the person and the UAV ranging from 5 to 25 meters were maximized to determine the effect of distance on the confidence level of the detections. The authors obtain significant findings where the increasing distance between the human subject and the UAV significantly decreases the detection accuracy. Correspondingly, Yu, H. et al. (2023) also elaborated that detecting objects at a closer distance is easier to detect than those objects seen from a relatively far distance. These studies emphasize the crucial impact of distance in the efficiency of object detection systems.

Apart from the distance between the camera and the target object, confidence threshold also affects the accuracy of object detectors. Object detection models generate detection instances or bounding boxes and utilize confidence threshold for filtering out predictions with low confidence levels. Moreover, the confidence threshold is also used for filtering false positive instances and redundant predictions (Wenkel et al., 2021; Arani et al., 2022). However, in the context of neural networks, several authors frequently

doi:10.6342/NTU202401594

choose a low classification threshold in order to achieve cutting edge performance on benchmark datasets. In some cases, this kind of approach is often preferred due to the fact that standard metrics used for evaluating the models do not come with a penalty term despite generating subsequent quantity of false positives instances (Wenkel et al., 2021). When applied in critical fields such as medicine and robotics, the implications of using object detection models with a certain level of unreliability are significantly more severe compared to their application in more common and lighter tasks. This concern has been mentioned in various studies (Amodei et al., 2016; Sünderhauf et al., 2018; Hall et al., 2020; Wenkel et al., 2021). This factor may be attributed to the confidence threshold set and selected for the specific task, since the extent of false positives or false negatives is greatly influenced by the threshold. Given the significance of accurate detections, the quantity of both false positives and false negatives should be maintained as low as possible through proper confidence threshold selection (Wenkel et al., 2021). Arani et al. (2022) who investigated the influence and effect of several deep learning factors, including confidence threshold, on the overall performance of detection models discovered that confidence threshold is relatively significant in the calculation of the accuracy of object detectors. Moreover, the authors found out that remarkable variation in accuracy is evident when tested on different threshold values. Results of their study demonstrated that higher threshold causes the accuracy to drop, while lower threshold value causes the inference time to be faster. Contrastingly, Ding et al. (2019) found out that increasing the confidence threshold offers a buffering effect against the degradation of accuracy. Setting a high threshold raises the True Positive (TP) rate, thus correctly identifying more positive instances. However, it also raises the False Negative (FN) rate, resulting in more missed positives. On the other hand, a low threshold causes the False Positive (FP) rate to rise, resulting to more incorrect positive predictions. Despite having

doi:10.6342/NTU202401594

a high rate of FP, low threshold also enhances the True Negative (TN) rate, leading to more correct negative identifications (Monaghan et al., 2021). This emphasizes the importance of carefully selecting the confidence threshold to achieve the desired balance between accuracy and efficiency in object detection models.

doi:10.6342/NTU202401594

Chapter 3. Materials and Methods

3.1 Overview of the System

The system being proposed in this study focuses on developing an autonomous system for detecting the maturity levels of cherry tomatoes grown in greenhouse environments using a DJI Tello drone and advanced deep learning models. The drone, integrated with a UDP protocol-based communication system, captures images and videos of the cherry tomatoes. These image datasets, supplemented with data from mobile phones and an Intel RealSense D435 depth camera, are used to train the YOLOv8n model. In order to achieve precise detection of the cherry tomatoes' maturity, three different detection models were developed. The first model encompasses the detection of three different stages of maturity: (a) unripe; (b) semi-ripe; and (c) fully-ripened tomatoes. Compared to the first detection model, the second model was developed for exclusively detecting cherry tomatoes at light-red maturity stage. Similar to the second model, the third model was also developed for detecting a single class which encompasses ripe tomatoes only. The model undergoes fine-tuning and ablation studies by incorporating CA blocks and a WIoU loss function to optimize performance, achieving high precision, recall, F1-score, and mAP. After successfully generating a robust and reliable model for cherry tomato maturity recognition, the developed and improved YOLOv8n models is used together with a fine-tuned BoT-SORT algorithm for the detection and tracking process. This step further aids in properly selecting the parameters needed for increasing either the sensitivity or specificity of detection. Furthermore, the system's performance is evaluated mainly based on object-to-camera distance and confidence threshold, revealing their impact on detection certainty and tracking accuracy. This integrated approach not only demonstrates the feasibility of using UAV-based systems and advanced deep

doi:10.6342/NTU202401594

learning models for agricultural monitoring but also sets a new benchmark for precision and reliability in the automation of crop management.

3.2 Area of Study

The research experiments were carried out within a cherry tomato greenhouse facility situated in Taichung City, Da'an District, Taiwan (24°20'52"N 120°36'31"E).

3.3 UAV model

The Tello EDU quadcopter emerged as the top and ideal choice to be used as case study due to its programmability and widespread availability in the commercial market. Moreover, the Tello EDU quadcopter has been extensively utilized and adopted for research purposes. The UAV model was developed through the collaborative effort between Ryze and DJI. Featuring an Intel processor and an integrated 5 MP resolution camera, it offers versatile communication capabilities via Wi-Fi and UDP connections. With a respectable flight time of approximately 13 minutes, it excels in maintaining precise positioning in windless environments and exhibits stable hovering capabilities. With an astonishingly low weight of 89 grams (including propeller guards and the battery), this device underscores the critical significance of weight management. To achieve precise control over its position and altitude, the drone relies on diverse set of sensors. These encompass a built-in barometer for altitude data, an infrared sensor, and a Time-of-Flight (ToF) camera discreetly situated on the device's underside, enhancing its in-plane positioning capabilities. Additionally, the inclusion of an IMU sensor is essential, delivering comprehensive three-way acceleration and angular position data.

A Python-based Software Development Kit (SDK) is readily accessible for the drone, encompassing comprehensive UDP communication protocols. Through connection to the drone's dedicated Wi-Fi network, users gain the capability to exert

doi:10.6342/NTU202401594

control using variety of commands and extract data. Notably, the UDP connection's maximum packet size is insufficient for transmitting camera images in real-time. As a solution, the drone ingeniously breaks down the image data into multiple packets for efficient transmission.

3.4 Communication Protocol Between Drone and Computer

For establishing the method for drone control, communication with the Tello drone is primarily facilitated through the ‘djitellopy’ library. It is a Python module built on the official SDK. This library, crafted by Damian Fuentes (https://github.com/damiafuentes/DJITelloPy), which empowers the system with access to the complete suite of functionalities, including live video streaming. Furthermore, keyboard controls and Graphic User Interface (GUI) were adapted from the research conducted by Szepessy (2019) and seamlessly integrated into the navigation system. The integration of keyboard controls serves as a critical intervention mechanism to address potential dangers or system malfunctions. Dedicated keys are assigned to initiate specific functions detailed in Appendix B. These functions are set in motion using the event class provided by the Python Programming Language, enabling seamless inter-thread communication.

Ryze Robotics (2018) provided an extensive guide detailing the Python-based control of the Tello Drone, achieved by connecting to the UAV's dedicated Wi-Fi network. Controlling the Tello Drone necessitates the initiation of a User Datagram Protocol (UDP) connection through IP address 192.168.10.1 and port 8889. A data transmission approach based on UDP involves sending a packet from the transmitter end to the receiver end at an appropriate time (Christensen et al., 2020); thus, any setup or bidirectionality is not necessary (Larmo et al., 2018). Moreover, this protocol is often

doi:10.6342/NTU202401594

used in small embedded systems (Christensen et al., 2020). Prior to executing any maneuvers, a specific command labeled as "command", must be directed at the UAV, prompting a change in its operational state, and preparing it to receive subsequent instructions. This protocol incorporates an element from the research of Szepessy (2019) for reading the status of the flight controller, where a distinct UDP socket connection is established with the server 0.0.0.0 and port 8890, operating concurrently in the background to receive drone control responses.

In this study, video frames captured by the drone's camera is also one of the important components that needs to be executed and acquired appropriately. This portion will be broadcasted through UDP by opening server 0.0.0.0 via post 11111. The encoded image data is delivered by the drone in blocks of 1460 bytes, with the last block being less than 1460 bytes, because its size exceeds the maximum payload size of a UDP transfer. After receiving the broadcast, the OpenCV package can be used to display the video (DJI, 2018) at 25 fps with a resolution of 902 x 664 pixels.

3.5 Geometric Calibration of the UAV's Camera

Through empirical geometric calibration, the camera of the Tello drone has been fine-tuned to elevate its imaging capabilities, ensuring that any distortions within the captured images are effectively eradicated, resulting in superior visual clarity. Subsequent experiments revealed that the camera matrix and distortion coefficients obtained by Szepessy (2019) who also used Tello drone for vision-based navigation system, resulted in an effective and precise pose estimation results. Hence, these set of parameters was adapted for the system developed in this study. The principal goal of the calibration was to achieve exceptionally accurate calibration results, essential for obtaining reliable and

doi:10.6342/NTU202401594

promising measurements for the computer vision tasks in this study. The camera calibration parameters are shown as follows:

$$\text{camera matrix (intrinsic parameters)} = \begin{bmatrix} f_x & 0 & c_y \\ 0 & f_y & c_y \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 941.17 & 0 & 471.52 \\ 0 & 940.42 & 363.68 \\ 0 & 0 & 1 \end{bmatrix} \quad 3.1$$

$$\text{distortion coeff.} = [k_1 \;\; k_2 \;\; p_1 \;\; p_2 \;\; k_3] = [8.67\,\text{x}10^{-2} \;\; -1.24 \;\; -3.95\,\text{x}10^{-4} \;\; -5.21\,\text{x}10^{-3} \;\; 4.39] \quad 3.2$$

Where:

f_x, f_y = focal length
c_x, c_y = optical centers
k_1, k_2, k_3 = radial distortion coefficients
$p_1, p_2,$ = tangential distortion coefficients

3.6 Tomato Maturity Recognition System

3.6.1 Dataset Collection

The dataset which is comprised of cherry tomato images were collected from a greenhouse in Taichung City, Da'an District, Taiwan (24°20'52"N 120°36'31"E) (Figure 2). The images were captured on the 2nd of March 2024, under natural greenhouse conditions. Employing mobile phone, Intel Realsense D435, and DJI Tello drone, hundreds of images were captured at different shooting distances varying from 10 cm to 100 cm. All the images were taken at different angles and varying conditions, partially similar to Wang, C. et al. (2023), by introducing elements such as (a) single cluster; (b) multiple clusters; (c) occlusion; (d) overlap; and (e ~ f) different light exposures (Figure 3). The careful consideration of these factors was performed to minimize and avoid the risk of overfitting, which could arise due to the limited diversity in the training samples.

doi:10.6342/NTU202401594

Figure 2. Cherry Tomato Greenhouse at Taichung City, Taiwan

a. Single Cluster b. Multiple Cluster c. Occlusion

d. Overlap e. Bright Light f. Dim Light

Figure 3. Different Kinds of Images in the Dataset

doi:10.6342/NTU202401594

3.6.2 Target Maturity Stage for the Detection Models

According to farmers' and consumers' perceptions, ripeness is one of the most significant indicators of the yield quality (El-Bendary et al., 2015). Despite its significance, determining the optimal time for harvest remain predominantly reliant on subjective judgment and farmers' experience (El-hariri et al., 2014). Hence, the main motivation of this research for addressing the unsolved problems comprises of the development and fine-tuning of three key deep learning models, each configured to target particular maturity stages: (a) three major maturity stages; (b) light-red stage; and (c) red stage. Throughout the development and training phases of all three models, consistent parameters and methodologies are employed to ensure uniformity and comparability of results. By adhering to standardized practices, the reliability and effectiveness of each model in accurately assessing cherry tomato maturity are optimized, contributing to improved agricultural practices and enhanced crop management strategies.

When gauging the ripeness of cherry tomatoes for harvest, the critical factor is the amount of coloration visible on the fruit's surface. Immature tomatoes have greenish color similar to branches and leaves, while mature ones exhibit a vibrant red color (Wang, C. et al., 2023). Firstly, specific criteria for the overall monitoring and surveillance model were adapted from the work of Camacho & Morocho-Cayamcela (2023) and from the owner of the farm where the dataset was collected. To further visualize the criteria for maturity classification, Figure 4 illustrates the maturity state of cherry tomatoes. Secondly, the light-red cherry tomato detection model was finely tuned to pinpoint tomatoes at the aforementioned maturity range, a critical stage identified through collaboration with farmers. At this stage of maturity, tomatoes are characterized by 60% but not more than 90% pinkish red or red surface color (El-Bendary et al., 2015). Lastly, the third model specializes in detecting cherry tomatoes at 100% maturity, characterized

doi:10.6342/NTU202401594

by their rich red coloration. A critical aspect of tomato cultivation lies in pinpointing the optimum harvesting period to prevent the fruit in overripening and spoiling while still on the plant. This moment is characterized by tomatoes reaching a rich red color while retaining a firm texture, indicating ripeness without excessive softening (Tsouvaltzis et al., 2023). Hence, a reliable detection model capable of detecting fully-ripe tomatoes is necessary for promptly identifying such fully ripe tomatoes and facilitating their immediate harvesting.

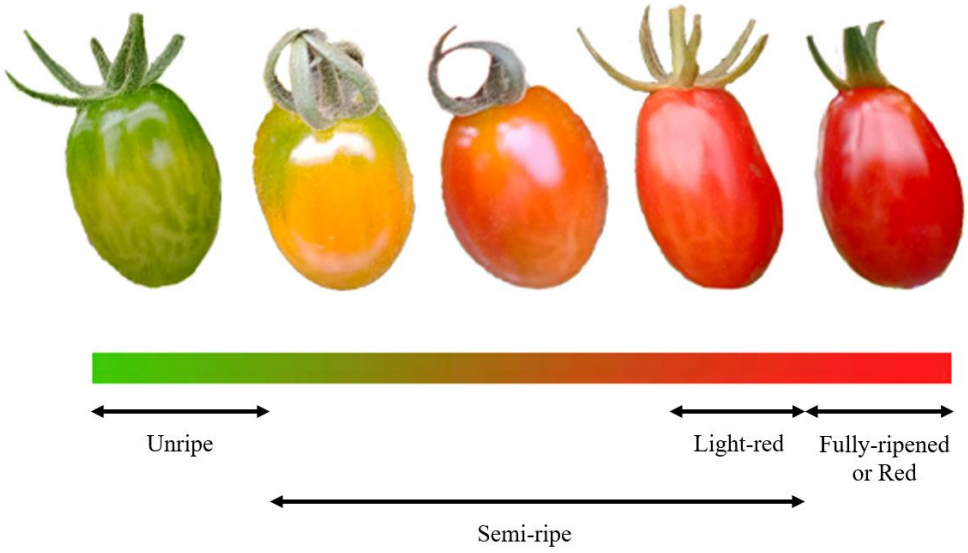


Figure 4. Developmental Stages of Cherry Tomatoes. First model encompasses unripe, semi-ripe, and red; Second model is focused on light-red; and the Third model is focused on red.

3.6.3 Experimental Setup

Model training and testing were carried out using a laptop computer with the following specifications: Intel Core i7-12650H 2.30 GHz, 16 GB running memory (RAM), 4G GDDR6 NVIDIA GeForce RTX 2050 Laptop GPU, and Windows 11 Operating System. Moreover, the configurations and versions of all the libraries used in this study are elaborated in Appendix A. Apart from those, an adaptable deep learning

doi:10.6342/NTU202401594

framework for diverse network training requirements was made possible by utilizing PyTorch. To enhance the performance and robustness of the deep learning models, the YOLO model was trained using a custom dataset collected during this study. The models were trained using a dataset consisting of images from different modalities with the following resolutions: RealSense images at 1280x720 pixels, drone images at 902x665 pixels, and phone images at 4080x3060 pixels. Additionally, a separate set of phone images with a resolution of 2000x2000 pixels was used specifically for detecting light red tomatoes. Finally, the tomato maturity classification was performed directly by the YOLOv8 architecture.

3.6.4 Experimental Workflow

Figure 5 presents the overall workflow of the object detection framework. After data cleaning process, the cherry tomato dataset has been subjected to annotation using an image labelling software. Accordingly, the dataset was divided using a 7:1:2 train-validation-test split ratio derived and adapted from Phan et al. (2023) and Tapia-Mendez et al. (2023). The three detection models in this study were trained and validated on the respective sets. As elaborated in Sections 3.6.2 and 3.6.7.1, the instances for the first detection model were categorized into three groups based on redness levels (Camacho & Morocho-Cayamcela, 2023), while the second and third models were trained to detect light-red and red tomatoes, respectively. Additional preprocessing was performed on the training dataset before performing training and validation. These techniques are used to increase the diversity of the training data and improve the model's ability to generalize to new data. Performance assessment of the trained models were conducted on the test set, employing four metrics: (1) Precision; (2) Recall; (3) F1-score; and the (4) mean Average Precision (mAP).

doi:10.6342/NTU202401594

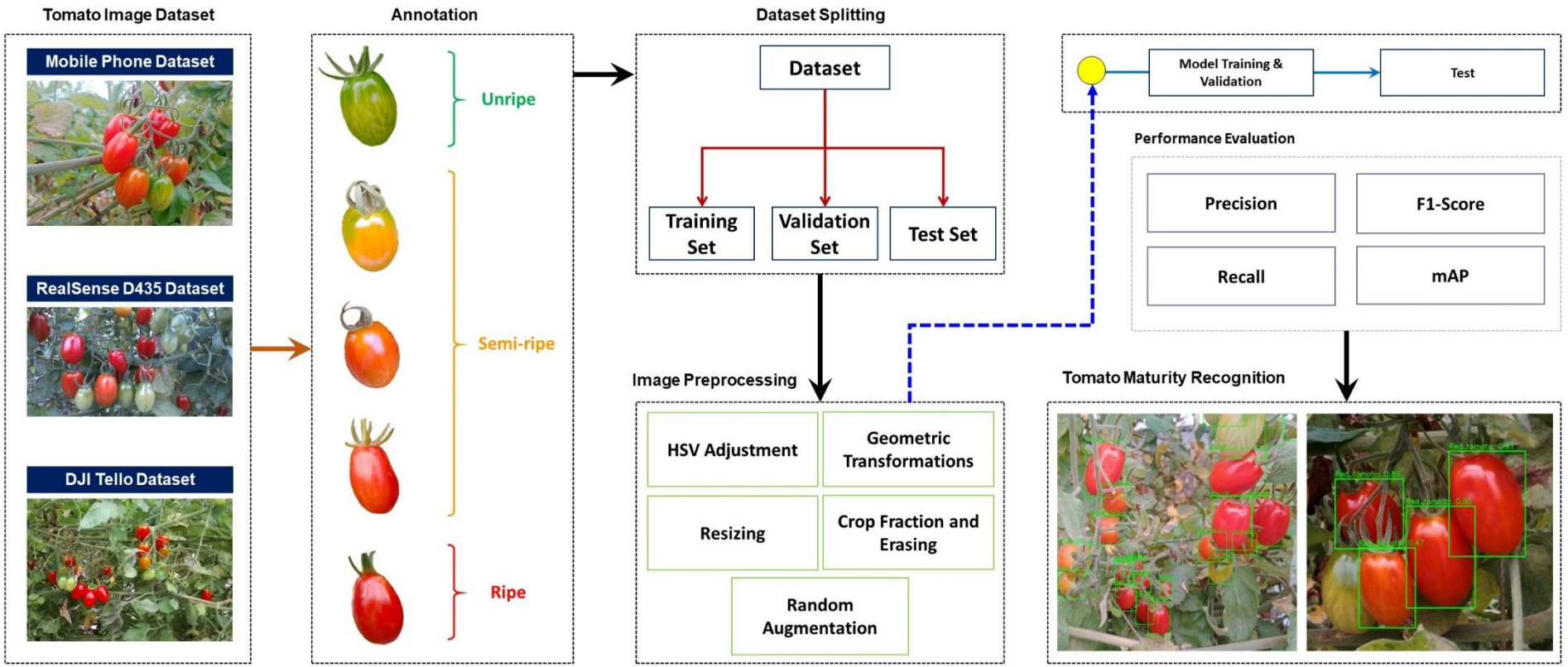


Figure 5. Workflow for the Object Detection Framework. The framework explains the process from left to right, starting from the image acquisition phase, image annotation or labelling, data partitioning, image preprocessing, model training, performance evaluation

doi:10.6342/NTU202401594

3.6.5 Method of Detection

Object detection approaches and techniques can be classified into two (2) types: single-stage detectors and two-stage detectors. In this study, object detection tasks focused on utilizing single-stage detectors since it outperforms two-stage detectors in terms of inference and capability for detecting smaller objects in the scene. The choice for tomato phenotyping in this study leaned towards a single-stage state-of-the-art detector commonly known as YOLO (You Only Look Once), a deep learning architecture renowned for its real-time object detection capabilities. YOLO excels in rapid identification of objects within a single image frame, employing anchor boxes through a single neural network. The YOLOv8n model represents the smallest variant in the YOLOv8 series. Despite trading off some accuracy to prioritize efficiency, this variant is distinguished by its compact model parameters and minimal hardware prerequisites.

YOLOv8 employs a deep neural network architecture with several key layers. As illustrated in Figure 6, the backbone of the architecture is a feature extractor composed of successive Conv and C2f layers, including an SPPF layer at the end, which helps in capturing intricate features from the input image. YOLOv8 uses a detection head with several convolutional layers to improve the accuracy of bounding box predictions and class probabilities. With a focus on speed and real-time applications, YOLOv8 strikes a balance between accuracy and computational efficiency, making it a popular choice for various computer vision tasks.

doi:10.6342/NTU202401594

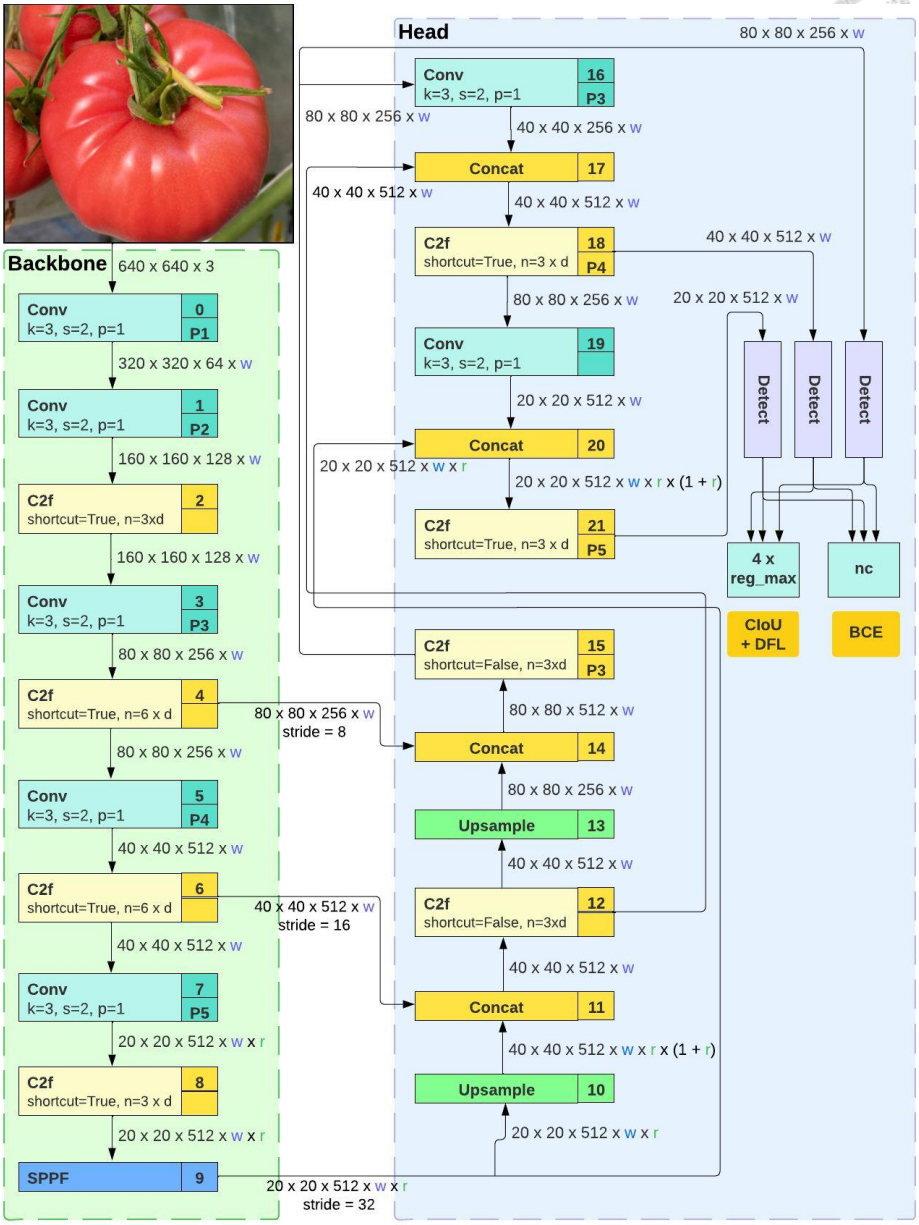


Figure 6. YOLOv8 Architecture (Camacho & Morocho-Cayamcela, 2023)

3.6.6 Ablation and Fine-tuning

In this study, the YOLOv8n network model was utilized and constructed as the baseline model. The network model in this study primarily focuses on improving the performance of the YOLOv8 object detection algorithm through parameter fine-tuning and ablation experiments. To conduct the ablation and fine-tuning experiment, the CA

doi:10.6342/NTU202401594

mechanism was first identified as a key component of the YOLOv8. This ablation phase involved systematically incorporating the CA mechanism to the YOLOv8 model with the aim of achieving more robust and precise localization and identification of cherry tomatoes. The specific principle behind CA mechanism is presented in Figure 7.

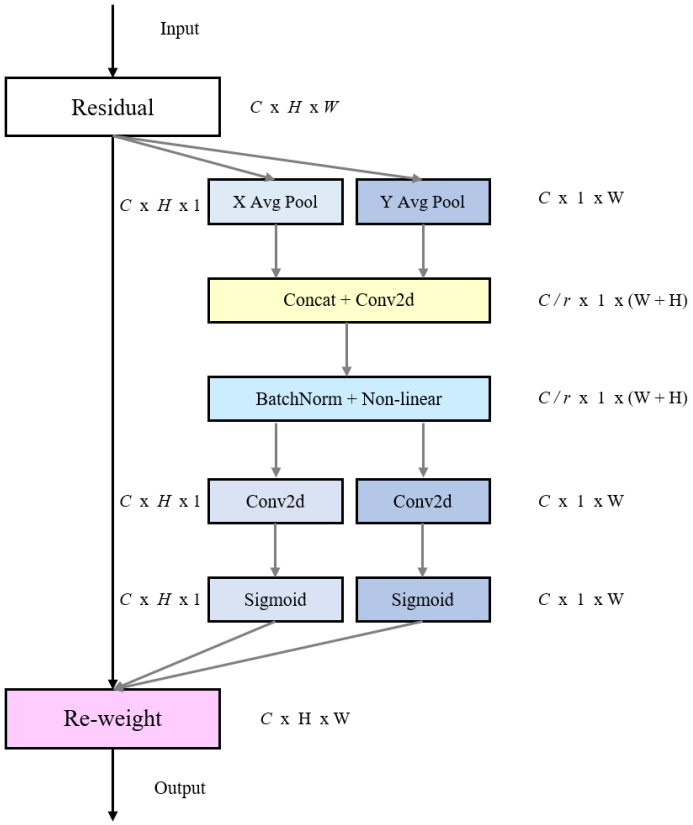


Figure 7. Coordinate Attention (CA) Mechanism (Hou et al., 2021)

The CA mechanism, as elucidated by Hou et al. (2021) and supported by Wang, C. et al. (2023) introduces a novel attention mechanism that integrates positional information into channel attention to enhance spatially selective attention maps. Unlike traditional channel attention methods, CA divides the channel attention process into two 1D feature encoding operations, capturing long-range dependencies along one spatial

doi:10.6342/NTU202401594

direction while preserving precise positional information along the other. This approach results in direction-aware and position-sensitive attention maps that can be applied to input feature maps to enhance object representations.

The mechanism involves two essential steps: coordinate information embedding and coordinate attention generation (Hou, et al., 2021; Wang, C. et al., 2023). In the first step, pooling kernels encode each channel along the horizontal and vertical coordinates, producing direction-aware feature maps. The second step involves the application of a shared convolutional transformation function to the concatenated output of the pooling layers, generating attention vectors for both horizontal and vertical coordinates. These attention vectors are then used to weight the input feature map, enabling accurate object localization. In this experiment, the attention mechanism used was adapted from the repository obtained from GitHub (URL: https://github.com/easyssun/yolov8-with-coordinate_attention). As illustrated in Figure 8, three CA blocks were incorporated at the head of the YOLOv8 architecture. Within the YOLOv8n head architecture, CA Blocks were strategically inserted after the C2f layers in the network’s head to augment the model's comprehension of spatial relationships within images.

doi:10.6342/NTU202401594

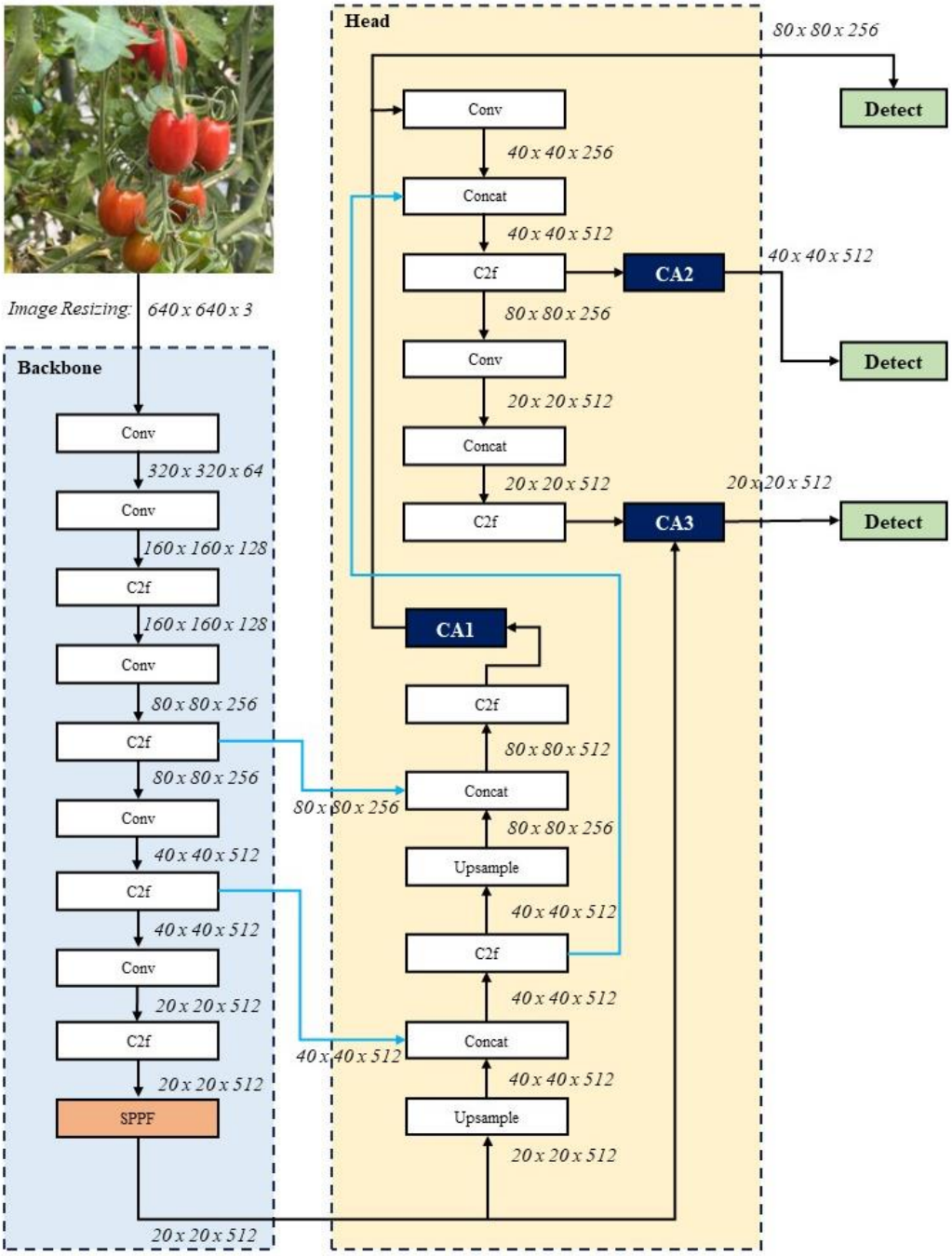


Figure 8. YOLOv8 Architecture with Coordinate Attention Blocks. The layout and design of this figure was adapted from Camacho, J. C., & Morocho-Cayamcela, M. E. (2023) with major modifications.

doi:10.6342/NTU202401594

Following the processing of the head's feature map corresponding to P3/8, CA1 is introduced, succeeded by CA2 after handling the head's feature map P4/16, and finally, CA3 is integrated post-processing the head's feature map P5/32. These blocks enhance the model's ability to focus on important spatial details, improving its performance in detecting objects in images. Moreover, these CA blocks have notable influence on the feature representations extracted by the network, ultimately contributing to the refinement of feature representations utilized by the detection layers.

With the comparative analysis between the base YOLOv8 model and the YOLOv8 with CA, this ablation experiment permitted the analysis of the specific contribution of the attention mechanism to the overall performance of the algorithm. In addition to the incorporation of attention mechanism, the aspect of loss function enhancement was also performed since it was claimed that this function directly affects the speed of training and detection performance of the model (Li, R. et al., 2023).

This adjustment is leaned towards further enhancing the algorithm's overall performance through refinement and reconstruction. In performing the fine-tuning experiments a specific loss function was introduced: (1) BBR with dynamic focusing mechanism or WIoU (Wang, C. et al., 2023). Using a two-layer attention mechanism (Ni et al., 2024), the WIoU is calculated as follows:

$$L_{IOU} = 1 - IOU \tag{3.3}$$

$$L_{WIOUv1} = R_{WIOU} - L_{IOU} \tag{3.4}$$

$$R_{WIOU} = \exp\left[\frac{\left(b_x - b_x^{gt}\right) + \left(b_y - b_y^{gt}\right)}{b_w^2 + b_h^2}\right] \tag{3.5}$$

doi:10.6342/NTU202401594

The coordinates b_x and b_y in Equation 3.5 represent the center of the actual box, while b_x^{gt} and b_y^{gt} represent the center of the prediction box. The equation also encompasses the width (b_w) and height (b_h) of the minimum perimeter rectangle that encloses both boxes.

For all the detection models developed in this study, the analysis included training the YOLOv8 model with different combinations among the CA mechanism and loss functions to determine the most effective settings for object detection tasks. All generated combinations were compared with the baseline (original) model with CIoU (Zheng et al., 2022) to evaluate which enhancement had the most outstanding impact on algorithm's overall detection performance concerning cherry tomato maturity recognition.

3.6.7 Detection Models for Maturity Assessment

3.6.7.1 Monitoring and Surveillance

The first model serves as comprehensive surveillance tool, equipped to detect tomatoes at various stages of ripeness, including unripe, semi-ripe, and fully ripe. This model's versatility enables holistic monitoring of cherry tomatoes throughout their maturation process, facilitating timely interventions such as precise harvesting and autonomous fruit picking. In developing the detection model for this particular purpose, dataset construction and preprocessing were performed.

All the images collected were subjected to data cleaning to remove images with low-quality, resulting to a total of 737 images. It is also worth considering that aside from data cleaning, the efficacy of object detection models depends on the availability of labeled data. This includes the precise identification and spatial delineation of ground truth bounding boxes in images—a process termed as image annotation. In this study, the 'labelImg' graphical annotation tool played a crucial role in defining bounding boxes

doi:10.6342/NTU202401594

around each tomato within the images and assigning them to the aforementioned maturity stage classifications, following the LWYS (Label What You See) approach. To strengthen the robustness of the ground truth data classification, a verification process was performed. During the image acquisition process, an interview with the farm owner was conducted to verify the correctness and robustness of the cherry tomato maturity classification. For highly occluded tomatoes in the images, the bounding boxes were designed based on assumed shapes derived from visible cues, utilizing human intelligence-driven approach.

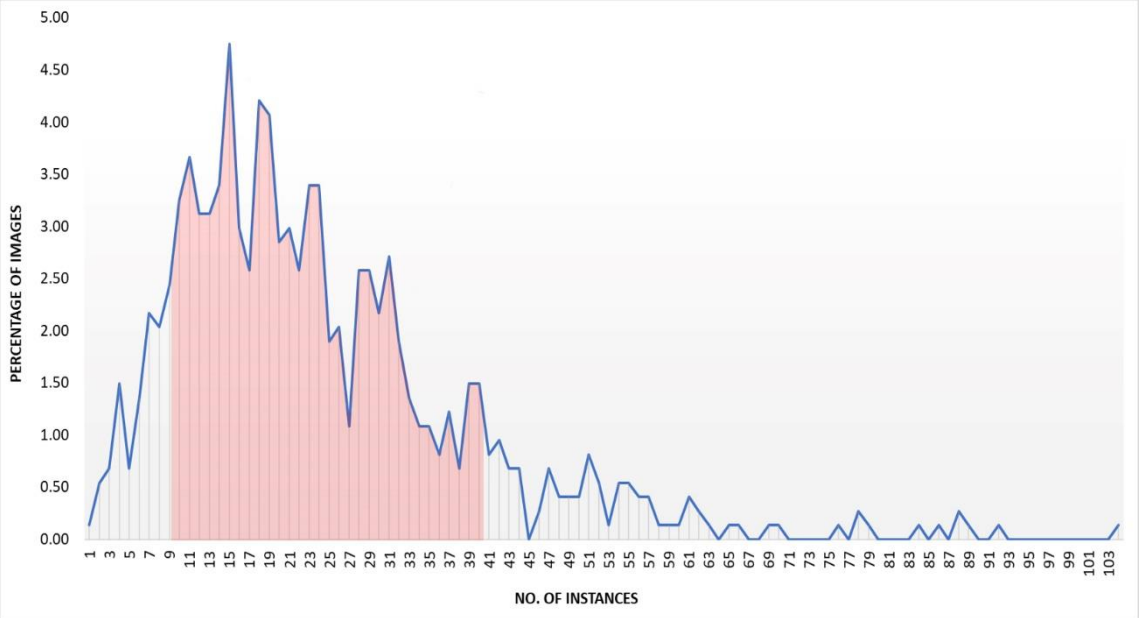


Figure 9. Distribution of Instances in the Dataset

Table 1. Dataset Information

Class \ Device	Mobile Phone	Intel RealSense	UAV	**TOTAL**
Unripe	6,679	1,355	884	8,918
Semi-ripe	2,791	586	354	3,731
Fully-ripened	3,488	816	780	5,084
TOTAL	**12,958**	**2,757**	**2,018**	**17,733**

doi:10.6342/NTU202401594

As shown in Table 1, the dataset used in this study was divided into three classifications: (a) unripe, (b) semi-ripe, and (c) fully-ripened, each with distinct characteristics. These images have 17,733 samples consisting of 8,918 unripe, 3,731 semi-ripe, and 5,084 fully-ripened cherry tomato images were utilized. In addition, Figure 9 further illustrates the distribution of instances across the entire dataset. Majority of the images contains instances ranging from 10 to 40, and less than 1 % of the images contains 50 to 104 instances. This information indicates some variance in the dataset because the presence of higher instances counts (50 to 104) can contribute to the diversity and scalability of the dataset, potentially allowing the model to learn and generalize to a broader range of scenarios. After successfully annotating the dataset, the images and labels were divided into three sets with a 7:1:2 split ratio resulting in 515 images for the training, 73 images for the validation, and 149 images for the test set.

3.6.7.2 Cherry Tomato at Light-red Stage

The tomato fruit exhibits a climacteric nature, allowing it to be harvested at an earlier maturity stage and subsequently ripen post-harvest, even after detachment from the plant (El-Bendary et al., 2015; Tsouvaltzis et al., 2023). Thus, the optimal maturity level for harvesting cherry tomatoes may depend on several factors, including their intended use and storage conditions. For example, cherry tomatoes that will be transported and stored for longer periods may benefit from being harvested at slightly less mature stage to prevent over-ripening and spoilage. On the other hand, cherry tomatoes intended for immediate consumption or short-term storage benefit from being harvested at a more mature stage. By focusing specifically on this stage, the model aims to enhance precision in harvesting operations, thus optimizing overall crop productivity and minimizing post-harvest losses.

doi:10.6342/NTU202401594

Developing this detection model involved utilizing a methodology similar to that applied in developing the model for monitoring and surveillance (Section 3.6.7.1). However, the number of samples representing light-red maturity stage initially appears to be insufficient compared to other stages such as green and red. Hence, supplementary data augmentation strategy was performed to increase the quantity of samples. Figure 10 showcases the implementation of a random cropping technique, generating four distinct images, each standardized to 2000 x 2000 pixels, from the original dataset.

Table 2. Size of the Dataset Before and After Data Augmentation

	Before Augmentation	**After Augmentation**
Train set	387	708
Validation set	55	98
Test set	113	193
TOTAL	**555**	**999**

Similar to Section 3.6.7.1, all augmented outputs were subjected to a thorough data cleaning process to eliminate images lacking the light-red cherry tomato samples, resulting to a total image quantity of 999 images. Table 2 further elucidates the size of the dataset before and after the augmentation procedure. Furthermore, the newly established dataset (Table 3) was re-annotated and was utilized for training the YOLOv8 model exclusively on light-red cherry tomato samples. Similarly, the dataset from different modalities (mobile phone, UAV, and RealSense D435) was divided into three sets with 7:1:2 split ratio resulting in 708 images for training, 98 images for the validation, and 193 images for the test set.

doi:10.6342/NTU202401594

Table 3. Quantity of Images and Samples for Light Red Cherry Tomatoes

Set type \ Quantity	No. of Images	No. of Samples
Train set	708	1,810
Validation set	98	260
Test set	193	513
TOTAL	**999**	**2,583**

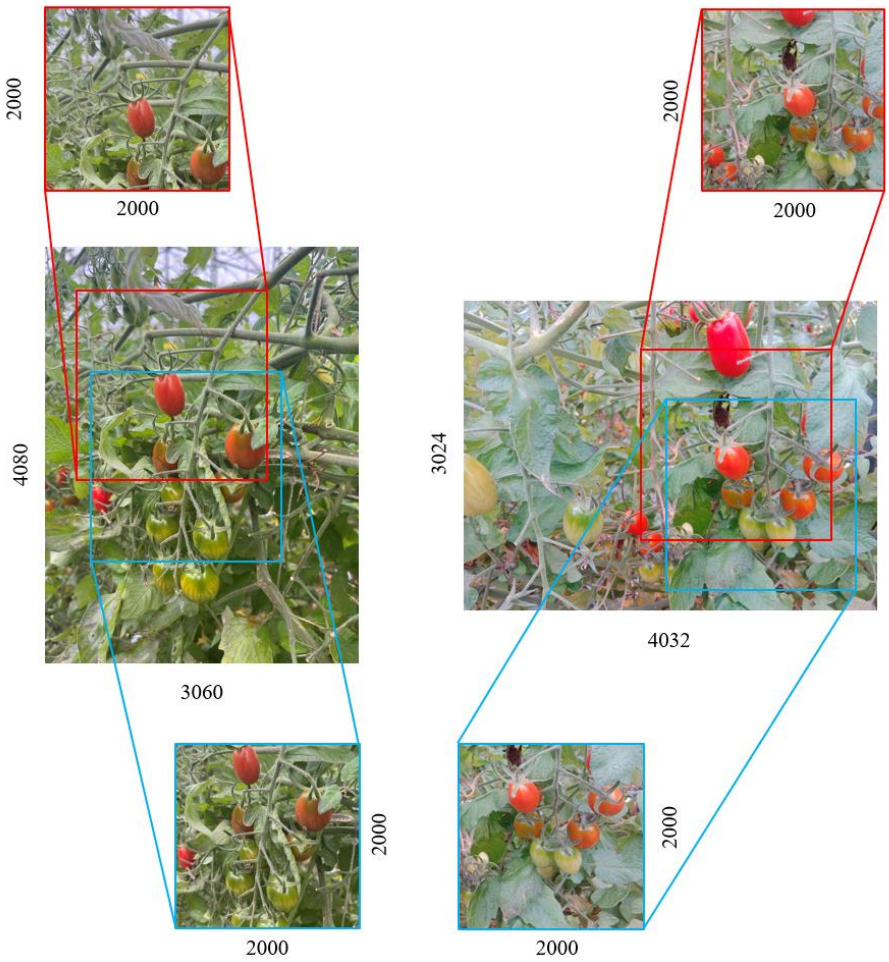


Figure 10. Data Augmentation Strategy for Light-red Tomato Detection Model using Mobile Phone Images. The image demonstrates the data augmentation technique where random cropped images highlighted by red and blue rectangles are generated. These cropped sections are included in the input dataset to increase its quantity and variability, thereby enhancing the model's training process.

doi:10.6342/NTU202401594

3.6.7.3 Cherry Tomato at Fully Mature (Red) Stage

Similar to the model developed for light-red cherry tomato detection, the original dataset used for training, validating, and testing the detection model for monitoring and surveillance was also used for this purpose. The dataset was re-annotated for the third time to be utilized for training the YOLOv8 model to exclusively detect red cherry tomatoes. Moreover, the dataset was also divided into three sets with the same split ratio resulting in 504 images for training, 70 images for the validation, and 144 images for the test set. Since the dataset is abundant with fully-ripe cherry tomato samples as evident in Table 4, additional data augmentation processes were not applied.

Table 4. Quantity of Images and Samples for Red Cherry Tomatoes

Set type \ Quantity	**No. of Images**	**No. of Samples**
Train set	504	4,444
Validation set	70	630
Test set	144	1,372
TOTAL	**718**	**6,446**

3.6.8 Multi-Source Data Training

To evaluate the efficacy of utilizing images from various devices—specifically, Tello drone, depth camera, and mobile phone—in training the Deep Learning Object Detection Model, a systematic and supplementary methodology was employed. Three distinct combinations were tested to determine the impact of using dataset consisting of diverse image natures (Table 5). The YOLOv8n object detector were trained using these combinations while adapting the identical process elucidated in the next section (Section 3.6.9).

doi:10.6342/NTU202401594

Table 5. Details and Information of the Dataset for Multi-Source Data Training

Training Configuration	Source	No. of Images	No. of Samples
A	Mobile Phone	417	10,155
B	Mobile Phone + Tello Drone	510	14,585
C	Mobile Phone + Tello Drone + Intel RealSense D435	588	13,934

Prior to training Object Detection models, an examination of the color spaces of images from three devices were carefully inspected to identify the consistency and uniformity of the quality of the images. Pixel values from sample images were analyzed, as depicted in Figure 11. The selection of Regions of Interest (RoIs) emphasized green and red colors, reflecting the dominant hues in greenhouse environments. Additionally, the primary objective was to assess the uniformity of the image datasets in terms of their color spaces, making the size of the RoIs irrelevant.

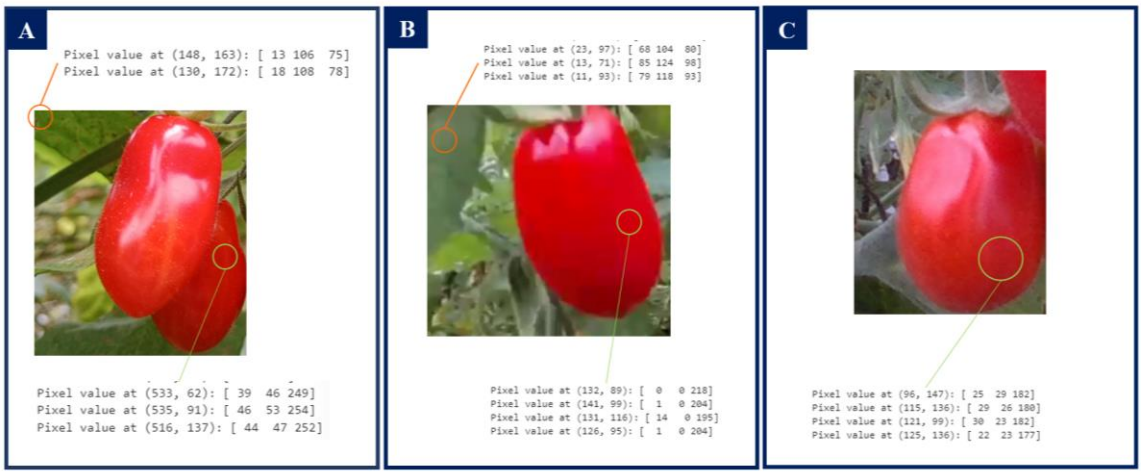


Figure 11. Inspection of Color Spaces through Pixel Value Extraction: (A) image from mobile phone; (B) image from Tello drone; and (C) image from Intel RealSense D435.

The examination revealed a unifying factor among images taken with three different devices: they are all in the BGR (Blue-Green-Red) color space. This

doi:10.6342/NTU202401594

harmonization in color representation across devices serves to alleviate concerns related to image variations that could compromise detection accuracy, effectively minimizing the likelihood of performance issues. Following the training phase, each model underwent rigorous testing using diverse test set combinations which are elaborated in Table 6. By systematically analyzing the results obtained from each test set combination, insights were gained into the comparative strengths and weaknesses of utilizing images from different devices in training deep learning object detection models. This protocol provides valuable guidance for recommending an effective strategy for utilizing diverse image sources in enhancing the performance and robustness of such models.

Table 6. Details and Information of the Dataset Configured for the Test Set

Test Configuration	Source	No. of Images	No. of Samples
A	Mobile Phone	105	2,803
B	Tello Drone	24	391
C	Intel RealSense D435	20	605
D	Mobile Phone + Tello Drone + RealSense D435	149	3,799

3.6.9 Model Training, Validation, and Testing

The detectors were trained using a set of hyperparameters elaborated in Table 7 and their performance was evaluated using the test set; the train-validation-test ratio was elucidated in Section 3.6.4. and 3.6.7. This set of parameters was adapted from Li, P. et al. (2023) as it generated better satisfactory results compared to the parameter settings retrieved from Li, R. et al. (2023) based on preliminary iterative and empirical experiments conducted in this study.

Preliminary experiments revealed that higher precision, recall, F1-score, and mAP can be achieved by using the parameters from Li, P. et al. (2023). Hyperparameter set 2

doi:10.6342/NTU202401594

(Li, P. et al., 2023) demonstrates several advantageous hyperparameter configurations that potentially enhance its performance compared to hyperparameter set 1 (Li, R. et al., 2023). Specifically, the initial learning rate and lower learning rate factor in set 2 facilitates a more gradual reduction in the learning rate, promoting a stable and refined training process that can achieve superior performance. Moreover, the slightly higher weight decay in set 2 serves as a more effective regularization mechanism, helping to mitigate overfitting and improve generalization. Furthermore, warmup parameters of set 2, including a slightly longer warmup period, lower warmup momentum, and a marginally lower warmup bias learning rate, contribute to more stable and less aggressive parameter updates during the initial training phase, thereby reducing the risk of overshooting and enhancing training stability. Collectively, these hyperparameter choices make set 2 better suited for achieving robust and generalizable performance compared to set 1. In addition, it became evident after conducting multiple iterative experiments that batch size of 8 delivers the best performance, therefore, this parameter is utilized in training the network model.

In this study, the Stochastic Gradient Descent (SGD) optimizer was used for accelerating the training of the neural network. Furthermore, efficiency of training was aimed to be improved by integrating an early stopping strategy through the use of the patience parameter. In the occasion where no model improvement was observed in the last 50 epochs, the training process will stop. In addition, other pre-processing parameters such as mosaic data augmentation were used to further enhance the training process, thereby resulting to an improved generalization capability and performance of the model. Sample images from the mosaic augmentation technique are shown in Figure 12.

doi:10.6342/NTU202401594

Table 7. Hyperparameter Settings

Hyperparameter	Value	Hyperparameter	Value
epochs	400	cls	0.5
batch	8	hsv (h)	0.015
patience	50	hsv (s)	0.7
confidence	0.5	hsv (v)	0.4
lr0	0.01	translate	0.1
lrf	0.01	scale	0.5
momentum	0.937	flip (left and right)	0.5
weight decay	0.0005	mosaic	1.0
warmup epochs	3.0	erasing	0.4
warmup momentum	0.8	crop fraction	1.0

Figure 12. Sample Images from the Mosaic Augmentation Technique employed in this Study

3.6.10 Performance Evaluation

The performance of YOLOv8n was evaluated using the test set. True Positives (TP) were categorized and selected as a detection instance if the Jaccard Index similarity

doi:10.6342/NTU202401594

coefficient known as Intersection-over-Union (IoU) (Afonso et al., 2020) is 0.7 or more with the ground truth instance. Given the measures of True Positives (TP), False Positives (FP), and False Negatives (FN), the following measures were calculated:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3.6}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3.7}$$

$$\text{F1-score} = \frac{2(Precision * Recall)}{Precision + Recall} \tag{3.8}$$

$$\text{mAP} = \frac{\sum_{i=1}^{C} AP_i}{C} \tag{3.9}$$

Where:

AP =Average Precision

3.7 Detection and Tracking in Greenhouse Environment

According to Rong et al. (2023), Multiple Object Tracking (MOT) is designed to dynamically identify and follow numerous objects of interest within a video sequence. This technique enables trajectory monitoring by flexibly linking multiple objects across consecutive frames without requiring prior knowledge of the exact number of targets present in the scene. Tracking-by-Detection is the foundation of the majority of currently existing MOT algorithms. This technique involves employing an object detector on a sequence of consecutive frames to determine the bounding boxes of objects. Subsequent steps involve calculating diverse features encompassing both visual and motion aspects for the identified objects. These features are then utilized to assess the similarity between

doi:10.6342/NTU202401594

objects across adjacent frames, facilitating the assignment of unique identifiers to each object.

Figure 13. Data Collection using UAV: (a) UAV's Planned Path; (b) UAV during Actual Flight Mission

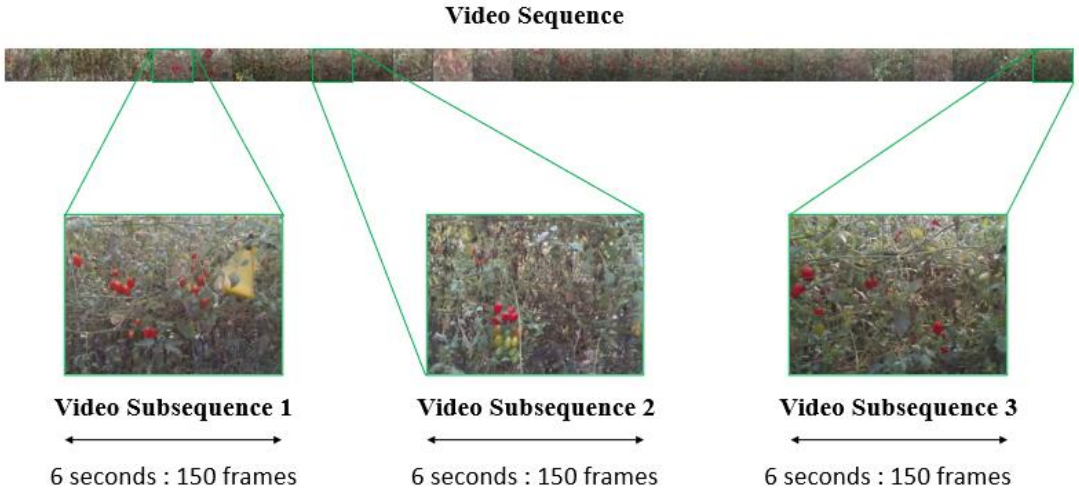


Figure 14. Video Partitioning and Selection Strategy

In computer vision, when it comes to tracking objects like tomatoes, the differences between the objects are subtle which makes them harder to track in a more precise and accurate manner. Furthermore, research into tomato cluster tracking found

doi:10.6342/NTU202401594

that even with the implementation of ByteTrack algorithm, the phenomenon of ID switching remained prevalent, indicating inconsistencies in the tracking process (Rong et al., 2023). Thus, BoT-SORT was selected as the tracking algorithm in this study.

In conducting the experiment, single row of plant in a greenhouse environment (Figure 13) was utilized for detecting and tracking ripe tomatoes using the Tello drone. The UAV was operated manually to efficiently cover the area, while recording livestream data at 25 frames per second (FPS) and encoding the frames into the universally compatible MP4 format. The video sequence was partitioned (Figure 14) into several sections and subsequences, each with a duration of 6 seconds (150 frames). Due to the labor-intensive nature of ground truth annotation, only three subsequences were selected for performance evaluation. The selected sequences represent three different scenarios: (1) multiple cluster tracking from varying camera distances; (2) multiple cluster tracking from fixed camera distance; and (3) single cluster tracking from fixed camera distance.

The selected sequences were annotated using the MOT-Annotator tool retrieved from a GitHub repository (URL: https://github.com/khalidw/MOT16_Annotator.git). Minor modifications were incorporated into the tool to allow the user to manually assign the IDs of the labelled objects. Afterwards, ripe tomatoes were detected and tracked from the sequences using the trained YOLOv8n + CA + WIoU and BoT-SORT algorithms, respectively. To provide a more comprehensive comparison, the ByteTrack algorithm was also applied to track ripe cherry tomatoes on the video subsequences. Both annotations and actual tracking results were saved in a MOT16 format for uniformity and compatibility with the performance calculations. Lastly, two specific metrics were adapted and utilized in evaluating the reliability of the algorithm in detecting and tracking ripe tomatoes. These metrics, elaborated in Equations 3.10 and 3.11, encompass the Multi-Object Tracking Accuracy (MOTA) (Rong et al., 2023; Wang, Z. et al., 2023) and

doi:10.6342/NTU202401594

the IDF1 (Wang, Z. et al., 2023). While being calculated based on FN, FP, and IDs, MOTA places greater emphasis on detection performance since FP and FN amounts are greater than IDs (Zhang, Y. et al., 2022). The other metric, IDF1, is defined as the ratio of accurate recognition detection to the average real number and the computed detection number. The Python implementation of the MOT metrics retrieved from GitHub (URL: https://github.com/cheind/py-motmetrics.git) was used in conducting the performance evaluation.

$$\text{MOTA} = 1 - \frac{\sum_t FN_t + FP_t + IDSW_t}{\sum_t GT_t} \tag{3.10}$$

$$\text{IDF1} = \frac{2 * IDTP}{2 * IDTP + IDFP + IDFN} \tag{3.11}$$

Where:

IDTP = True Positive ID Number

IDFP = False Negative ID Number

IDFN = False Negative ID Number

GT_t = Total Ground Truth Instances

After successfully establishing the tracker, the fine-tuned system was utilized to count ripe cherry tomatoes present in three selected video sequences specified in this section. In the future, the ripe cherry tomato counting system aims to provide preliminary data support for yield estimation. To evaluate the accuracy and performance of the counting system, the Counting Error Rate (CER) and Counting Accuracy (CA) metrics were employed. The Counting Error Rate (CER) is determined using the following equation:

doi:10.6342/NTU202401594

$$\text{Counting Error Rate (CER)} = \left|\frac{C_{estimated} - C_{ground\ truth}}{C_{ground\ truth}}\right| \quad 3.12$$

where $C_{estimated}$ is the estimated count of ripe cherry tomatoes, and $C_{ground\ truth}$ is the actual count. Concurrently, the Counting Accuracy (CA) is calculated as:

$$\text{Counting Accuracy (CA)} = 1 - \text{CER} \quad 3.13$$

These metrics provide a quantitative assessment of the system's counting accuracy and error rate, thereby enabling a thorough evaluation of its performance in estimating yield.

3.8 Impact of Distance and Confidence on Detection Performance

This present study also investigates two variables: (a) the distance between the camera and the object of interest; and the (b) confidence thresholds, and their impact on the detection performance of the deep learning model. This experiment aims to provide insights and determine the optimized parameters necessary in performing screening and validation phases using a UAV system. The screening stage was designed to have high sensitivity to avoid missing instances of ripe tomatoes, despite the consequence of having lower specificity. On the other hand, the secondary validation stage was designed to have high specificity and low sensitivity, thereby avoiding false positives. The major goal of the screening stage is to cast a wide net, capturing as many potential ripe tomatoes as possible, even if it includes some false positives. Following the screening stage is the validation process which scans the same area with more stringent and rigorous analysis to confirm the definitive presence or absence of ripe tomatoes. By integrating these two-step approach, the detection of ripe tomatoes can be further optimized, balancing speed and precision to deliver high-quality results.

doi:10.6342/NTU202401594

For this set of investigation, the protocol of Bhusnoor et al. (2023) was adapted. The study offers a straightforward, practical, and logical approach for determining the effect of the object-to-camera distance on the confidence scores of the detection outputs. This phase of the investigation was conducted to further validate the outcome of the study (Bhusnoor et al., 2023) with the same hypothesis about the correlation between the confidence percentage (accuracy) of object detection and the increasing distance between the UAV and target object.

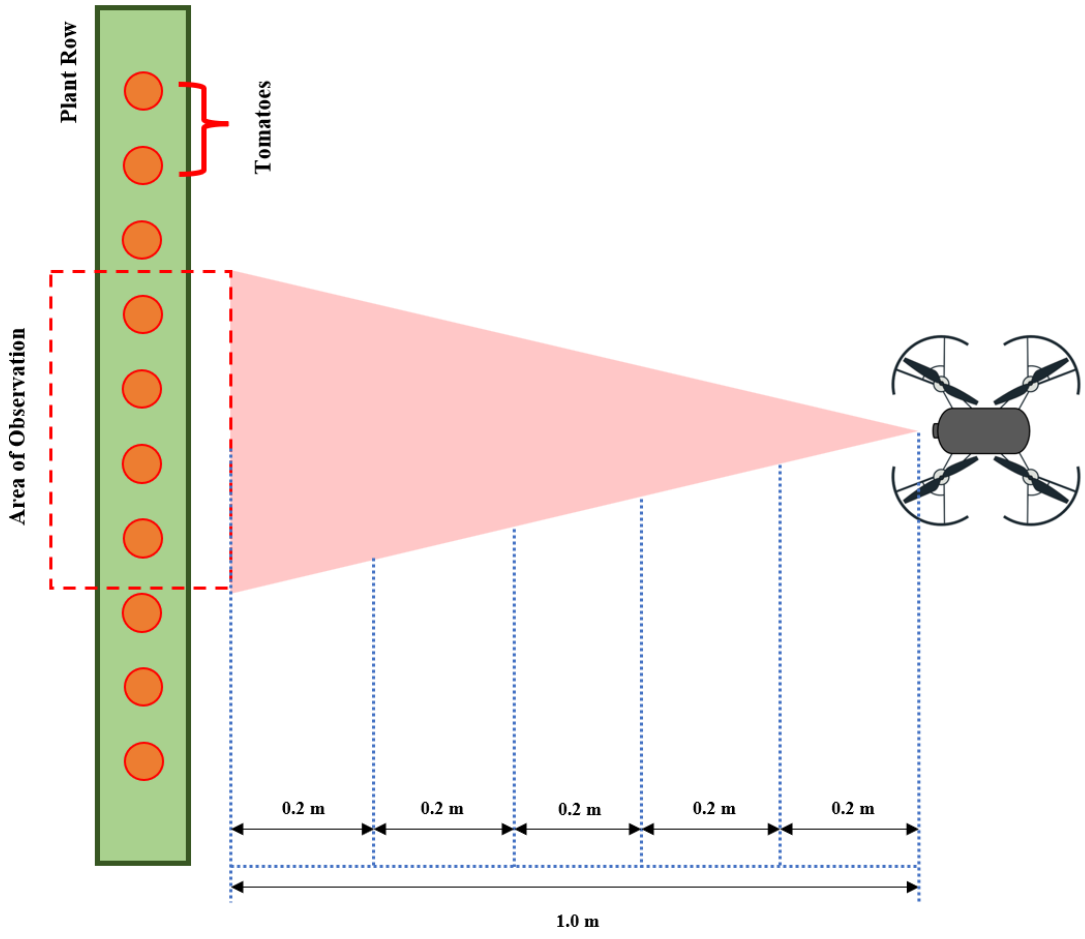


Figure 15. Experimental Design for Investigating the Impact of Distance on the Performance of the Detection Models

doi:10.6342/NTU202401594

Figure 16. Determination of the Influence of Distance on the Object Detection Performance: (A) Actual Setup; (B) Sample Frame (at 0.2 m OTC Distance); (C) Sample Frame (at 0.4 m OTC Distance); (D) Sample Frame (at 0.6 m OTC Distance); (E) Sample Frame (at 0.8 m OTC Distance); and (F) Sample Frame (at 1.0 m OTC Distance)

doi:10.6342/NTU202401594

Video streams were retrieved while flying the UAV in stationary position, capturing samples at distances of 0.2, 0.4, 0.6, 0.8, and 1.0 meter from the target objects within the specified range (Figure 15). The actual setup of video acquisition, including the actual frames obtained during the data collection process were illustrated in Figure 16. Throughout this process, the confidence percentages of object detection for the target cherry tomatoes were measured and observed, providing valuable insights into the efficiency and accuracy of the detection system under different distances.

Concurrently, the set of data used for MOT evaluation at Section 3.7 were also utilized for determining the relationship between the confidence threshold and the performance of the object detector. Different confidence thresholds were incorporated to the detection model's tracking algorithm to gain further insights about the detection capability of the object detector in the context of ripe tomato detection. These thresholds will range from 0.4 to 0.8, with intervals of 0.1 between each threshold. Moreover, the ID Switching instances, MOTA, and IDF1 were utilized as evaluation metrics, resulting to three different performance curves: (1) Confidence Threshold vs. ID Switch; (2) Confidence Threshold vs. MOTA; and (3) Confidence Threshold vs. IDF1 curves (Figure 17).

Through systematic adjustment of the aforementioned thresholds, insights into the model's performance across a spectrum of confidence levels will be gained, providing valuable insights into its detection accuracy.

doi:10.6342/NTU202401594

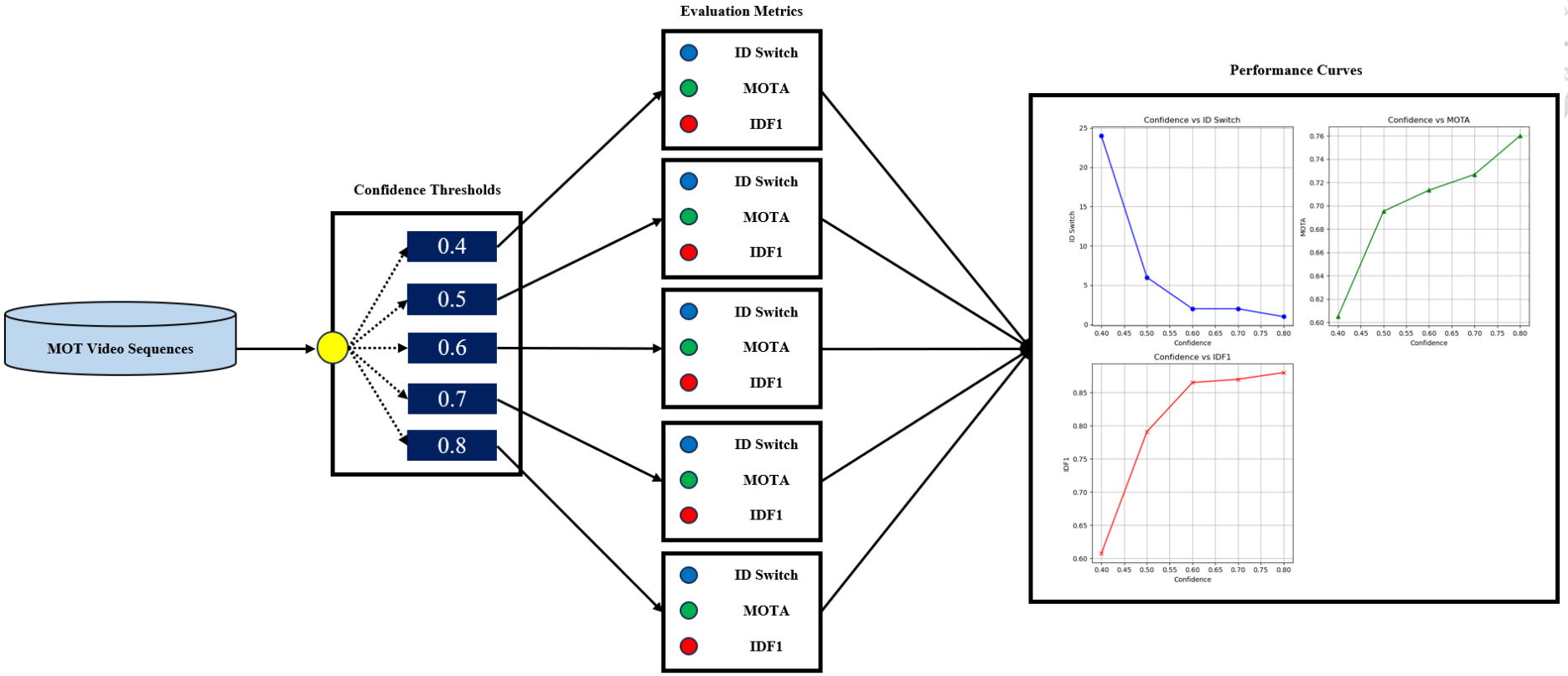


Figure 17. Overall Process for Determining the Influence of Different Confidence Thresholds on the Detection Performance of the Deep Learning Model

doi:10.6342/NTU202401594

Chapter 4. Results and Discussion

4.1 Communication Protocols Between Drones and Computers

The successful implementation of drone-computer communication through the utilization of UDP port communication and the pyGame GUI, as elaborated in Section 3.4, proved to be highly successful in this study. The use of UDP facilitate considerably efficient data transmission between the drone and the computer. Moreover, it provided reliable communication channel between the drone and the computer, ensuring the swift exchange of data packets without the need for connection-oriented setup. This communication framework highlighted the effectiveness of this approach in achieving reliable and responsive drone control.

However, upon the execution of “t” key (key for the take-off command), the system experiences a delay of roughly 1.5 to 2 seconds in the context of camera to computer server image frame transfer. This situation may be attributed to factors such as buffering and network latency. Overall, the GUI enabled efficient handling of the keystrokes which guaranteed successful data acquisition, smooth control, and on-point intervention in case of dangerous or unprecedented situations across all experiments despite the aforementioned data transmission delay.

The findings highlight the effectiveness of utilizing a drone-computer communication method using the UDP communication in conjunction with a simple GUI for controlling the Tello EDU drone. UDP, known for its simplicity and speed, facilitated connectionless transmission of commands and data between the drone and the computer, contributing to efficient drone control (Christensen et al., 2020). The lightweight nature of UDP, particularly suitable for small embedded systems, proved instrumental in maintaining responsive communication without the need for complex setup procedures.

doi:10.6342/NTU202401594

Through this protocol, the receiver remains attentive on a specified UDP port, ready to accept incoming data whenever it arrives, while the transmitter has the freedom to send data whenever it desires. Concurrently, the GUI not only assisted in visualizing and interacting with the data being exchanged but also facilitated real-time interventions which thereby improve overall UAV control and maneuverability.

In spite of the communication framework's overall effectiveness, it is crucial to acknowledge the data transmission delay observed upon the execution of the take-off command. Prior studies, such as those by Bhujbal & Barahate (2022) and Boonsongsrikul & Eamsaard (2023), effectively utilized the image transmission interface and camera video recording functionalities of the default DJI Tello Python library. However, these works did not involve any manual control mechanisms, thus requiring no modifications to the original Tello library. In contrast, this present study integrates additional controls and features, necessitating modifications to the library's functionality in order to accomplish the primary objectives. Moreover, the data transmission in this study is point-to-point, which means data is directly transmitted from the sender to the receiver without intermediary devices, minimizing data security issues and potential interference.

Furthermore, the current system focuses on retrieving video stream recordings and performing post-processing tasks which involves cherry tomato detection and tracking. Since real-time operations are not an immediate requirement, the delay in data transmission does not impede the system’s overall functionality. However, for future scenarios necessitating real-time surveillance and monitoring, it is crucial to conduct experimental tuning to address the existing delay.

doi:10.6342/NTU202401594

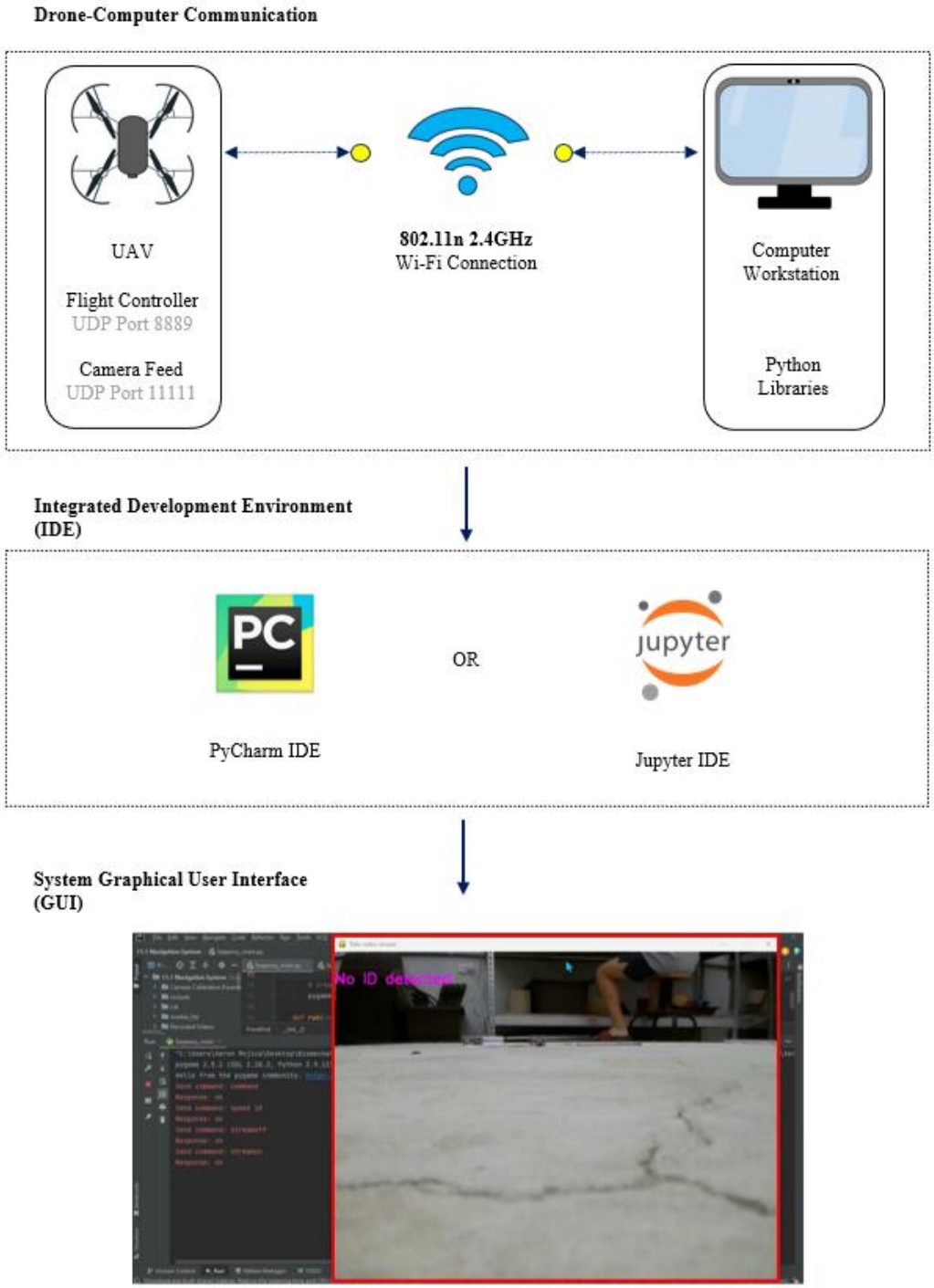


Figure 18. Drone-Computer Communication Mechanism. The area highlighted with red color at the image represents the Real-time Video Stream and GUI of the System

doi:10.6342/NTU202401594

4.2 Multi-Source Data Training

In this experiment, the possibility of combining datasets from different sources or devices to train deep learning models was investigated. This was conducted to solve the hypothesis that using multimodal imaging approach can enhance the generalization performance of CNN architectures due to the diverse nature of the input data. Table 8 ~ 11 presents the performance of all the YOLOv8 model training configurations across all the test sets explained in Section 3.6.8. The models were trained for recognizing and classifying the identification classes intended for the first detection model of this study (see Section 3.6.2). Moreover, the YOLOv8 was trained using the set of hyperparameters elaborated in Section 3.6.9. Furthermore, the CA blocks were incorporated and the WIoU loss function was utilized since preliminary trials and a previous study (Wang, C. et al., 2023) proved the general effectiveness of these combinations in performing maturity recognition tasks.

The analysis of performance metrics across various training configurations (Table 5) on multiple test sets (Table 6) reveals significant insights into the adaptability and efficacy of each configuration (Tables 8 ~ 11). The model trained using images from the three modalities (Training Configuration C) consistently performs better than those trained on images from the mobile phone (Configuration A) and images from mobile phone and Tello drone (Configuration B) by demonstrating higher mAP and F1-score. This finding implies that Configuration C is highly sensitive to the nuances and difficulties unique to Test Set A. Upon extending the investigation to other test sets, it still emerged as the preeminent detection model, displaying superior performance in all aspect among all configurations. The findings further reveal that the adaptability of Training Configuration C, proves its ability to generalize well on unseen data, which thereby demonstrates its resilience in a variety of test scenarios.

doi:10.6342/NTU202401594

Table 8. Performance of the Detection Models on Test Set A

	Precision	Recall	F1-Score	mAP
Training Configuration A	83.0	85.1	84.04	89.6
Training Configuration B	81.8	85.5	83.61	89.6
Training Configuration C	**84.0**	**84.5**	**84.25**	**89.7**

Note: Highest values for each performance metric are highlighted in bold.

Table 9. Performance of the Detection Models on Test Set B

	Precision	Recall	F1-Score	mAP
Training Configuration A	76.0	76.6	76.30	80.6
Training Configuration B	82.5	82.2	82.35	89.7
Training Configuration C	**84.1**	**83.8**	**83.95**	**90.6**

Note: Highest values for each performance metric are highlighted in bold

Table 10. Performance of the Detection Models on Test Set C

	Precision	Recall	F1-Score	mAP
Training Configuration A	75.2	78.9	77.01	82.5
Training Configuration B	**80.4**	78.1	79.23	84.2
Training Configuration C	77.9	**83.6**	**80.65**	**87.6**

Note: Highest values for each performance metric are highlighted in bold

Table 11. Performance of the Detection Models on Test Set D

	Precision	Recall	F1-Score	mAP
Training Configuration A	82.0	81.7	81.85	87.7
Training Configuration B	81.7	83.7	82.69	88.7
Training Configuration C	**84.1**	**83.1**	**83.60**	**89.4**

Note: Highest values for each performance metric are highlighted in bold

Through this experiment, it became evident that integrating images from diverse sources and devices could potentially enhance the efficiency and generalization capability of an object detection model. Based on the findings, the concept of utilizing images taken using different devices which was adapted from Kung et al. (2021) was further proved to be effective in improving model performance. Despite the clear variation in size and resolution of the images in the training set, the third model for this experiment which was trained using images from three different modalities achieved remarkable accuracy and

doi:10.6342/NTU202401594

recall rates of 77.9% ~ 84.1% and 83.1% ~ 84.5%, respectively. The concept of this analysis may help solving the assumption which states that integrating images from diverse sources and devices could potentially enhance the efficiency and generalization capability of an object detection model. Nevertheless, it is imperative to conduct further validations encompassing a broader spectrum of factors to strengthen the claims and verify the outcomes such as the use of uniform quantity of images in each dataset configurations. This ensures equitable representation from diverse sources and devices, facilitating fair evaluation of model performance across varying data inputs.

4.3 Ablation Study and Fine-tuning

To evaluate the effectiveness of the improvements incorporated in the base YOLOv8n, a series of comparative analysis was conducted on the cherry tomato dataset before and after the model was enhanced and fine-tuned. This investigation sought to evaluate how well each of the trained models performed over variety of conditions and cherry tomato maturity stages. The performance of improved YOLOv8n model was compared with the base model in the context of overall monitoring and surveillance (Tables 12), light-red cherry tomato (Tables 13), and red cherry tomato (Tables 14) detection.

Table 12. Comparison of Detection Results for Overall Surveillance and Monitoring Models Before and After the Enhancements

	Precision	**Recall**	**F1-Score**	**Overall mAP**
YOLOv8n	80.8	82.2	81.49	87.5
YOLOv8n + CA	83.6	82.3	**82.94**	**89.0**
YOLOV8n + WIoU	**84.1**	81.4	82.73	87.6
YOLOV8n + CA + WIoU	81.4	**82.8**	82.09	88.1

Note: Highest values for each performance metric are highlighted in bold.

doi:10.6342/NTU202401594

Table 13. Comparison of Detection Results for Light-red Tomato Models Before and After Enhancement

	Precision	Recall	F1-Score	mAP
YOLOv8n	**83.5**	79.9	81.66	87.4
YOLOv8n + CA	82.2	80.1	81.14	88.1
YOLOV8n + CA + WIoU	81.2	**83.1**	**82.14**	**88.3**

Note: Highest values for each performance metric are highlighted in bold.

Table 14. Comparison of Detection Results for Red Tomato Models Before and After Enhancement

	Precision	Recall	F1-Score	mAP
YOLOv8n	90.3	87.1	88.67	93.1
YOLOv8n + CA	**90.9**	87.0	**88.91**	93.3
YOLOV8n + CA + WIoU	90.2	**88.5**	89.34	**93.7**

Note: Highest values for each performance metric are highlighted in bold.

Table 12 presents the performance of YOLOv8n trained with different combinations of CA and loss function. Based on the results, YOLOv8n with CA mechanism displayed the most remarkable performance. Despite YOLOv8n + WIoU achieving the highest precision, and YOLOv8n + CA + WIoU achieving the highest recall rate, YOLOv8n + CA demonstrated superiority as evident from its F1-score and mAP. These values indicate a more balanced trade-off between precision and recall, resulting in a more robust and reliable performance metric and an overall improvement in the model's ability to accurately localize and classify objects across various categories. The enhancement in performance observed with the incorporation of coordinate attention (CA) suggests that the inclusion of attention mechanisms can effectively improve object detection accuracy. By integrating positional information into channel attention, CA offers a novel approach to enhancing spatially selective attention maps. The outcomes of this enhancement emphasize the consistent trend observed across all the tomato maturity stages. For this ablation study, the results of incorporating CA blocks to YOLOv8n

doi:10.6342/NTU202401594

demonstrated partially similar results to Wang, C. et al. (2023) who utilized YOLOv5n with CA mechanism. In both studies, the model equipped with CA demonstrated improved precision, F1-score, and mAP, but exhibited a contrasting outcome in recall. However, no direct comparisons can be performed since different datasets were used in training and testing the models from both studies. Evidently, it was proven that the integration of CA into the YOLOv8n architecture yielded even better enhancements across all evaluated parameters. This signifies that the improved model has the ability to identify and locate the target objects more accurately by extracting features from cherry tomatoes and effectively filtering out unnecessary information by concentrating on key feature channels.

Furthermore, the loss curves of the training phase before fine-tuning (YOLOv8n) and after fine-tuning (YOLOv8+WIoU) was illustrated (Figure 19).

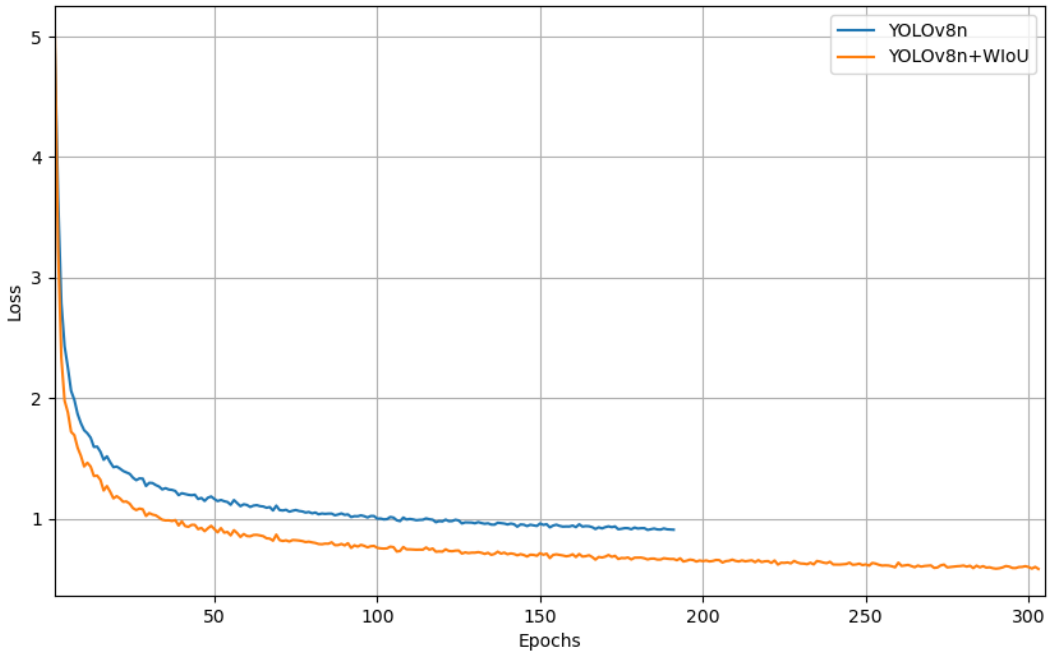


Figure 19. Loss curves during training before (YOLOv8n) and after loss function enhancement (YOLOv8n + WIoU)

doi:10.6342/NTU202401594

The difference in the total number of epochs which is apparent in the figure can be depicted as the result of the early stopping strategy utilized during training. In the context of the first model which is designed for detecting three different maturity classes, the introduction of WIoU into the YOLOv8n model did not emerge as the optimal configuration. However, it can be observed that WIoU substantially reduced the loss during the training, as illustrated in Figure 17, while simultaneously enhancing the precision of the model, as evidenced in Table 12.

These findings may be attributed to the loss function's distinct ability to minimize the negative gradients caused by poor-quality data while focusing the model's attention on training samples of average quality by modifying the approach for gradient gain allocation. WIoU loss function assigns different weights to different parts of the predicted bounding boxes, often improving the model's ability to localize objects precisely. This strategic optimization enables the model to learn the significant features of the object of interest with greater precision and consistency, facilitating enhanced learning outcomes (Wang, C. et al., 2023). Additionally, it speeds up the fitting procedure, suggesting that the model became more capable of accurately capturing the properties of the object of interest.

Apart from those aspect, the detection performance of the model for individual classes is clearly influenced by the number of samples allocated for training each class. As shown in Table 1 of Section 3.6.7.1, the unripe class has highest number of samples (8,918 samples), followed by fully-ripened (5,084 samples), and semi-ripe classes (3,731 samples). The highest prediction accuracy obtained by the unripe class may highly be attributed to that factor. Similar to the findings of Egi et al. (2022), the results of this experiment validated that the quantity of samples influence the performance outcome of the network model. However, in contrast to their results, the unripe class achieved the

doi:10.6342/NTU202401594

highest performance in this experiment. These outcomes are similar with Li, P. et al. (2023) where the best recognition performance was recorded for the immature or unripe, followed by the ripe category. Despite the complex greenhouse environment and the green color of the plants' leaves and branches which became the cause for Egi et al. (2022)'s lower performance for the green class, this analysis further proves the influence of the total quantity of samples into the model's performance.

That was further verified through the performance for the fully-ripened and semi-ripe classes. In this experimental procedure, the semi-ripe class has the lowest quantity of samples which led to its lowest performance among the three classes. Apart from having higher number of samples, the superiority of ripe class over the semi-ripe may be attributed to the unique color characteristic and feature of red cherry tomatoes makes them more detectable compared to semi-ripe cherry tomatoes (Yuan et al., 2020; Egi et al., 2022). In addition, the model was trained to detect and recognize the semi-ripe class which encompasses cherry tomatoes at multiple maturity stages (see Section 3.6.2) which may also affected its overall performance. This may have caused an increased intra-class variation which made it harder for the model to learn distinctive features and accurately classify those that were assigned in the semi-ripe class.

Lastly, the confusion matrix for the highest performing model (YOLOv8n + CA) was generated as shown in Figure 20. The provided confusion matrix represents the performance of a classification model across three major classes: unripe, semi-ripe, and ripe, and a background class. The matrix shows that the model is highly accurate in predicting the unripe class, with 1729 true positives and minimal misclassifications into other categories. The semi-ripe class has 591 true positives but also shows significant misclassification, particularly into the background class. For the ripe class, the model performs well with 930 true positives, though a notable number of ripe instances are

doi:10.6342/NTU202401594

misclassified as background. The background class is the most misclassified, indicating the model struggles to distinguish background from other categories, particularly unripe. Overall, the model shows high accuracy in the unripe and ripe classes. This confusion matrix provides a clear indication of where the classification model excels and where it requires improvement, particularly in reducing the misclassification of background instances.

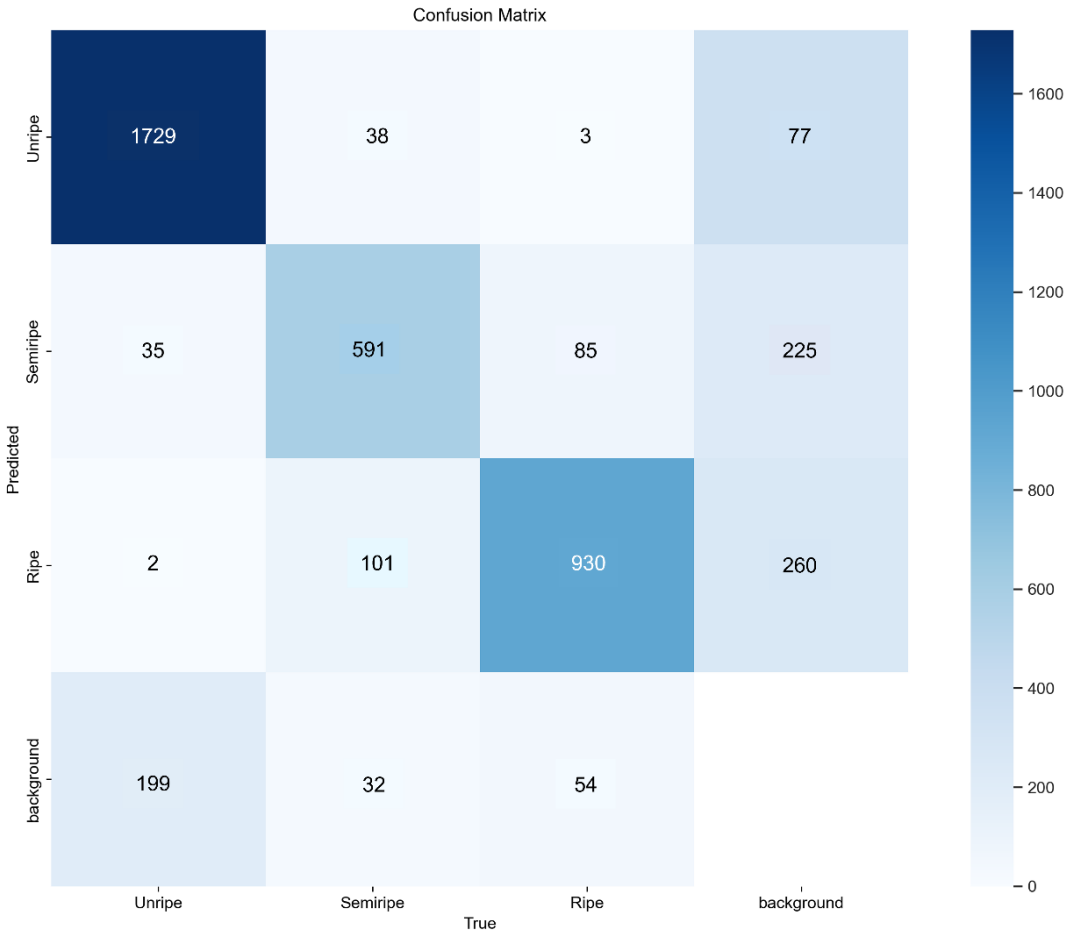


Figure 20. Confusion Matrix for the Best Multiclass Cherry Tomato Detection Model

doi:10.6342/NTU202401594

4.4 Detection Models Focusing on Single Classes

To further verify the effectiveness of the configurations discussed in Section 4.3, models for detecting single classes were trained and developed. The second model is designed for detecting cherry tomatoes at light-red maturity stage while the third model is focused on the red (fully ripe) maturity stage. The performance curves of these models are comprehensively elaborated in Appendix E. In this experiment, YOLOv8n with WIoU was not included because the primary goal of testing this loss function on the first model was to determine if it could minimize loss and improve detection accuracy. The results confirmed this hypothesis, demonstrating that WIoU effectively enhanced performance. Hence, for this investigation, the combination of Coordinate Attention (CA) and WIoU was tested to evaluate whether this combination could further enhance single-class detection performance.

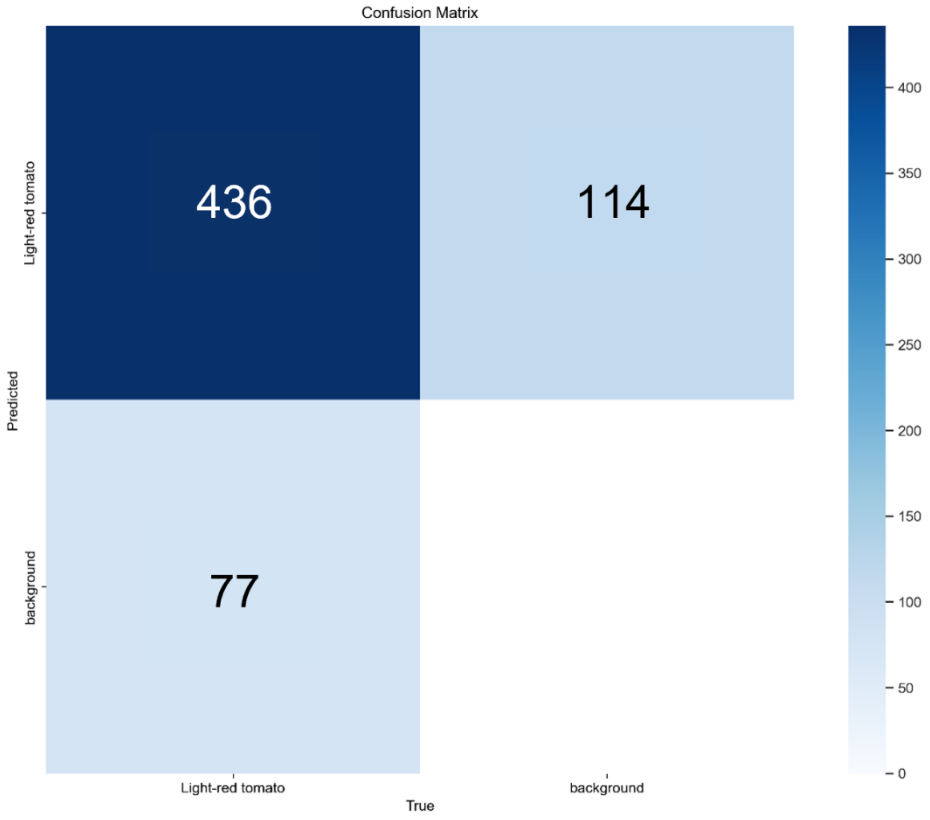


Figure 21. Confusion Matrix for the Best Light-Red Cherry Tomato Detection Model

doi:10.6342/NTU202401594

In the context of detecting light-red cherry tomatoes, a consistent trend with decreasing precision, increasing recall, and improving F1-score and mAP were observed (Table 13). Based on the findings, YOLOV8n + CA + WIoU achieved the best overall performance among the trained models. To further validate its outstanding performance based on the performance metrics elucidated in Table 13, the confusion matrix was generated (Figure 21).

Out of the total instances, the model successfully identified 436 'Light-red tomato' cases (TP), incorrectly classified 114 'background' instances (FP) as 'Light-red tomato,' and overlooked 77 'Light-red tomato' instances (FN). In addition, fluctuations were observed in the mAP curve during the training stage (Appendix F1). Despite these fluctuations, it can still be observed that the model was able to maintain consistently escalating trajectory, indicating that the model's overall performance was not compromised.

Expanding the evaluation scope to encompass red tomato detection, the effectiveness of the YOLOv8n enhancements becomes more evident (Table 14). For this model, YOLOv8n + CA achieved the highest precision. However, YOLOv8n + CA + WIoU demonstrated the best performance by achieving the highest recall, F1-score, and mAP. In order to more intuitively evaluate the actual effect of the trained cherry tomato detection model (YOLOv8n + CA + WIoU) in recognizing ripe instances, this experiment has generated the confusion matrix of the model (Figure 22). As depicted in the figure, the detection of ripe cherry tomatoes encountered recognition errors, occurring in certain situations. In the model's performance evaluation, it correctly identified 'Red tomato' in 1115 instances (True Positives), misclassified 'Red tomato' as 'background' in 119 instances (False Negatives), and incorrectly identified 'background' as 'Red tomato' in 167 instances (False Positives). This performance proves that the trained detection model is

doi:10.6342/NTU202401594

generally accurate in identifying and recognizing ripe cherry tomatoes. Nevertheless, misclassification is still subject to a tiny margin of error because of several variables, including the complexity, quality, and resolution of the images.

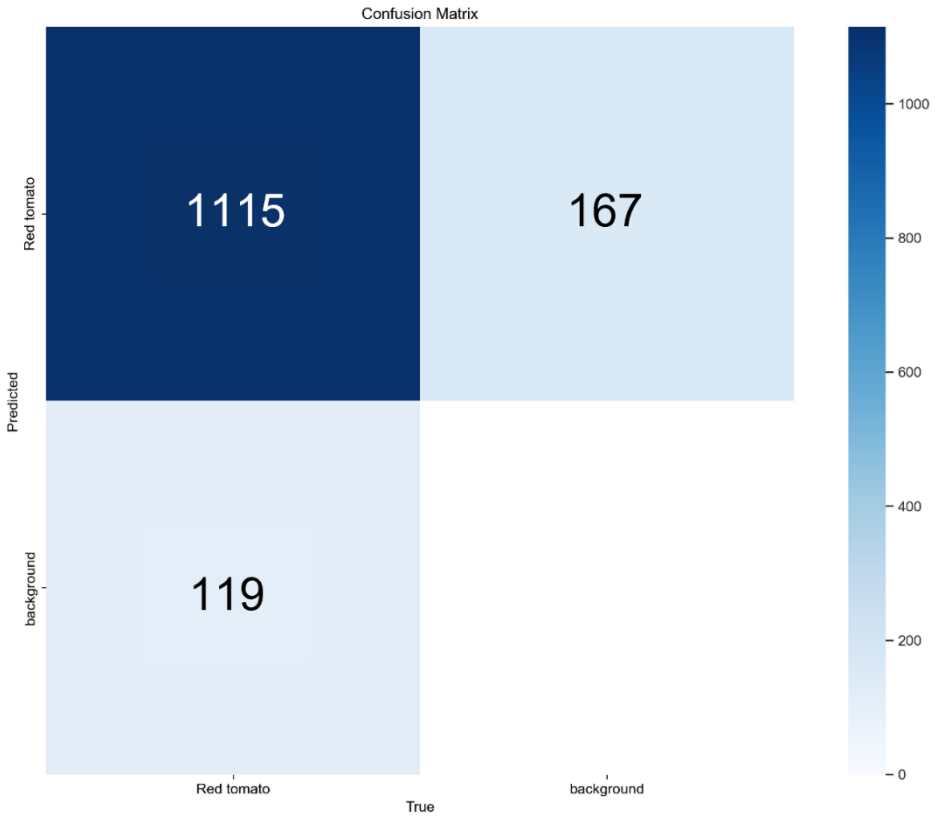


Figure 22. Confusion Matrix for the Best Red Cherry Tomato Detection Model

In addition to the observed and evaluated performance metrics, overfitting of the model is considered as the greatest concern during the training process (Li, P. et al., 2023). Hence, relevant curves that can be obtained during the training and validation processes were generated and displayed in Appendix F2. The YOLOv8n + CA + WIoU model achieved optimal precision, recall, and mAP within the first 50 epochs of training. Both training and validation losses followed consistent downward trajectory, indicative of

doi:10.6342/NTU202401594

stable convergence. Significantly, convergence occurred in the early stage, prior to completing even one-third of the total iterations.

For the single class detection models, it is clearly evident that YOLOv8n + CA + WIoU exhibited the best performance between the two configurations. However, the mAP of the model for light-red (Table 13) and red (Table 14) features a relatively noticeable discrepancy of 5.4%. These outcomes may be attributed to the insufficient quantity of samples for light-red cherry tomatoes, resulting to poor preliminary performance. Compared to the semi-ripe class samples for the overall surveillance and monitoring model, the model for light-red cherry tomato is more specific and exclusively encompasses the fourth stage of tomato maturity (El-Bendary et al., 2015) (see Section 3.6.2), leading to a slightly different outcome from the overall surveillance and monitoring models. Apart from having insufficient number of samples for the light-red maturity stage, the low performance results displayed by the trained models during the initial phases of the experiment could also be attributed to the influence of varying illumination which causes a certain degree of feature similarity. Similar to the findings of Li, P. et al. (2023), illumination variation may have caused some feature similarity between semi-mature and mature tomatoes. These outcomes have proven the challenging nature of training a deep learning model for detecting cherry tomatoes during the light-red maturity stage.

After integrating data augmentation strategies and successfully increasing the quantity of the samples, the performance of the trained detection models was significantly improved. Numerous authors have utilized data augmentation strategies and elaborated that such techniques increase data variability (Egi et al., 2022) and data volume (Ge et al., 2022), and gives robustness to the detection model (Camacho & Morocho-Cayamcela, 2023) by introducing prior knowledge into the dataset used for model training (Ge et al.,

doi:10.6342/NTU202401594

2022). These preprocessing strategies improve the model's training (Wang, C. et al., 2023) and detection accuracy (Liu, G. et al., 2020; Appe et al., 2023) which thereby highlights and proves the effectiveness of adapting data augmentation techniques for improving the performance of the deep learning object detector.

Interestingly, the observed fluctuations in the mAP curve (Appendix F1) during training epochs reveal a dynamic learning process influenced by several factors. While the overall trend shows an increase in performance, characterized by improvements in AP values over time, the presence of local maxima and minima suggests that the learning trajectory is not entirely smooth. These fluctuations indicate that the model encounters varying challenges and learning rates throughout training. The model's performance might exhibit rapid improvements at certain epochs (local maxima) while encountering slight drops at others (local minima). These fluctuations can be caused by various factors such as the stochastic nature of optimization algorithms, variations in the training data, or the model encountering challenging samples.

Compared to the detection model for light-red cherry tomatoes, relatively higher overall performance was achieved for the third model which is designed for recognizing red cherry tomatoes. The results of this experiment further strengthen the conclusion of several authors (Yuan et al., 2020; Egi et al., 2022; Wang, C. et al., 2023) stating that the distinct color feature of red tomatoes is a significant factor which contributes to the high accuracy of detecting this maturity stage. The highly distinct color differences between the ripe cherry tomatoes and the background (which includes leaves, stems, and other objects inside the greenhouse environment caught in the scene) enables the deep learning model to accurately and precisely distinguish and segment the vibrant red color of the fruit from the mostly green environment. On the other hand, the aforementioned missed detections (FN) may be attributed to some extent of feature similarity between ripe cherry

doi:10.6342/NTU202401594

tomatoes and its preceding maturity stage brought by illumination factors (Li, P. et al., 2023).

Furthermore, the observed performance of YOLOv8n + CA + WIoU illustrated in Appendix F2, showcases a consistent and smooth trajectory indicative of a stable and effective training process. This sustained improvement over time suggests a robust learning mechanism with minimized performance fluctuations, and highlights the model's ability to maintain a stable learning process. In addition, the model exhibits good convergence, signifying its effectiveness in learning intricate pattern in the data and optimizing its parameters towards achieving the desired detection performance. These relevant findings not only validate the efficacy of the improved network model but also highlight its ability to learn and adapt efficiently, which is crucial for its practical applicability in real-world scenarios.

With the goal of further enhancing the robust framework, the fine-tuning and ablation study conducted on YOLOv8 have demonstrated notable improvements in detection accuracy, resulting in an approximate 0.8% ~ 4.7% enhancement. Such advancements are particularly pertinent in precision agriculture, where accurate identification of objects like ripe cherry tomatoes directly impacts yield estimation and operational efficiency. However, the integration of these modifications introduces significant practical considerations. Implementation often necessitates increased computational resources and energy consumption, which can pose logistical and financial challenges, particularly in resource-limited agricultural settings. Moreover, the elevated complexity associated with deployment and maintenance demands proficient technical expertise and support, potentially limiting accessibility and scalability.

doi:10.6342/NTU202401594

4.5 Detection and Tracking

Table 15 (BoT-SORT) and Table 16 (ByteTrack) presents the calculated performance of the tracking algorithms on the selected video subsequences. Ripe cherry tomatoes were detected using a set of parameters consisting of: (a) Higher Tracking Threshold: 0.5; (b) Lower Tracking Threshold: 0.1; (c) Matching Threshold: 0.8; (d) New Track Threshold: 0.6; and (e) Tracking Buffer: 30. The results evidently demonstrate that, BoT-SORT consistently outperforms ByteTrack in all aspect. BoT-SORT showcased an average MOTA of 65% compared to ByteTrack's 63%, representing 2% margin in accuracy. Moreover, BoT-SORT maintains remarkable lead in IDF1 metrics, surpassing ByteTrack by an average of 14.3%. In the context of tracking ripe cherry tomatoes, BoT-SORT demonstrates better performance as evidenced by the prevalence of ID switching instances. Furthermore, both algorithms exhibit identical behavior in terms of false positive and false negative occurrences with false positives prevailing and false negatives remaining relatively low.

Table 15. Performance Evaluation of the Bot-SORT Algorithm Across Different Scenarios

	Ground Truth	False Positive	False Negative	ID Switch	MOTA	IDF1
Video sequence 1	1562	390	45	14	0.71	0.87
Video sequence 2	1192	369	191	4	0.53	0.79
Video sequence 3	443	75	46	2	0.72	0.87

Table 16. Performance Evaluation of the ByteTrack Algorithm Across Different Scenarios

	Ground Truth	False Positive	False Negative	ID Switch	MOTA	IDF1
Video sequence 1	1562	370	70	21	0.70	0.73
Video sequence 2	1192	356	194	9	0.53	0.77
Video sequence 3	443	82	62	13	0.65	0.61

doi:10.6342/NTU202401594

To boost the tracking performance of BoT-SORT algorithm, the tracking parameters were empirically fine-tuned. For tracking the ripe cherry tomatoes using the enhanced BoT-SORT algorithm, the main parameters consist of: (a) Higher Tracking Threshold: 0.8; (b) Lower Tracking Threshold: 0.5; (c) Matching Threshold: 0.9; (d) New Track Threshold: 0.5; and (e) Tracking Buffer: 40. Table 17 shows the performance of the optimized detection and tracking algorithm on the input video sequences which represents three different scenarios. Throughout the whole sequence spanning of 150 frames (6 seconds) each, it can be observed that relatively few ID switches occurred in all tests. Overall, the performance metrics across the three video sequences reveal subtle distinctions in tracking accuracy.

Table 17. Performance Evaluation of the Fine-tuned Algorithm Across Different Scenarios

	Ground Truth	False Positive	False Negative	ID Switch	MOTA	IDF1
Video Sequence 1	1562	22	182	3	0.87	0.90
Video Sequence 2	1192	13	301	1	0.74	0.86
Video Sequence 3	443	11	93	1	0.76	0.88

Video sequence 1, characterized by the highest ground truth instances, resulted in MOTA of 0.87 and IDF1 of 0.90. Despite detecting fewer instances, video sequence 2 presented a challenge with significant rise in FN impacting both MOTA (0.74) and IDF1 (0.86). Conversely, video sequence 3, with the least instances, demonstrated a relatively lower MOTA of 0.76 despite moderate FP count, suggesting potential issues in detection precision. However, its IDF1 score of 0.88 suggests a relatively stronger performance in preserving object identities despite the lower overall detection rate. Figure 23 presents some examples of the image frames where ID switching occurred. It can clearly be

doi:10.6342/NTU202401594

depicted in the figure that the ID switching cases happen when the cherry tomatoes are occluded.

Table 18. Detection and Tracking Performance for Counting Ripe Cherry Tomatoes

Sequence	Ground Truth	Estimated	CER (%)	CA (%)
Video Sequence 1	19	26	36.84	63.16
Video Sequence 2	20	23	15.00	85.00
Video Sequence 3	5	7	40.00	60.00

Apart from that analysis, performance of the detection and tracking system for counting ripe cherry tomatoes are elucidated in Table 18. The results given in the table reveal variability in the system's accuracy, with the best performance in Video Sequence 2 and the most significant error in Video Sequence 3. This variability suggests that while the system is effective under specific conditions, it encounters substantial challenges in scenarios characterized by a lower number of objects, which leads to higher error rates and diminished accuracy. The key point is that CER reflects the relative difference between estimated and actual values. In instances where ground truth data is scant, minor absolute differences can result in an observably high CER and CA. Given these findings, advancements in detection algorithms and targeted strategies to mitigate these particular challenges could potentially enhance overall system performance.

The present investigation conducted ripe cherry tomato detection, tracking, and counting with the aim of establishing a mechanism which will allow autonomous monitoring systems to accurately detect or locate the ripe cherry tomatoes within the complex environment of greenhouse setting. This may provide tracking capability which may enhance the efficiency and effectiveness of the UAV-based cherry tomato field screening and validation system. In addition, this investigation may also contribute some insights that may become significant for an autonomous harvesting and picking system by ensuring that robot arms can navigate, reach, and pick fruits of interest with high

doi:10.6342/NTU202401594

accuracy, even in scenarios involving occlusion, overlapping, or varying illumination conditions.

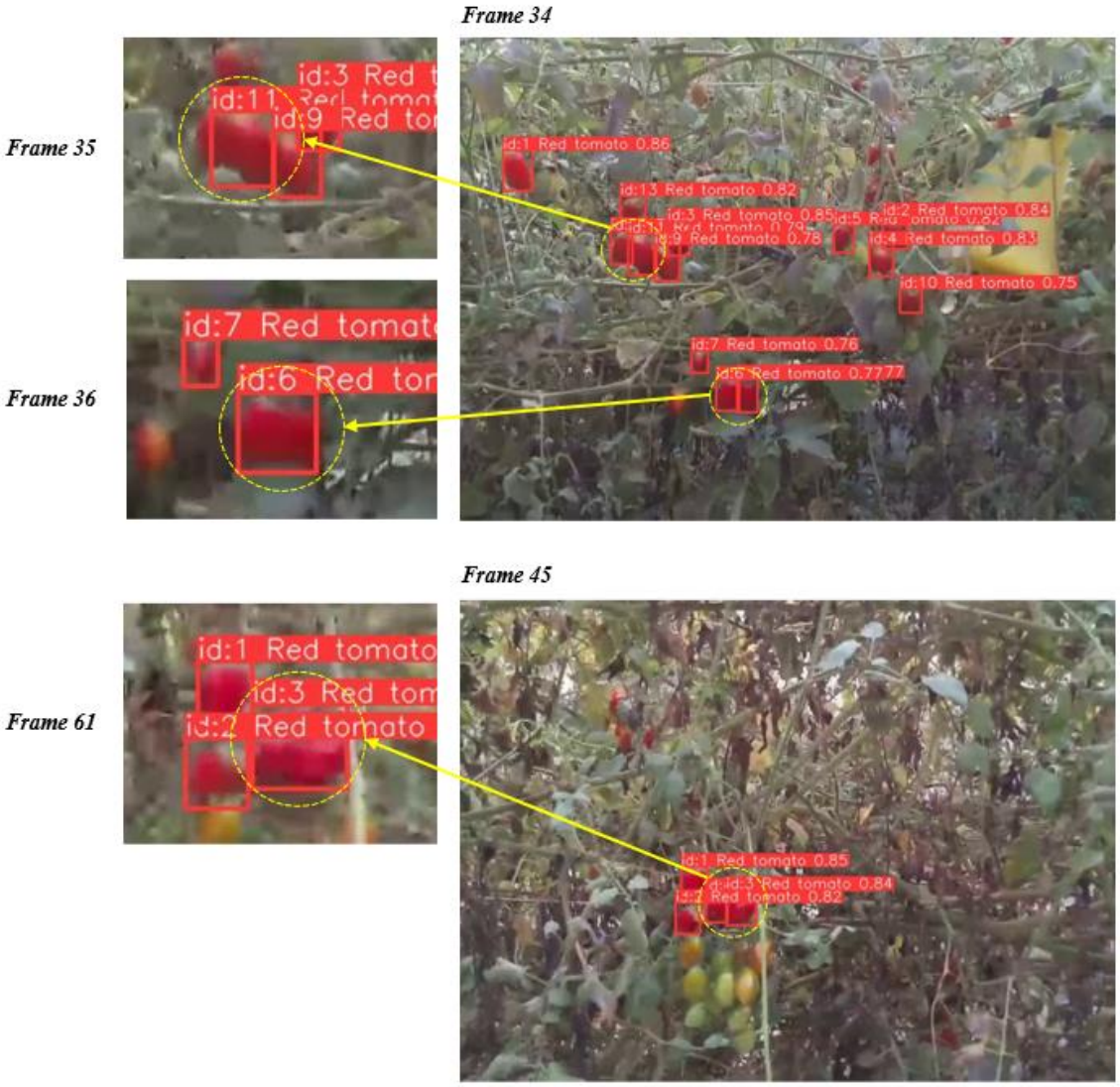


Figure 23. Sample of image frames illustrating the occurrence of ID switch. Frames 34 and 45 illustrate the image frames with correct detection (encircled objects) while Frames 35, 36, and 61 show the image frames with the occurrence of ID switch and tomatoes recognized as a single entity.

Through this experiment, it was found out that the adjustment of the confidence threshold plays a crucial role. Based on the results in Table 17, BoT-SORT demonstrated its appropriateness and suitability for tracking ripe cherry tomatoes. This is evident from

doi:10.6342/NTU202401594

both the tracking accuracy and the stability of the ID during the tracking process. The fine-tuned tracking parameters allowed for a fine trade-off between the risk of incorrect identifications (FP) and the risk of missing actual positive instances (FN), thereby providing a mechanism to address the algorithms' inclination towards over-predicting positive outcomes.

Furthermore, it was also found out that when two distinct objects are occluded or closely interacting in a way that the tracking algorithm perceives them as a single entity, merging of identities may be encountered. As observed in Figure 23, some cherry tomatoes were in close proximity and overlapping to the extent that the tracker associated them as a single object, resulting in the consolidation of their IDs. These findings, along with numerous authors, emphasize the influence of occlusion in restricting the performance of classifiers (Kurtulmuş et al., 2013; Liu, G. et al., 2019) and the importance of reducing its impact on fruit detection and precision picking (Luo, L. et al., 2016; Liu, G. et al., 2020). This factor significantly impacts the ability of the detection and tracking system to accurately track and identify ripe cherry tomatoes which possess strong feature similarity.

4.6 Impact of Distance and Confidence on Detection Performance

To further validate the findings of previous authors (Bhusnoor et al., 2023; Giap et al., 2023; Hendrawan & Istiono, 2023; Yu, H. et al., 2023) who emphasized that farther distances decrease the level of accuracy, the effect of different distances on the confidence scores of the detections were investigated. The results of this investigation have shown and validated that distance has a significant effect on the confidence of detections.

doi:10.6342/NTU202401594

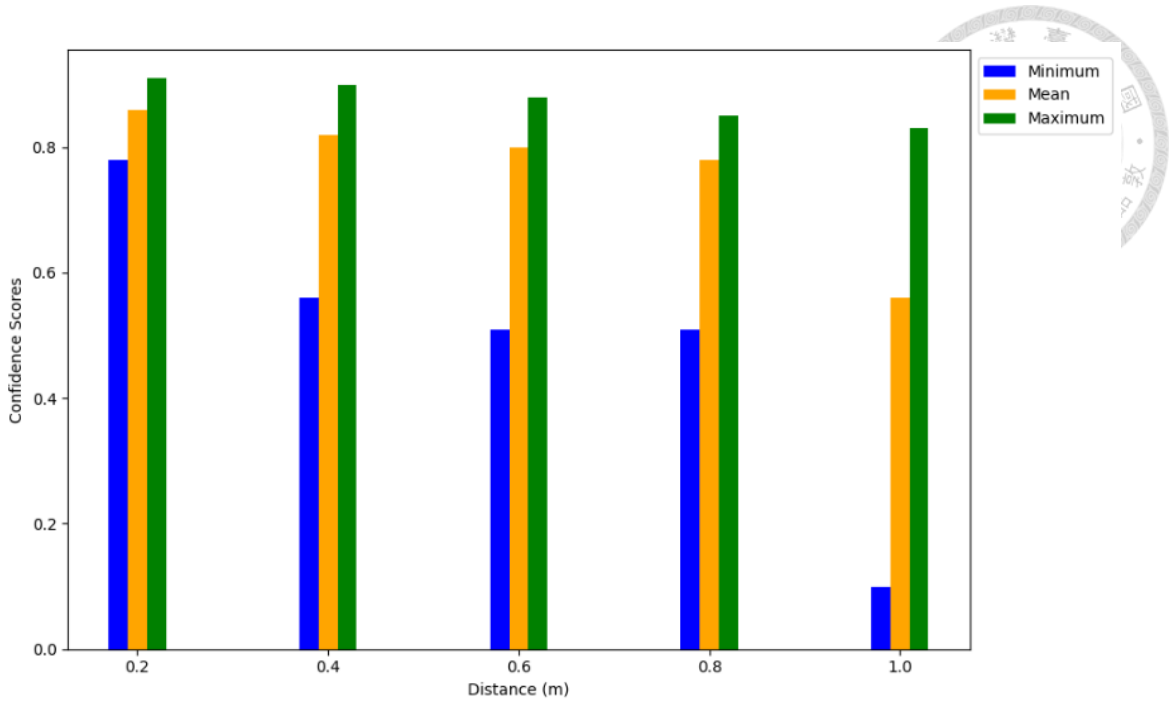


Figure 24. Minimum (blue), Maximum (green), and Mean (yellow) Confidence Scores Across Distances. The data presented were obtained from 5-second Video Sequences at 25 fps.

As shown in Figure 24, a decreasing trend of data was recorded for all values (minimum, maximum, and mean) as the distance between the target object and the camera increase. In terms of maximum confidence scores attained throughout the whole detection process, a gradual decrease in scores were recorded. Meanwhile, it can be observed that the marginal difference of the highest recorded confidence scores across different distances were less than 0.10%. On the other hand, an extreme drop in scores characterized by a 68% decrease were recorded as the distance between the target object and the camera reached 1.0 meter.

Through these experiments, the researcher successfully achieved the principal goal of validating the conclusion of the previous authors who also investigated the influence of distance on the accuracy of object detectors. Similar to the findings of Bhusnoor et al. (2023), a negative correlation between the factor of distance and

doi:10.6342/NTU202401594

confidence scores was also observed. The maximum values of confidence scores fall within 80-90% range despite the increasing object-to-camera distance. Hence, revealing the robustness of the YOLOv8n + CA + WIoU detection model to precisely recognize the object of interest even as the distance increases. However, as the distance increases, the detection algorithm was not able to maintain the high confidence scores. Similar behavior can be observed in other categories, particularly the minimum scores. Moreover, YOLOv8n + CA + WIoU detection algorithm shows decreased certainty in predictions across all categories as the distance increases. These outcomes thereby emphasize the significant role played by distance factors in developing a UAV-based detection system with remarkable reliability and efficiency.

Apart from analyzing the relationship between the confidence scores of the detected instances and varying distances, the presence of detections throughout the observation process were also observed (Figure 25 ~ 29). The figure illustrates the tracking of ripe cherry tomatoes over time, with each identity represented by a specific color: blue for Identity 1, green for Identity 2, and red for Identity 3. In some cases, Identity 4 which is represented by the cyan color appears due to the occurrence of ID switch. These identities were assigned to distinguish and monitor the individual tomatoes throughout the observation process, enabling detailed analysis of the presence of their detections across the frames. Specifically, Figure 25 illustrates that all objects were consistently detected across all frames within the five-second video (total of 125 frames) at 0.2 m object-to-camera distance. However, as the distance increased from 0.4 m (Figure 26) to 1.0 m (Figure 29), the algorithm began to miss detections. Moreover, at the same distance ranges, instances of ID switching were observed. Furthermore, at distances of 0.6 m (Figure 27) and 0.8 m (Figure 28), no instances of ID switching occurred; nevertheless, missed detections, absent from any frames, were recorded.

doi:10.6342/NTU202401594

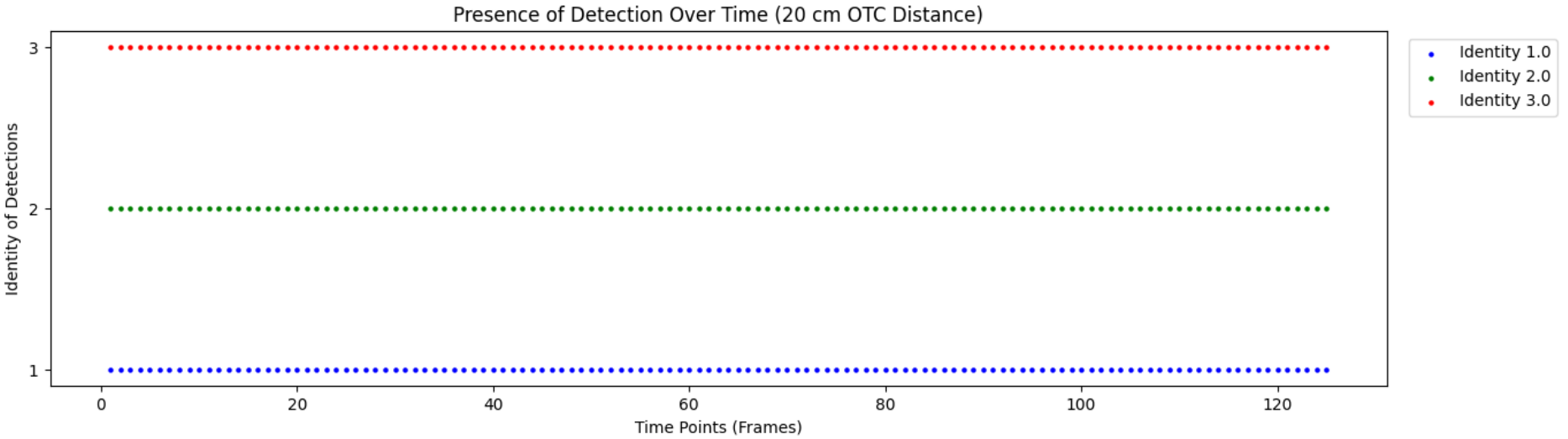


Figure 25. Tracking the presence of detection instances over time at 0.2 m object-to-camera (OTC) distances

doi:10.6342/NTU202401594

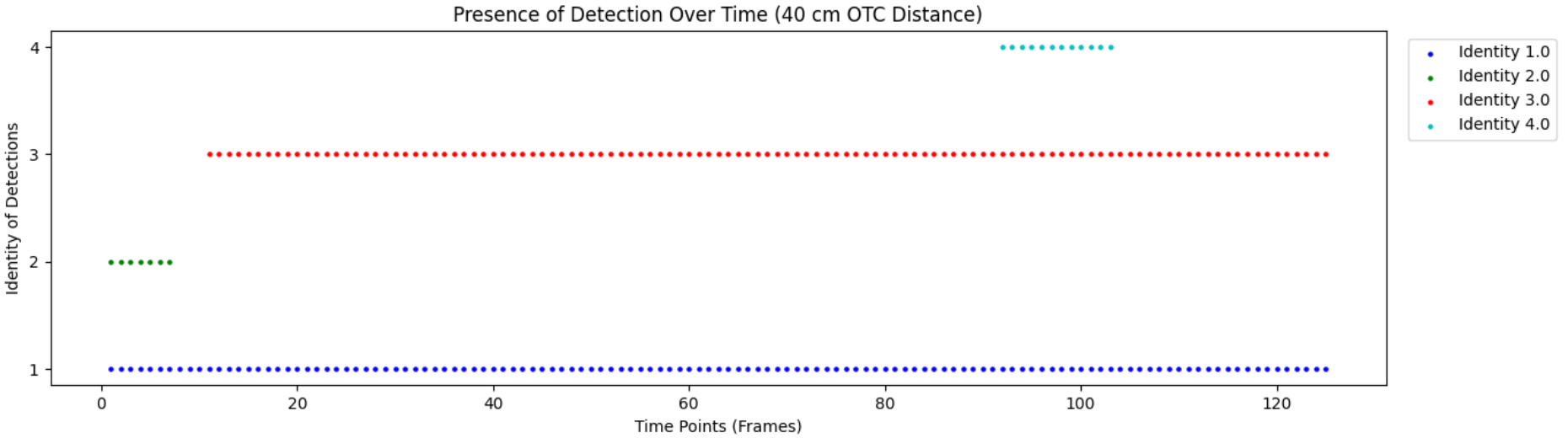


Figure 26. Tracking the presence of detection instances over time at 0.4 m object-to-camera (OTC) distances

doi:10.6342/NTU202401594

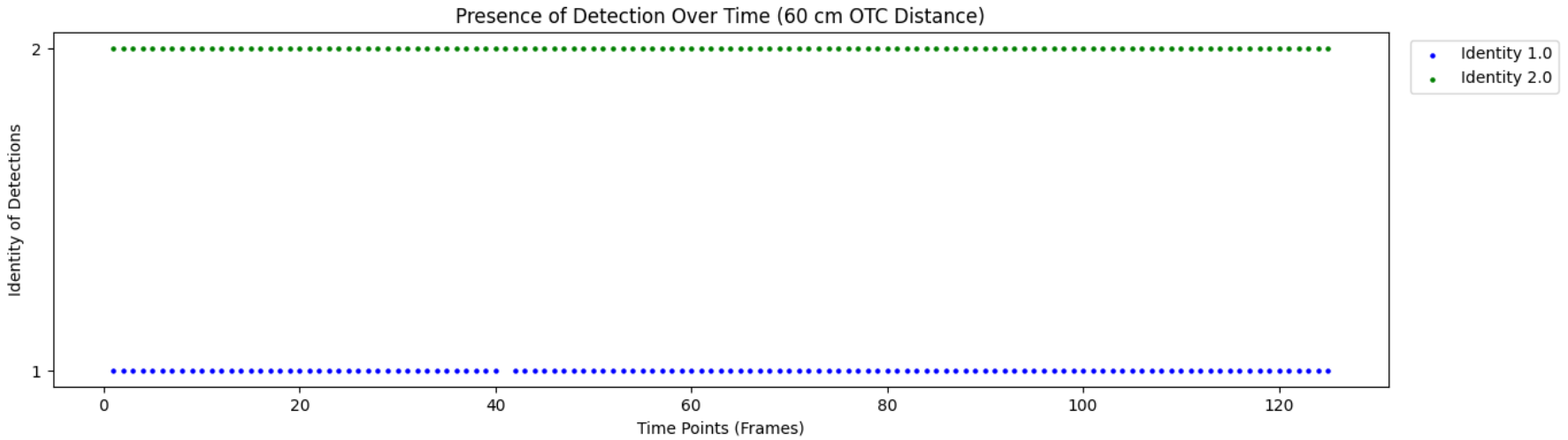


Figure 27. Tracking the presence of detection instances over time at 0.6 m object-to-camera (OTC) distances

doi:10.6342/NTU202401594

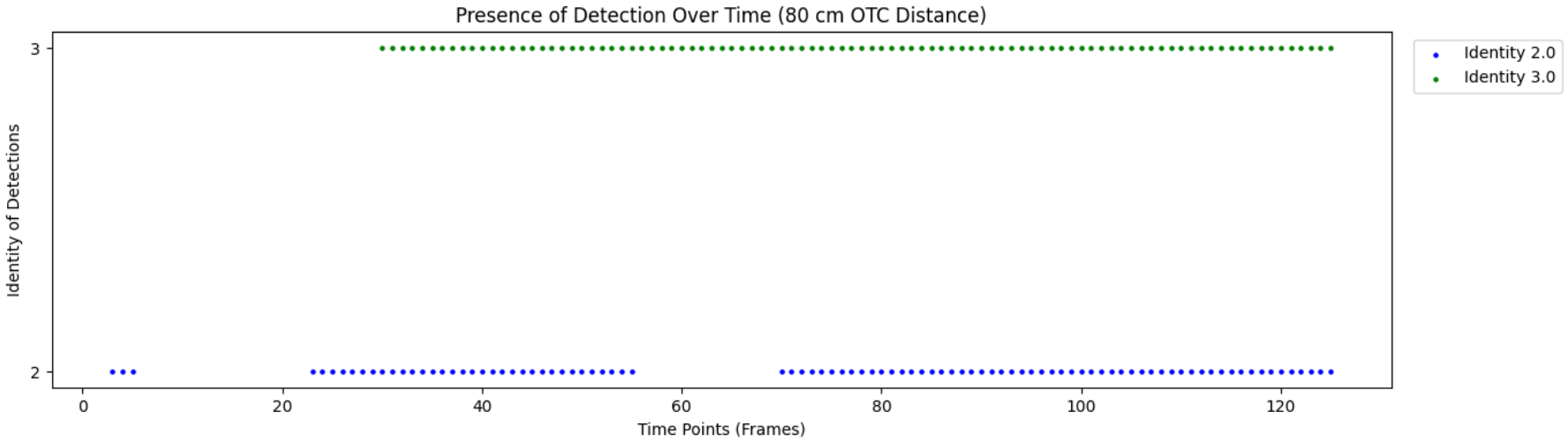


Figure 28. Tracking the presence of detection instances over time at 0.8 m object-to-camera (OTC) distances

doi:10.6342/NTU202401594

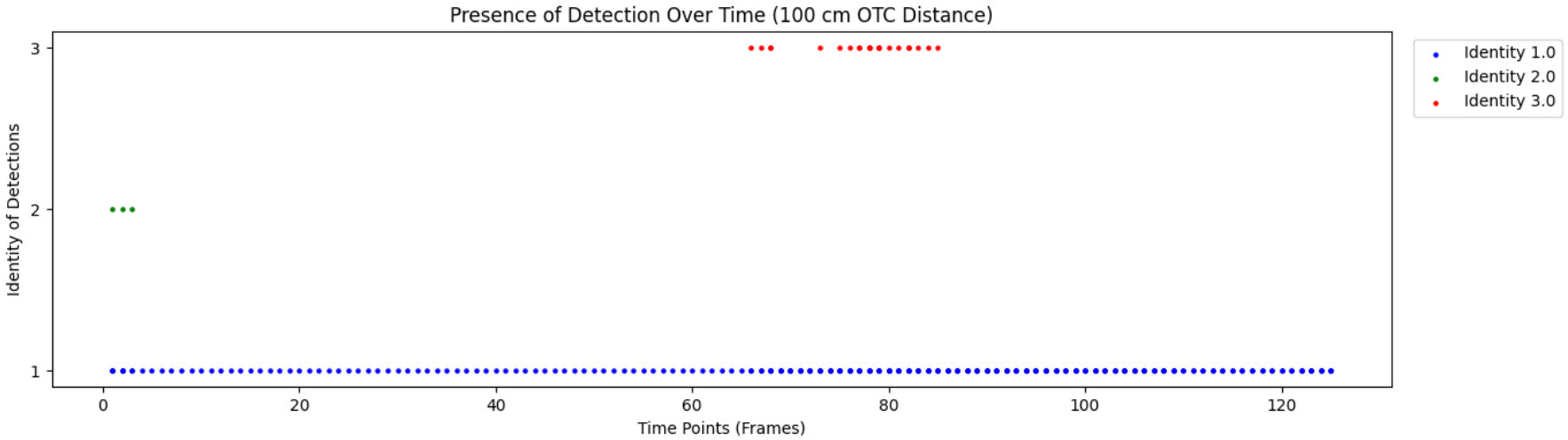


Figure 29. Tracking the presence of detection instances over time at 1.0 m object-to-camera (OTC) distances

doi:10.6342/NTU202401594

Although the findings further validated that the distance between the camera and the tomato in the scene pose a correlation to the detection performance, the results also show that the occurrence of ID switching and missed detections were attributed to the presence of occlusions in the scene (Figure 30). Cherry tomatoes often form clusters, which can obscure ripe instances from the view and perspective of a UAV camera. This occlusion presents a challenge for the detection model (Yuan et al., 2020), potentially hindering its ability to recognize these tomatoes as positive instances due to the obstruction of their features. The findings of this experiment consistently support the conclusions of the previous authors mentioned in this section, thereby strengthening the experiment's purpose and affirming the credibility of its interpretations. Objects which appear small in images due to their small size nature and the object-to-camera distance may lack valuable and distinguishable information such as color and texture, thereby making the detection challenging (Rekavandi et al., 2022; Yu, H. et al., 2023). Moreover, Mirzaei et al. (2023) stated that objects with relatively small size such as tomatoes (Tsai et al., 2023) increases the likelihood of being misinterpreted as image noise, thereby compromising the precision of detection and tracking algorithms. These factors pose greater challenge on cherry tomato detection considering their relatively small size. By approaching the cherry tomatoes at a closer distance using the UAV’s camera, a larger area of the frame is occupied resulting in increased visibility and thereby facilitating more accurate detection. This phenomenon, as Tsai et al. (2023) have demonstrated, highlights the variation in detection accuracy depending on both the size of the target in the image and the level of occlusion. However, while closer distances enhance object detection performance, they limit the field of view, capturing fewer cherry tomatoes per frame and increasing the time and energy required for scanning larger areas. Additionally, closer distances may necessitate higher computational complexity due to the need for greater

doi:10.6342/NTU202401594

resolution and precision. Considering these factors, a distance range of 0.4 m to 0.6 m may be recommended for practical applications. This range balances high confidence scores (maximum ~0.9, mean ~0.7) with improved area coverage, optimizing the trade-off between distance, performance, and operational efficiency. This approach ensures effective detection while minimizing the constraints of time, energy, and computational resources.

doi:10.6342/NTU202401594

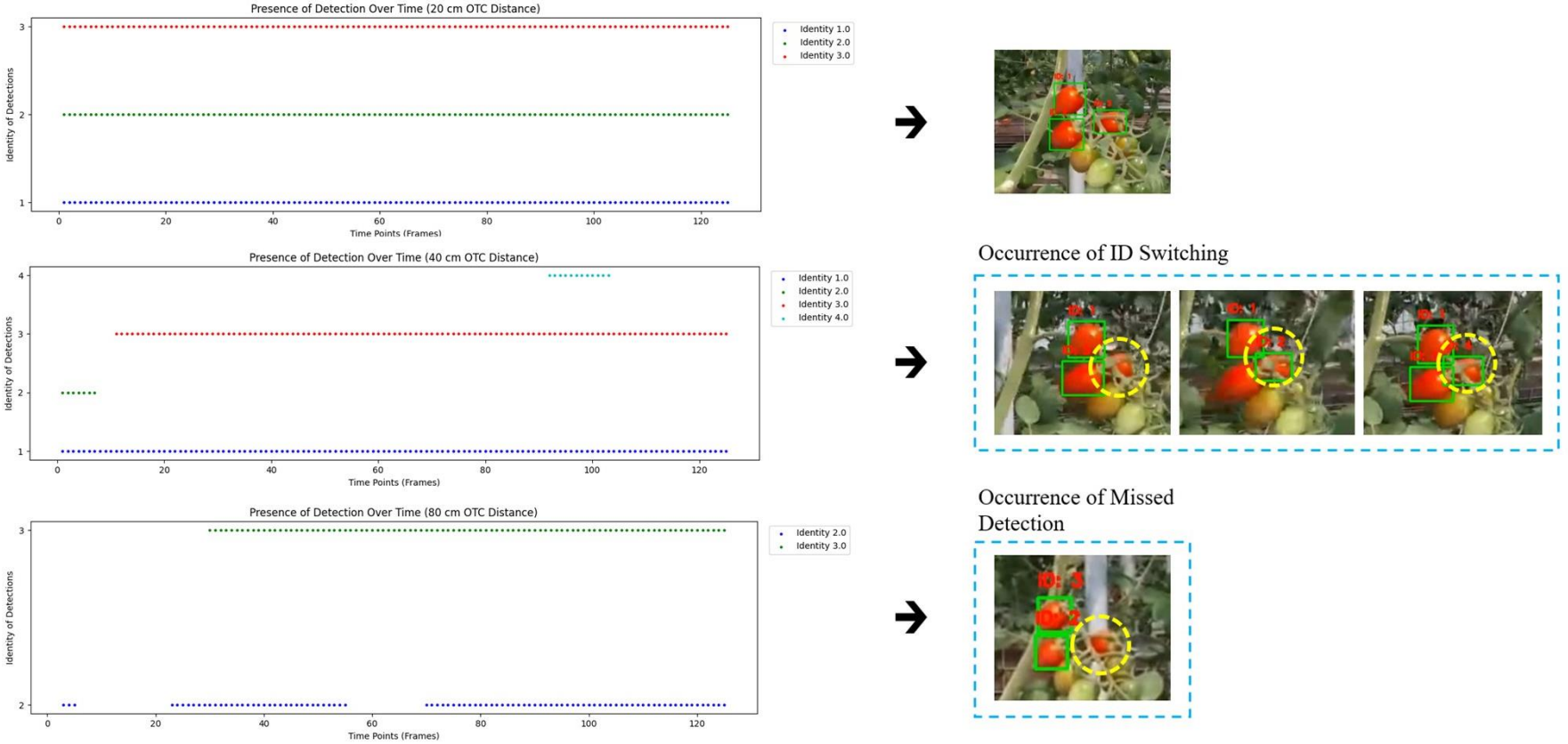
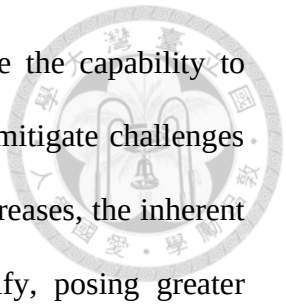


Figure 30. Occurrence of ID Switching Cases and Missed Ripe Cherry Tomato Detections due to Presence of Occlusions

doi:10.6342/NTU202401594



These findings signify that a closer distance not only have the capability to enhance the clarity and distinction of the cherry tomatoes but also mitigate challenges associated with background noises. As object-to-camera distance increases, the inherent complexities of the scene in a cherry tomato greenhouse intensify, posing greater obstacles for more robust and accurate cherry tomato detection. The subtle appearance and limited distinguishing features of small objects result in a scarcity of crucial information, complicating the tracking process and often leading to diminished efficiency and accuracy (Mirzaei et al., 2023).

In addition to the batch of analyses to determine the factors influencing the overall performance of object detectors, the impact of confidence threshold on the tracking performance of the YOLOv8n + CA + WIoU was also investigated (Figure 31). Results of the experiment consistently demonstrated that with a higher and more stringent confidence threshold, the occurrence of ID switches were minimized. Based on the figure, ID switching instances can be minimized to as few as 1 to 4 occurrences when the confidence threshold is set to 80% and above.

Meanwhile, increasing the confidence threshold proved to be an effective strategy for enhancing both Multi-Object Tracking Accuracy (MOTA) and IDF1 metrics. The highest MOTA values, ranging from 70% to 87%, were recorded when the confidence threshold was set to 80%. Similarly, the IDF1 scores for all input video sequences were also highest at the 80% confidence threshold, with values ranging from 86% to over 90%.

By carefully and comprehensively fine-tuning this parameter as several authors also conducted (Wojke et al., 2017; Zhang, Y. et al., 2022; Aharon et al., 2022; Wang, Z. et al., 2023), the overall capability and performance of the utilized tracking algorithm were improved. Thus, ensuring more reliable and efficient cherry tomato detection and tracking where effective decision-making and real-time response is crucial and

doi:10.6342/NTU202401594

significant. This present experiment demonstrates the significant impact of adjusting the confidence threshold on the performance of multi-object tracking algorithms, specifically for cherry tomato-related applications. By setting the confidence threshold to 80%, remarkably high MOTA and IDF1 scores were obtained, showcasing the effectiveness of this approach in enhancing both tracking accuracy and identity preservation. Furthermore, Figure 32 depict the behavior of FN and FP at different confidence thresholds.

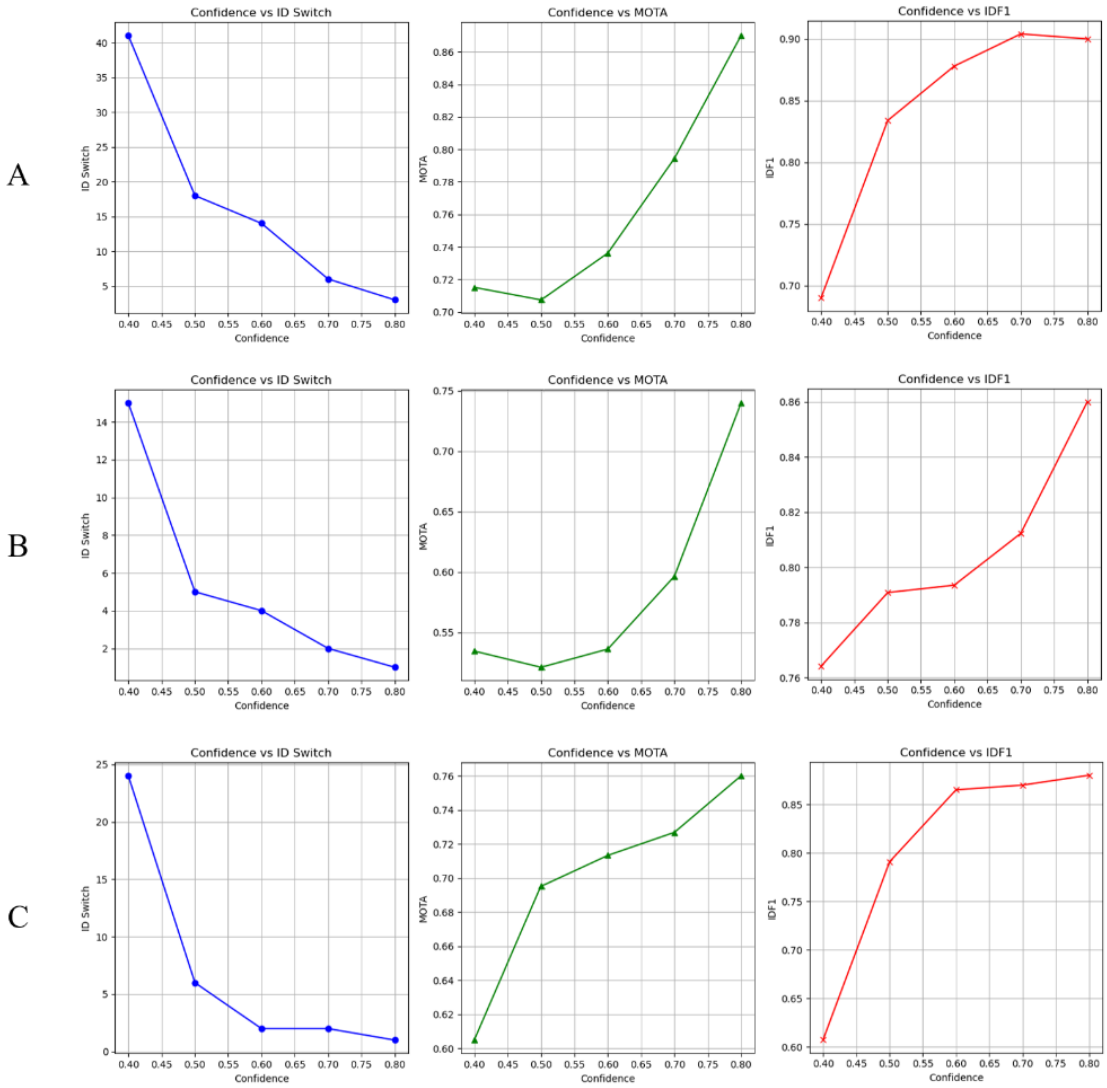
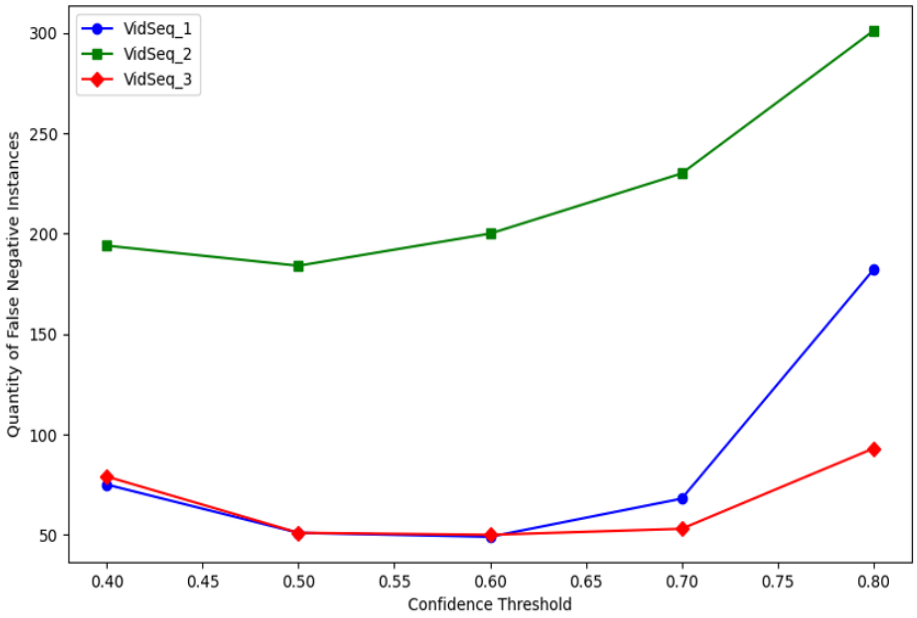
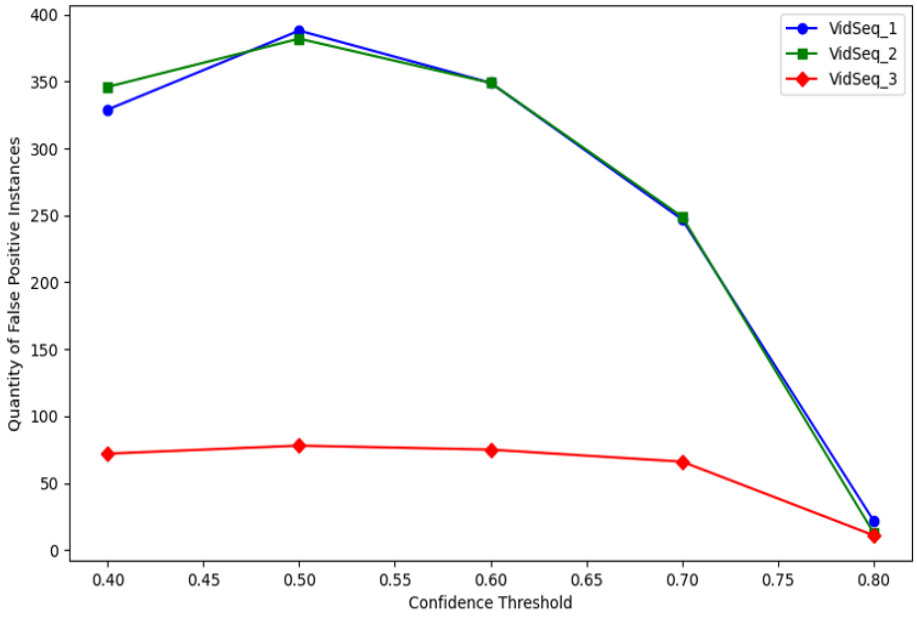


Figure 31. Impact of Confidence Threshold as evaluated using ID Switching Occurrence, MOTA, and IDF1. The Three Video Sequences (A, B, and C) used in evaluating the tracking performance of the Detection Model were also utilized for this set of experiment.

doi:10.6342/NTU202401594



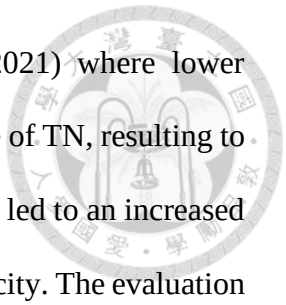
A



B

Figure 32. Behavior of Detections Throughout a Set of Confidence Thresholds: (A) False Negative and (B) False Positive

doi:10.6342/NTU202401594



These set of findings are identical to Monaghan et al. (2021) where lower confidence threshold resulted to high number of FP and enhanced rate of TN, resulting to a highly sensitive algorithm. Conversely, higher confidence threshold led to an increased number of both TP and FN, resulting to an algorithm with high specificity. The evaluation of optimal confidence thresholds for maximizing sensitivity and specificity reveals distinct ranges conducive to each goal. High sensitivity, which seeks to minimize FN, is best achieved with a confidence threshold ranging from 45% to 50%. This range effectively lowers the incidence of FN, particularly for VidSeq_1 and VidSeq_2, while VidSeq_3 consistently shows minimal FN. For high specificity, defined by the reduction of FP, a threshold between 75% and 80% is most effective. Within this range, FP are significantly minimized across all video sequences, with VidSeq_3 achieving the lowest rates. The lowest rates achieved by VidSeq_3 can be attributed to the low ground truth instances for the sequence since only a single cluster was observed at a fixed camera distance (see Table 15 ~ 17). Meanwhile, the observably high quantity of FN observed in VidSeq_2 may be attributed to the fixed camera distance while observing multiple clusters. This setting may have led to more frequent occlusions and overlaps among clusters, thereby increasing the quantity of FN. On the other hand, as the camera gradually gets closer to the cherry tomatoes (VidSeq_1), detection certainty increases while the number of samples present and viewed in the frames decrease, resulting to lower FN. By taking advantage of the observed stability and performance throughout the video sequences, these thresholds offer a well-balanced solution for applications demanding strict sensitivity and specificity.

Despite the success of previous works, the current results, and the method being the most common practice (Stanojevic & Todorović, 2024), a recent study still considered the technique of confidence threshold fine-tuning naïve given that tuning is effective

doi:10.6342/NTU202401594

when prior knowledge with the video stream data is available (Ma et al., 2023). This suggests that while confidence threshold fine-tuning remains a widely adopted approach, its efficacy is highly dependent on the availability of comprehensive prior data. Future work should therefore explore advanced methods that can perform effectively even in the absence of extensive prior knowledge.

doi:10.6342/NTU202401594

Chapter 5. Conclusion, Limitation, and Perspective

5.1 Conclusion

This study focused on the development of a digital monitoring approach for greenhouse cherry tomatoes through the utilization of UAV and the construction of an AI model based on YOLOv8. Additionally, the factors that affect the overall detection performance of object detectors were also investigated.

Experiments conducted in this study have validated that the utilization of UDP protocol for drone-computer communication system can provide efficient data transmission, allowing for smooth UAV control and prompt response. Despite the slight delay of 1.5 to 2 seconds in UAV-to-Computer data transfer during the drone's take-off, the system demonstrated successful data acquisition and intervention capabilities. Moreover, utilizing images from the established drone-computer communication system, alongside additional modalities such as mobile phones and the Intel RealSense D435 depth camera, were proven significant in training the deep learning model by showcasing the highest overall performance on the test sets. These findings reveal that integration of images from diverse modalities could be effective in enhancing the efficiency and generalization capability of YOLOv8n.

Moreover, fine-tuning and ablation studies conducted in this study emphasized that the precision, F1-score, and mAP metrics of the all the trained models for monitoring and surveillance shown remarkable performances, highlighting the effectiveness of introducing a set of CA blocks. Meanwhile, the development of detection models for light-red and red cherry tomato, through the introduction of both CA and WIoU, proves to be a robust tool for accurate and reliable single-class tomato detection. The combination of Coordinate Attention and WIoU worked better for single-class models

doi:10.6342/NTU202401594

potentially because the model can focus more narrowly on optimizing both feature extraction and localization without the added complexity of multiple classes. However, while the performance enhancements are compelling, the decision-making process regarding model selection must carefully weigh these benefits against the practical constraints inherent in agricultural applications. This entails a nuanced evaluation of not only performance metrics but also factors such as operational feasibility, sustainability, and resource efficiency. Ultimately, the suitability of adopting advanced models like fine-tuned YOLOv8 hinges on their ability to align with the operational realities and long-term objectives of agricultural stakeholders, ensuring that technological advancements translate into tangible benefits without compromising practical utility.

In addition to the advancements made in the cherry tomato detection system, this study highlighted the critical importance of tracking and counting individual cherry tomatoes to monitor their current maturity stage. By refining the confidence thresholds of the BoT-SORT algorithm, balanced trade-off between FP and FN was achieved, enabling the customization of parameters suitable for emphasizing high sensitivity in field screening tasks and high specificity in field validation tasks. Further exploring the capability of the detection system, this study has clearly validated the critical influence of distance and confidence threshold on the accuracy and performance of object detection, particularly evident in the detection of cherry tomatoes using UAV-based systems. At an effective range of 20 cm, wherein a mean confidence score of 86% was attained for ripe cherry tomato detection, it was validated that closer distances could significantly enhance the detection clarity and may reduce occlusion-related challenges, resulting in more successful detections. However, as the distance increases, detection certainty diminishes, highlighting the need for careful consideration of distance factors in developing reliable detection systems. Interestingly, it was identified that challenges such as occlusion can

doi:10.6342/NTU202401594

cause missed detections and cases of close proximity between tomatoes with the same class can cause identity or bounding box merging. Moreover, the outcomes of the study showed that BoT-SORT was effective in tracking the cherry tomatoes, as evidenced by Multi-Object Tracking Accuracy (MOTA) values between 74% and 87%. Furthermore, results of the study show that high sensitivity could be achieved with thresholds between 45% and 50%, whereas specificity is maximized with thresholds from 75% to 80%.

As a whole, while the findings of this study may be considered preliminary, they present valuable and significant insights for digital cherry tomato monitoring systems using UAV and AI technologies. Through a systematic analysis, four factors—namely distance, confidence threshold, occlusion, and inter-tomato proximity of the same class—were discovered to greatly influence the detection accuracy and certainty. This research could become a sufficient foundation for future improvements in digital agriculture automation, ensuring more reliable and efficient crop management.

5.2 Limitations and Future Perspectives of the Study

In this study, one major consideration depends on the UAV's programmability leading to the selection of the Tello drone as a case study. Thus, hardware limitations were unavoidable which may influence the quadrotor's overall performance. This includes the constraint in the drone's battery capacity which became a major consideration in making the system as simple and as computationally inexpensive as possible. Some areas that need immediate focus may be the mitigation of the observed delay through optimization of data transmission protocols or exploration of alternative communication mechanisms, and the possibility of initializing real-time operations. Moreover, autonomous navigation systems guided by fiducial markers or through the

doi:10.6342/NTU202401594

utilization of on-board UAV sensors may be integrated to further minimize human intervention within the process.

In the cherry tomato tracking experiments and comparative analysis phases of the UAV, the study only focused on utilizing ripe cherry tomatoes due to time constraints. Moreover, given the preliminary nature of the study, the effectiveness of confidence threshold fine-tuning depends on the availability of comprehensive prior data. Hence, future research should explore advanced methods to enhance detection and tracking robustness, particularly in scenarios lacking extensive prior knowledge. Furthermore, the discovery of all the identified factors which affects detection performance further emphasizes an area for further study to provide a more optimized and more precise detection and localization system.

Finally, the integration of the synergistic capability of Unmanned Aerial Vehicle (UAV) and Unmanned Ground Vehicle (UGV) systems could be a viable solution to mitigate the limitations of the UAV. A sophisticated data fusion method can be employed in future works with the availability of 3D point cloud data. Point clouds are extensively utilized across various domains, particularly in 3D object detection tasks. These involve not only assigning labels to point sets but also localizing objects of interest using 3D bounding boxes with remarkable accuracy. Researches are already being conducted to enhance 3D object detection methodologies despite the complexities behind image and point cloud data fusion. By integrating data from both Unmanned Vehicle (UV) systems, the system may be able to provide a more comprehensive understanding of the tomatoes maturity through ensembled prediction. This holistic view enables better-informed decision-making regarding a more efficient and precise autonomous cherry tomato harvesting and picking.

doi:10.6342/NTU202401594

References

Afonso, M., Fonteijn, H., Fiorentin, F. S., Lensink, D., Mooij, M., Faber, N., Polder, G., & Wehrens, R. (2020). Tomato Fruit Detection and Counting in Greenhouses Using Deep Learning. *Frontiers in Plant Science*, *11*. https://doi.org/10.3389/fpls.2020.571299.

Aharon, N., Orfaig, R., & Bobrovsky, B. (2022). BOT-SORT: Robust Associations Multi-Pedestrian Tracking. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2206.14651.

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A. Q., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. Journal of Big Data, 8(1). https://doi.org/10.1186/s40537-021-00444-8.

Amodei, D., Olah, C., Steinhardt, J., Christiano, P.F., Schulman, J., & Mané, D. (2016). Concrete Problems in AI Safety. ArXiv, abs/1606.06565.

Appe, S. N., Arulselvi, G., & Balaji, G. N. (2023). CAM-YOLO: tomato detection and classification based on improved YOLOv5 using combining attention mechanism. *PeerJ Computer Science*, *9*. https://doi.org/10.7717/peerj-cs.1463

Arani, E., Gowda, S., Mukherjee, R., Magdy, O., Kathiresan, S.K., & Zonooz, B. (2022). A Comprehensive Study of Real-Time Object Detection Networks Across Multiple Domains: A Survey. ArXiv, abs/2208.10895.

Arah, I. K., Ahorbo, G. K., Anku, E. K., Kumah, E. K., & Amaglo, H. (2016). Postharvest handling practices and treatment methods for tomato handlers in developing

doi:10.6342/NTU202401594

countries: a mini review. Advances in Agriculture, 2016, 1–8. https://doi.org/10.1155/2016/6436945.

Baek, S., Lim, J., Lee, J. G., McCarthy, M. J., & Kim, S. M. (2020). Investigation of the maturity changes of cherry tomato using magnetic resonance imaging. Applied Sciences (Basel), 10(15), 5188. https://doi.org/10.3390/app10155188.

Bhujbal, K., & Barahate, S. (2022). Custom Object detection Based on Regional Convolutional Neural Network & YOLOv3 With DJI Tello Programmable Drone. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.4101029.

Bhusnoor, Mallikarjun & Patel, Jyoti & Mehta, Akshara & Sainkar, Sandeep & Patel, Dhrumil & Mehendale, Ninad. (2023). Investigating The Impact Of Distance On Object Detection Accuracy in Unmanned Aerial Vehicle Systems Using MobileNetV3. 10.36227/techrxiv.24155598.

Bochkovskiy, A., Wang, C., & Liao, H. M. (2020). YOLOV4: Optimal speed and accuracy of object detection. arXiv (Cornell University). https://arxiv.org/pdf/2004.10934v1.

Boonsongsrikul, A., & Eamsaard, J. (2023). Real-Time Human Motion Tracking by Tello EDU Drone. *Sensors*, *23*(2). https://doi.org/10.3390/s23020897.

Camacho, J. C., & Morocho-Cayamcela, M. E. (2023). Mask R-CNN and YOLOv8 Comparison to Perform Tomato Maturity Recognition Task. *Communications in Computer and Information Science*, *1885 CCIS*, 382–396. https://doi.org/10.1007/978-3-031-45438-7_26.

Cocchioni, F., Pierfelice, V., Benini, A., Mancini, A., Frontoni, E., Zingaretti, P., Ippoliti, G., & Longhi, S. (2014). Unmanned Ground and Aerial Vehicles in extended range

doi:10.6342/NTU202401594

indoor and outdoor missions. 2014 International Conference on Unmanned Aircraft Systems (ICUAS). https://doi.org/10.1109/icuas.2014.6842276.

Christensen, M. J., & Richter, T. (2020). Achieving reliable UDP transmission at 10 Gb/s using BSD socket for data acquisition systems. Journal of Instrumentation, 15(09), T09005. https://doi.org/10.1088/1748-0221/15/09/t09005.

de la Escalera, A., & Armingol, J. M. (2010). Automatic chessboard detection for intrinsic and extrinsic camera parameter calibration. *Sensors*, *10*(3), 2027–2044. https://doi.org/10.3390/s100302027.

de Luna, R. G., Dadios, E. P., Bandala, A. A., & Vicerra, R. R. P. (2019). *Tomato Fruit Image Dataset for Deep Transfer Learning-based Defect Detection*. https://doi.org/10.1109/CIS-RAM47153.2019.9095778.

De Silva, S. C., Phlernjai, M., Rianmora, S., & Ratsamee, P. (2022). Inverted Docking Station: A Conceptual Design for a Battery-Swapping Platform for Quadrotor UAVs. *Drones*, *6*(3). https://doi.org/10.3390/drones6030056.

Ding, Z., Lai, Z., Li, S., Li, P., Yang, Q., & Wong, E.K. (2019). Confidence-Triggered Detection: Accelerating Real-time Tracking-by-detection Systems. Retrieved 14 May 2024 at https://arxiv.org/html/1902.00615v4.

DJI. (2018). DJI SDK for Tello-Python. Retrieved 17 October 2023 at https://github.com/dji-sdk/Tello-Python/blob/master/doc/readme.pdf.

Douterloigne, K., Gautama, S., & Philips, W. (2009). Fully automatic and robust UAV camera calibration using chessboard patterns. 2009 IEEE International Geoscience and Remote Sensing Symposium, 2, II-551-II-554.

doi:10.6342/NTU202401594

Duan, E., Han, G., Zhao, S., Ma, Y., Lv, Y., & Bai, Z. (2023). Regulation of Meat Duck Activeness through Photoperiod Based on Deep Learning. *Animals (Basel), 13*(22), 3520. https://doi.org/10.3390/ani13223520.

Egi, Y., Hajyzadeh, M., & Eyceyurt, E. (2022). Drone-Computer Communication Based Tomato Generative Organ Counting Model Using YOLO V5 and Deep-Sort. Agriculture, 12(9), 1290. https://doi.org/10.3390/agriculture12091290.

El-Bendary, N., Hariri, E. E., Hassanien, A. E., & Badr, A. (2015). Using machine learning techniques for evaluating tomato ripeness. Expert Systems with Applications, 42(4), 1892–1905. https://doi.org/10.1016/j.eswa.2014.09.057.

El-hariri, Esraa & El-Bendary, Nashwa & Hassanien, Aboul Ella & Badr, Amr. (2014). AUTOMATED RIPENESS ASSESSMENT SYSTEM OF TOMATOES USING PCA AND SVM TECHNIQUES. Retrieved 5 February 2024 at https://www.researchgate.net/publication/263889773_AUTOMATED_RIPENESS_ASSESSMENT_SYSTEM_OF_TOMATOES_USING_PCA_AND_SVM_TECHNIQUES.

Ge, Y., Lin, S., Zhang, Y., Li, Z., Cheng, H., Dong, J., Shao, S., Zhang, J., Qi, X., & Wu, Z. (2022). Tracking and counting of tomato at different growth period using an improving YOLO-Deepsort network for inspection robot. Machines, 10(6), 489. https://doi.org/10.3390/machines10060489.

Giap, Y. C., Muljono, M., Soeleman, M. A., Affandy, & Basuki, R. S. (2023). Effect of Distance and Light Intensity on Multiple Detection Object. 2023 6th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI). https://doi.org/10.1109/isriti60336.2023.10467935.

doi:10.6342/NTU202401594

Giernacki, W., Rao, J., Sladic, S., Bondyra, A., Retinger, M., & Espinoza-Fraire, T. (2022). DJI Tello Quadrotor as a Platform for Research and Education in Mobile Robotics and Control Engineering. *2022 International Conference on Unmanned Aircraft Systems, ICUAS 2022*, 735–744. https://doi.org/10.1109/ICUAS54217.2022.9836168.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. 2014 IEEE Conference on Computer Vision and Pattern Recognition. https://doi.org/10.1109/cvpr.2014.81.

Girshick, R. (2015). Fast R-CNN. 2015 IEEE International Conference on Computer Vision (ICCV). https://doi.org/10.1109/iccv.2015.169.

Grammatikopoulos, L., Karras, G., & Petsa, E. (2007). An automatic approach for camera calibration from vanishing points. *ISPRS Journal of Photogrammetry and Remote Sensing*, *62*(1), 64–76. https://doi.org/10.1016/j.isprsjprs.2007.02.002.

Gruen, Armin. (2001). Calibration and Orientation of Cameras in Computer Vision. 10.1007/978-3-662-04567-1.

Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., & Lew, M. S. (2016). Deep learning for visual understanding: A review. Neurocomputing, 187, 27–48. https://doi.org/10.1016/j.neucom.2015.09.116.

Hall, D. R., Dayoub, F., Skinner, J., Zhang, H., Miller, D., Corke, P., Carneiro, G., Angelova, A., & Sünderhauf, N. (2020). Probabilistic Object Detection: Definition and Evaluation. 2020 IEEE Winter Conference on Applications of Computer Vision (WACV). https://doi.org/10.1109/wacv45572.2020.9093599.

doi:10.6342/NTU202401594

Hamdu, I. H., Yunus, A., & Hardi, I. M. (2016). Maturity and ripening on the biochemical characteristics of three local varieties of tomato. DOAJ (DOAJ: Directory of Open Access Journals). https://doaj.org/article/b393275265b64d90b30d465fa5a1570c.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 37(9), 1904–1916. https://doi.org/10.1109/tpami.2015.2389824.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 2961-2969).

Hendrawan, Rido & Istiono, Wirawan. (2023). Analyzing the Distance and Intensity of Light in Learning Augmented Reality Marker Based Tracking Application. Journal of Advances in Mathematics and Computer Science. 58-70. 10.9734/jamcs/2023/v38i21746.

Hnoohom, N., Chotivatunyu, P., Maitrichit, N., Nilsumrit, C., & Iamtrakul, P. (2024). The video-based safety methodology for pedestrian crosswalk safety measured: The case of Thammasat University, Thailand. Transportation Research Interdisciplinary Perspectives, 24, 101036. https://doi.org/10.1016/j.trip.2024.101036.

Host, K., Pobar, M., & Ivašić-Kos, M. (2023). Analysis of movement and activities of handball players using deep neural networks. Journal of Imaging, 9(4), 80. https://doi.org/10.3390/jimaging9040080.

Hou, Q., Zhou, D., & Feng, J. (2021). Coordinate Attention for Efficient Mobile Network Design. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.1109/cvpr46437.2021.01350.

doi:10.6342/NTU202401594

Hubel, D. H., & Wiesel, T. N. (1968). Receptive fields and functional architecture of monkey striate cortex. The Journal of Physiology, 195(1), 215–243. https://doi.org/10.1113/jphysiol.1968.sp008455.

Junaid, A. B., Lee, Y., & Kim, Y. (2016). Design and implementation of autonomous wireless charging station for rotary-wing UAVs. Aerospace Science and Technology, 54, 253–266. https://doi.org/10.1016/j.ast.2016.04.023.

Junaid, A. B., Konoiko, A., Zweiri, Y., Sahinkaya, M. N., & Seneviratne, L. (2017). Autonomous wireless Self-Charging for Multi-Rotor unmanned aerial vehicles. Energies, 10(6), 803. https://doi.org/10.3390/en10060803.

Kaur, R., & Singh, S. (2023). A comprehensive review of object detection with deep learning. Digital Signal Processing, 132, 103812. https://doi.org/10.1016/j.dsp.2022.103812.

Kemper, F.P., Suzuki, K.A.O. & Morrison, J.R. UAV Consumable Replenishment: Design Concepts for Automated Service Stations. *J Intell Robot Syst* **61**, 369–397 (2011). https://doi.org/10.1007/s10846-010-9502-z.

Kimura, S., & Sinha, N. (2008). Tomato (Solanum lycopersicum): A model fruit-bearing crop. *Cold Spring Harbor Protocols*, *3*(11). https://doi.org/10.1101/pdb.emo105.

Kung, R., Pan, N., Wang, C. C., & Lee, P. (2021). Application of deep learning and unmanned aerial vehicle on building maintenance. Advances in Civil Engineering, 2021, 1–12. https://doi.org/10.1155/2021/5598690.

Kurtulmuş, F., Lee, W. S., & Vardar, A. (2013). Immature peach detection in colour images acquired in natural illumination conditions using statistical classifiers and

doi:10.6342/NTU202401594

neural network. Precision Agriculture, 15(1), 57–79. https://doi.org/10.1007/s11119-013-9323-8.

Larmo, A., Ratilainen, A., & Saarinen, J. (2018). Impact of COAP and MQTT on NB-IoT system performance. Sensors, 19(1), 7. https://doi.org/10.3390/s19010007.

Lawal, M. O. (2021). Tomato detection based on modified YOLOv3 framework. *Scientific Reports*, *11*(1). https://doi.org/10.1038/s41598-021-81216-5

Leahy, K., Zhou, D., Vasile, C., Oikonomopoulos, K., Schwager, M., & Belta, C. (2015). Persistent surveillance for unmanned aerial vehicles subject to charging and temporal logic constraints. Autonomous Robots, 40(8), 1363–1378. https://doi.org/10.1007/s10514-015-9519-z.

Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., Ke, Z., Li, Q., Cheng, M., Nie, W., Li, Y., Zhang, B., Liang, Y., Zhou, L., Xu, X., Chu, X., Wei, X., & Wei, X. (2022). YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. ArXiv, abs/2209.02976.

Li, P., Zheng, J., Li, P., Long, H., Li, M., & Gao, L. (2023). Tomato Maturity Detection and Counting Model Based on MHSA-YOLOv8. *Sensors*, *23*(15). https://doi.org/10.3390/s23156701.

Li, R., Ji, Z., Hu, S., Huang, X., Yang, J., & Li, W. (2023). Tomato Maturity Recognition Model Based on Improved YOLOv5 in Greenhouse. *Agronomy*, *13*(2). https://doi.org/10.3390/agronomy13020603.

Lin, T., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature Pyramid Networks for Object Detection. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.1109/cvpr.2017.106.

doi:10.6342/NTU202401594

Liu, G., Christian, N. J., Mbouembe, P. L. T., & Kim, J. H. (2020). YOLO-Tomato: A robust algorithm for tomato detection based on YOLOV3. Sensors, 20(7), 2145. https://doi.org/10.3390/s20072145.

Liu, G., Mao, S., & Kim, J. H. (2019). A Mature-Tomato detection algorithm using machine learning and color analysis. Sensors, 19(9), 2023. https://doi.org/10.3390/s19092023.

Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., & Pietikäinen, M. (2019). Deep Learning for Generic Object Detection: A survey. International Journal of Computer Vision, 128(2), 261–318. https://doi.org/10.1007/s11263-019-01247-4.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C., & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. In Lecture Notes in Computer Science (pp. 21–37). https://doi.org/10.1007/978-3-319-46448-0_2.

Luo, L., Tang, Y., Zou, X., Chen, Z., Zhang, P., & Feng, W. (2016). Robust grape cluster detection in a vineyard by combining the ADABoost framework and multiple color components. Sensors, 16(12), 2098. https://doi.org/10.3390/s16122098.

Ma, Linh & Hussain, Muhammad Ishfaq & Park, JongHyun & Kim, Jeongbae & Jeon, Moongu. (2023). Adaptive Confidence Threshold for ByteTrack in Multi-Object Tracking. 370-374. 10.1109/ICCAIS59597.2023.10382403.

Millot, Y., & Man, P. P. (2012). Active and passive rotations with Euler angles in NMR. *Concepts in Magnetic Resonance Part A: Bridging Education and Research*, *40 A* (5), 215–252. https://doi.org/10.1002/cmr.a.21242.

doi:10.6342/NTU202401594

Mirzaei, B., Nezamabadi-pour, H., Raoof, A., & Derakhshani, R. (2023). Small Object Detection and Tracking: A Comprehensive review. Sensors, 23(15), 6887. https://doi.org/10.3390/s23156887.

Monaghan, T. F., Rahman, S. N., Agudelo, C. W., Wein, A. J., Lazar, J. M., Everaert, K., & Dmochowski, R. R. (2021). Foundational Statistical Principles in Medical Research: Sensitivity, Specificity, Positive Predictive Value, and Negative Predictive Value. Medicina (Kaunas, Lithuania), 57(5), 503. https://doi.org/10.3390/medicina57050503.

Moneruzzaman, K. M., Hossain, A. B. M. S., Sani, W., Saifuddin, M., & Alenazi, M. (2009). Effect of harvesting and storage conditions on the post harvest quality of tomato (Lycopersicon esculentum Mill) cv. Roma VF. Australian Journal of Crop Science, 3(2), 113–121. http://www.cropj.com/Moneeruzaman_3_2_2009.pdf.

Mostafa, T. M., Muharam, A., & Hattori, R. (2017). Wireless battery charging system for drones via capacitive power transfer. 2017 IEEE PELS Workshop on Emerging Technologies: Wireless Power Transfer (WoW). https://doi.org/10.1109/wow.2017.7959357.

Mourgelas, Christos & Kokkinos, Sokratis & Milidonis, Athanasios & Voyiatzis, Ioannis. (2020). Autonomous drone charging stations: A survey. 10.1145/3437120.3437314.

Mulgaonkar, Yash. (2014). Autonomous Charging to Enable Long-Endurance Missions for Small Aerial Robots. 90831S. 10.1117/12.2051111.

doi:10.6342/NTU202401594

Ni, J., Zhu, S., Tang, G., Ke, C., & Wang, T. (2024). A Small-Object Detection Model Based on Improved YOLOv8s for UAV Image Scenarios. Remote Sensing, 16, 2465. https://doi.org/10.3390/rs16132465.

Njume, C. A., Ngosong, C., Krah, C. Y., & Mardjan, S. (2020). Tomato food value chain: managing postharvest losses in Cameroon. IOP Conference Series. Earth and Environmental Science, 542(1), 012021. https://doi.org/10.1088/1755-1315/542/1/012021.

Ntouskos, V., Karras, G., Douskos, V., & Kalisperakis, I. (2007). *Automatic calibration of digital cameras using planar chess-board patterns View project AUTOMATIC CALIBRATION OF DIGITAL CAMERAS USING PLANAR CHESS-BOARD PATTERNS*. https://www.researchgate.net/publication/228345254.

Phan, Q., Nguyen, V., Lien, C., Duong, T., Hou, M. T., & Le, N. (2023). Classification of tomato fruit using YoLov5 and convolutional neural network models. Plants, 12(4), 790. https://doi.org/10.3390/plants12040790.

Radiansyah, S & Kusrini, Mirza & Prasetyo, Lilik. (2017). Quadcopter applications for wildlife monitoring. IOP Conference Series: Earth and Environmental Science. 54. 012066. 10.1088/1755-1315/54/1/012066.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7263-7271).

doi:10.6342/NTU202401594

Redmon, J., & Farhadi, A. (2018). YOLOV3: an incremental improvement. arXiv (Cornell University). https://arxiv.org/pdf/1804.02767.

Rekavandi, Aref & Xu, Lian & Boussaid, Farid & Seghouane, Abd-Krim & Hoefs, Stephen & Bennamoun, Mohammed. (2022). A Guide to Image and Video based Small Object Detection using Deep Learning: Case Study of Maritime Surveillance. 10.48550/arXiv.2207.12926.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28.

Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., & Savarese, S. (2019). Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. IEEE/CVF Conference on Computer Vision and Pattern Recognition. https://doi.org/10.1109/cvpr.2019.00075.

Rodríguez-Ortega, W. M., Martínez, V., Nieves, M., Simón, I., Lidón, V., Fernandez-Zapata, J. C., Martinez-Nicolas, J. J., Cámara-Zapata, J. M., & García-Sánchez, F. (2019). Agricultural and Physiological Responses of Tomato Plants Grown in Different Soilless Culture Systems with Saline Water under Greenhouse Conditions. *Scientific Reports*, *9*(1). https://doi.org/10.1038/s41598-019-42805-7

Rong, J., Zhou, H., Zhang, F., Yuan, T., & Wang, P. (2023). Tomato cluster detection and counting using improved YOLOv5 based on RGB-D fusion. Computers and Electronics in Agriculture, 207, 107741. https://doi.org/10.1016/j.compag.2023.107741.

doi:10.6342/NTU202401594

Ryze Robotics (2018). Tello SDK 2.0 User Guide. https://dl-cdn.ryzerobotics.com/downloads/Tello/Tello%20SDK%202.0%20User%20Guide.pdf.

Sacchi, E., Sayed, T., & deLeur, P. (2013). A comparison of collision-based and conflict-based safety evaluations: The case of right-turn smart channels. Accident Analysis and Prevention, 59, 260–266. https://doi.org/10.1016/j.aap.2013.06.002.

Saleem, M.H., Potgieter, J. & Arif, K.M. (2021). Automation in Agriculture by Machine and Deep Learning Techniques: A Review of Recent Developments. Precision Agric 22, 2053–2091. https://doi.org/10.1007/s11119-021-09806-x.

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229.

Song, B. D., Kim, J., Kim, J., Park, H., Morrison, J. R., and Shim, D. H. (2013). "Persistent UAV service: An improved scheduling formulation and prototypes of system components," *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, Atlanta, GA, USA, pp. 915-925, doi: 10.1109/ICUAS.2013.6564777.

Stanojevic, V. D., & Todorović, B. (2024). BoostTrack: boosting the similarity measure and detection confidence for improved multiple object tracking. Machine Vision and Applications, 35(3). https://doi.org/10.1007/s00138-024-01531-5.

Su, K., Cao, L., Zhao, B., Li, N., Wu, D., & Han, X. (2023). N-IoU: better IoU-based bounding box regression loss for object detection. Neural Computing & Applications. https://doi.org/10.1007/s00521-023-09133-4.

doi:10.6342/NTU202401594

Sünderhauf, N., Brock, O., Scheirer, W. J., Hadsell, R., Fox, D., Leitner, J., Upcroft, B., Abbeel, P., Burgard, W., Milford, M., & Corke, P. (2018). The limits and potentials of deep learning for robotics. The International Journal of Robotics Research, 37(4–5), 405–420. https://doi.org/10.1177/0278364918770733.

Szegedy, C., Toshev, A., & Erhan, D. (2013). Deep neural networks for object detection. Neural Information Processing Systems, 26, 2553–2561. https://papers.nips.cc/paper/5207-deep-neural-networks-for-object-detection.pdf.

Szepessy, Tamas. (2019). Optical Drone Control. Retrieved 5 June 2023 at https://github.com/TamasSzepessy/DJITelloOpticalControl/tree/master.

Tang, C., Feng, Y., Yang, X., Zheng, C., & Zhou, Y. (2017). The Object Detection Based on Deep Learning. 2017 4th International Conference on Information Science and Control Engineering (ICISCE). https://doi.org/10.1109/icisce.2017.156.

Tapia-Mendez, E., Cruz-Albarrán, I. A., Tovar-Arriaga, S., & Morales-Hernández, L. A. (2023). Deep Learning-Based Method for Classification and Ripeness Assessment of fruits and vegetables. Applied Sciences, 13(22), 12504. https://doi.org/10.3390/app132212504.

Terven, J. R., Córdova-Esparza, D., & Romero-González, J. (2023a). A comprehensive review of YOLO architectures in computer vision: from YOLOV1 to YOLOV8 and YOLO-NAS. Machine Learning and Knowledge Extraction, 5(4), 1680–1716. https://doi.org/10.3390/make5040083.

Terven, Juan & Cordova-Esparza, Diana-Margarita & Ramirez-Pedraza, Alfonzo & Chavez-Urbiola, Edgar. (2023b). Loss Functions and Metrics in Deep Learning. A Review. Retrieved 2/29/2024 at

doi:10.6342/NTU202401594

https://www.researchgate.net/publication/372163006_Loss_Functions_and_Metrics_in_Deep_Learning_A_Review.

Thuan, D. (2021). Evolution of Yolo algorithm and Yolov5: The State-of-the-Art object detention algorithm. Retrieved 3/20/2024 at https://www.semanticscholar.org.

Tian, Y., Su, D., Lauria, S., & Liu, X. (2022). Recent advances on loss functions in deep learning for computer vision. Neurocomputing, 497, 129–158. https://doi.org/10.1016/j.neucom.2022.04.127.

Tolasa, M., Gedamu, F., & Woldetsadik, K. (2021). Impacts of harvesting stages and pre-storage treatments on shelf life and quality of tomato (Solanum lycopersicum L.). Cogent Food & Agriculture, 7(1). https://doi.org/10.1080/23311932.2020.1863620.

Tong, Z., Chen, Y., Xu, Z., & Yu, R. (2023). Wise-IoU: Bounding Box Regression Loss with Dynamic Focusing Mechanism. ArXiv, abs/2301.10051.

Tsai, F. T., Nguyen, V., Duong, T., Phan, Q., & Lien, C. (2023). Tomato Fruit Detection Using Modified Yolov5m Model with Convolutional Neural Networks. Plants, 12(17), 3067. https://doi.org/10.3390/plants12173067.

Tsironis, V., Bourou, S., & Stentoumis, C. (2020). Tomatod: Evaluation of object detection algorithms on a new real-world tomato dataset. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, *43*(B3), 1077–1084. https://doi.org/10.5194/isprs-archives-XLIII-B3-2020-1077-2020.

Tsouvaltzis, P., Gkountina, S., & Siomos, A. S. (2023). Quality traits and nutritional components of cherry tomato in relation to the harvesting period, storage duration

doi:10.6342/NTU202401594

and fruit position in the truss. Plants, 12(2), 315. https://doi.org/10.3390/plants12020315.

Wang, C., Wang, C., Hu, X., Wang, J., Liao, J., Li, Y., & Lan, Y. (2023). A lightweight cherry tomato maturity Real-Time Detection algorithm based on improved YOLOV5N. Agronomy, 13(8), 2106. https://doi.org/10.3390/agronomy13082106.

Wang, C., Bochkovskiy, A., & Liao, H.M. (2022). YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 7464-7475.

Wang, Z., Zhang, X., Su, Y., Li, W., Yin, X., Li, Z., Ying, Y., Wang, J., Wu, J., Miao, F., & Zhao, K. (2023). Abnormal Behavior Monitoring Method of Larimichthys crocea in Recirculating Aquaculture System Based on Computer Vision. *Sensors*, *23*(5), 2835. https://doi.org/10.3390/s23052835.

Wenkel, S., Alhazmi, K., Liiv, T., Alrshoud, S. R., & Simón, M. (2021). Confidence Score: The Forgotten Dimension of Object Detection Performance Evaluation. Sensors, 21(13), 4350. https://doi.org/10.3390/s21134350.

Wojke, N., Bewley, A., & Paulus, D. (2017). Simple online and realtime tracking with a deep association metric. 2017 IEEE International Conference on Image Processing (ICIP). https://doi.org/10.1109/icip.2017.8296962.

Yamashita, R., Nishio, M., Gian, R. K., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. Insights Into Imaging, 9(4), 611–629. https://doi.org/10.1007/s13244-018-0639-9.

Yang, Chuankai & He, Yuanjian & Qu, Haoyue & Wu, Jingfeng & Hou, Zhe & Lin, Zhongzheng & Cai, Changsong. (2019). Analysis, design and implement of

doi:10.6342/NTU202401594

asymmetric coupled wireless power transfer systems for unmanned aerial vehicles. AIP Advances. 9. 025206. 10.1063/1.5080955.

Yu, C., Feng, Z., Wu, Z., Wei, R., Song, B., & Cao, C. (2023). HB-YOLO: an improved YOLOV7 algorithm for DIM-Object tracking in satellite remote sensing videos. Remote Sensing (Basel), 15(14), 3551. https://doi.org/10.3390/rs15143551.

Yu, H., Pei, H., Lyu, Y., Yuan, Z., Rizzo, J. R., Wang, Y., & Fang, Y. (2023). Understanding the Impact of Image Quality and Distance of Objects to Object Detection Performance. 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). https://doi.org/10.1109/iros55552.2023.10342139.

Yuan, T., Li, L., Zhang, F., Fu, J., Gao, J., Zhang, J., Li, W., Zhang, C., & Zhang, W. (2020). Robust cherry tomatoes detection algorithm in Greenhouse Scene based on SSD. Agriculture, 10(5), 160. https://doi.org/10.3390/agriculture10050160.

Zhang, X., Yang, Y. H., Han, Z., Wang, H., & Gao, C. (2013). Object class detection: A survey. *ACM Computing Surveys (CSUR)*, *46*(1), 1-53.

Zhang, Y. et al. (2022). ByteTrack: Multi-object Tracking by Associating Every Detection Box. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds) Computer Vision – ECCV 2022. ECCV 2022. Lecture Notes in Computer Science, vol 13682. Springer, Cham. https://doi.org/10.1007/978-3-031-20047-2_1.

Zhang, Z. (2000). A Flexible New Technique for Camera Calibration. In *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE* (Vol. 22, Issue 11).

doi:10.6342/NTU202401594

Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X. (2019). Object Detection With Deep Learning: A Review. IEEE transactions on neural networks and learning systems, 30(11), 3212–3232. https://doi.org/10.1109/TNNLS.2018.2876865.

Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., & Ren, D. (2020). Distance-IOU loss: Faster and better learning for bounding box regression. Proceedings of the AAAI Conference on Artificial Intelligence, 34(07), 12993–13000. https://doi.org/10.1609/aaai.v34i07.6999.

Zheng, Z., Wang, P., Ren, D., Liu, W., Ye, R., Hu, Q., & Zuo, W. (2022). Enhancing geometric factors in model learning and inference for object detection and instance segmentation. IEEE Transactions on Cybernetics, 52(8), 8574–8586. https://doi.org/10.1109/tcyb.2021.3095305.

Zou, Z., Chen, K., Shi, Z., Guo, Y., & Ye, J. (2023). Object Detection in 20 years: A survey. Proceedings of the IEEE, 111(3), 257–276. https://doi.org/10.1109/jproc.2023.3238524.

doi:10.6342/NTU202401594

Appendices

Appendix A
Version of Libraries used in the Study

Table A1. Drone-Computer Communication System

Library	Version
Python	3.9.13
djitellopy	2.5.0
opecv-python	4.5.3.56
opecv-contrib-python	4.5.3.56
pygame	2.5.2
Scipy	1.11.4
pyYAML	6.0.1

Table A2. Detection and Tracking System

Library	Version
Python	3.9.13
ultralytics	8.1.5
torch	2.0.1+cu117
torchaudio	2.0.2+cu117
torchvision	0.15.2+cu117
numpy	1.24.3
matplotlib	3.8.2
pandas	2.1.4
Scipy	1.11.4
pyYAML	6.0.1
labelImg	1.8.6
motmetrics	1.1.2

doi:10.6342/NTU202401594

Appendix B
Dedicated keys for UAV control

Key	Key down Function	Key up Function
Arrow key: Up and Down	The key responsible for setting up forward and backward velocity.	The key that initiates a zero-velocity state for both forward and backward translation.
Arrow key: Left and Right	The key responsible for setting up left and right velocity.	The key that initiates a zero-velocity state for sideward translation.
w and s	The keys employed to establish the vertical velocity parameters for ascent and descent.	The key that initiates a zero-velocity state for vertical translation.
a and d	The key responsible for setting the velocity of yaw in both clockwise and counterclockwise direction.	The key for setting the yaw velocity to zero.
t	Not Applicable	The essential key for initiating the take-off command.
l	Not Applicable	The key necessary to initiate the landing command.
m	Not Applicable	The key used to save a video frame as image.
r	Not Applicable	The key used to record a video sequence.

doi:10.6342/NTU202401594

Appendix C

Performance of Detection Model Trained under Different Training Configurations

Table C1. Training Configuration A

		Precision	Recall	F1-Score	mAP
Test Set A	**Overall**	**83.0**	**85.1**	**84.04**	**89.6**
	Green	89.0	87.6	88.29	93.2
	Semi-ripe	76.3	83.2	79.60	86.9
	Fully-ripe	83.5	84.6	84.05	88.8
Test Set B	**Overall**	**76.0**	**76.6**	**76.30**	**80.6**
	Green	74.4	61.2	67.16	71.8
	Semi-ripe	76.9	79.8	78.32	79.9
	Fully-ripe	76.8	88.9	82.41	90.1
Test Set C	**Overall**	**75.2**	**78.9**	**77.01**	**82.5**
	Green	80.8	77.9	79.32	84.1
	Semi-ripe	74.1	69.6	71.78	78.2
	Fully-ripe	70.7	89.3	78.92	85.2
Test Set D	**Overall**	**82.0**	**81.7**	**81.85**	**87.7**
	Green	87.0	82.2	84.53	90.2
	Semi-ripe	77.1	78.2	77.65	84.5
	Fully-ripe	81.2	84.6	82.87	88.3

Table C2. Training Configuration B

		Precision	Recall	F1-Score	mAP
Test Set A	**Overall**	**81.8**	**85.5**	**83.61**	**89.6**
	Green	89.2	85.6	87.36	92.9
	Semi-ripe	73.0	85.5	78.76	85.3
	Fully-ripe	83.1	85.4	84.23	90.5
Test Set B	**Overall**	**82.5**	**82.2**	**82.35**	**89.7**
	Green	83.5	75.4	13.60	88.9
	Semi-ripe	78.1	80.0	79.04	85.3
	Fully-ripe	85.8	91.0	88.32	94.8
Test Set C	**Overall**	**80.4**	**78.1**	**79.23**	**84.2**
	Green	81.9	71.6	76.40	81.4
	Semi-ripe	80.9	76.8	78.80	83.6
	Fully-ripe	78.4	85.9	81.98	87.7
Test Set D	**Overall**	**81.7**	**83.7**	**82.69**	**88.7**
	Green	88.0	82.4	85.11	90.9
	Semi-ripe	74.6	82.5	78.35	84.5
	Fully-ripe	82.5	86.3	84.36	90.6

doi:10.6342/NTU202401594

Table C3. Training Configuration C

		Precision	Recall	F1-Score	mAP
Test Set A	**Overall**	**84.0**	**84.5**	**84.25**	**89.7**
	Green	90.8	86.7	88.70	93.6
	Semi-ripe	75.5	85.2	80.06	86.3
	Fully-ripe	85.7	81.6	83.60	89.3
Test Set B	**Overall**	**84.1**	**83.8**	**83.95**	**90.6**
	Green	88.2	78.4	83.01	90.6
	Semi-ripe	77.0	85.3	80.94	86.4
	Fully-ripe	87.0	87.7	87.35	94.9
Test Set C	**Overall**	**77.9**	**83.6**	**80.65**	**87.6**
	Green	82.9	83.8	83.35	88.6
	Semi-ripe	76.3	77.4	76.85	84.5
	Fully-ripe	74.6	89.5	81.37	89.6
Test Set D	**Overall**	**84.1**	**83.1**	**83.60**	**89.4**
	Green	90.2	84.2	87.10	92.6
	Semi-ripe	76.7	82.1	79.31	85.5
	Fully-ripe	85.5	82.9	84.18	90.1

doi:10.6342/NTU202401594

Appendix D

Performance of the Detection Model for Overall Monitoring and Surveillance

		Precision	Recall	F1-Score	mAP
YOLOV8n	**Overall**	**80.8**	**82.2**	**81.49**	**87.5**
	Green	86.0	83.8	84.89	90.9
	Semi-ripe	76.8	77.4	77.10	83.1
	Fully-ripe	79.7	85.4	82.45	88.5
YOLOV8n + WIoU	**Overall**	**84.1**	**81.4**	**82.73**	**87.6**
	Green	89.9	83.4	86.53	91.9
	Semi-ripe	79.7	77.8	78.74	82.5
	Fully-ripe	82.8	83	82.90	88.4
YOLOV8n + CA	**Overall**	**83.6**	**82.3**	**82.94**	**89.0**
	Green	89.7	83.1	86.27	91.7
	Semi-ripe	78.3	78.8	78.55	85.8
	Fully-ripe	82.9	85.2	84.03	89.4
YOLOV8n + CA + WIoU	**Overall**	**81.4**	**82.8**	**82.09**	**88.1**
	Green	85	85.6	85.30	91.3
	Semi-ripe	76.3	79.4	77.82	83.8
	Fully-ripe	82.8	83.4	83.10	89.2

doi:10.6342/NTU202401594

Appendix E
Precision-Recall Curves

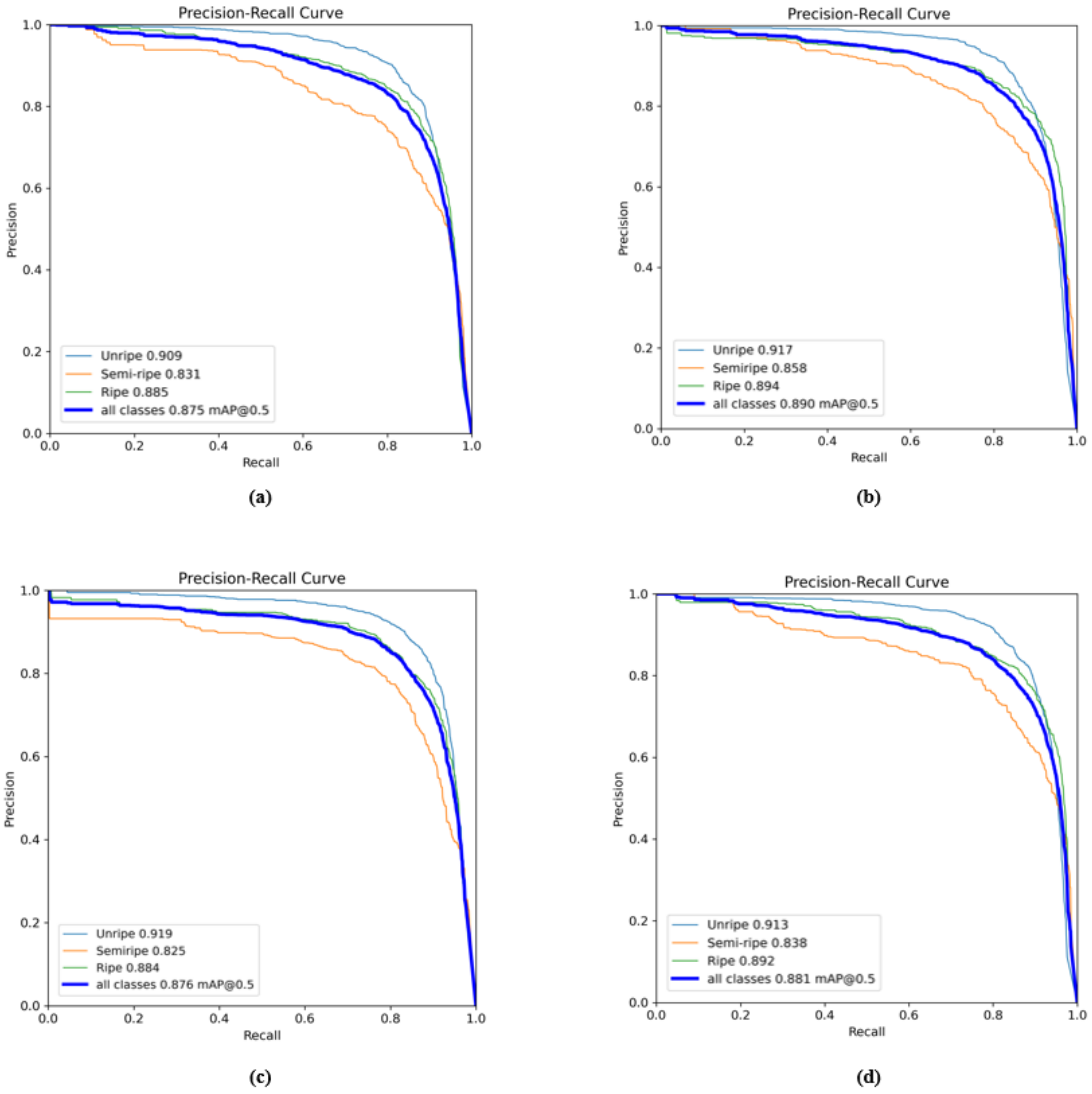


Figure E1. Surveillance and Monitoring Detection Model's Performance Parameters: (a) YOLOv8n; (b) YOLOv8n + CA; (c) YOLOv8n + WIoU; (d) YOLOv8n + CA + WIoU

doi:10.6342/NTU202401594

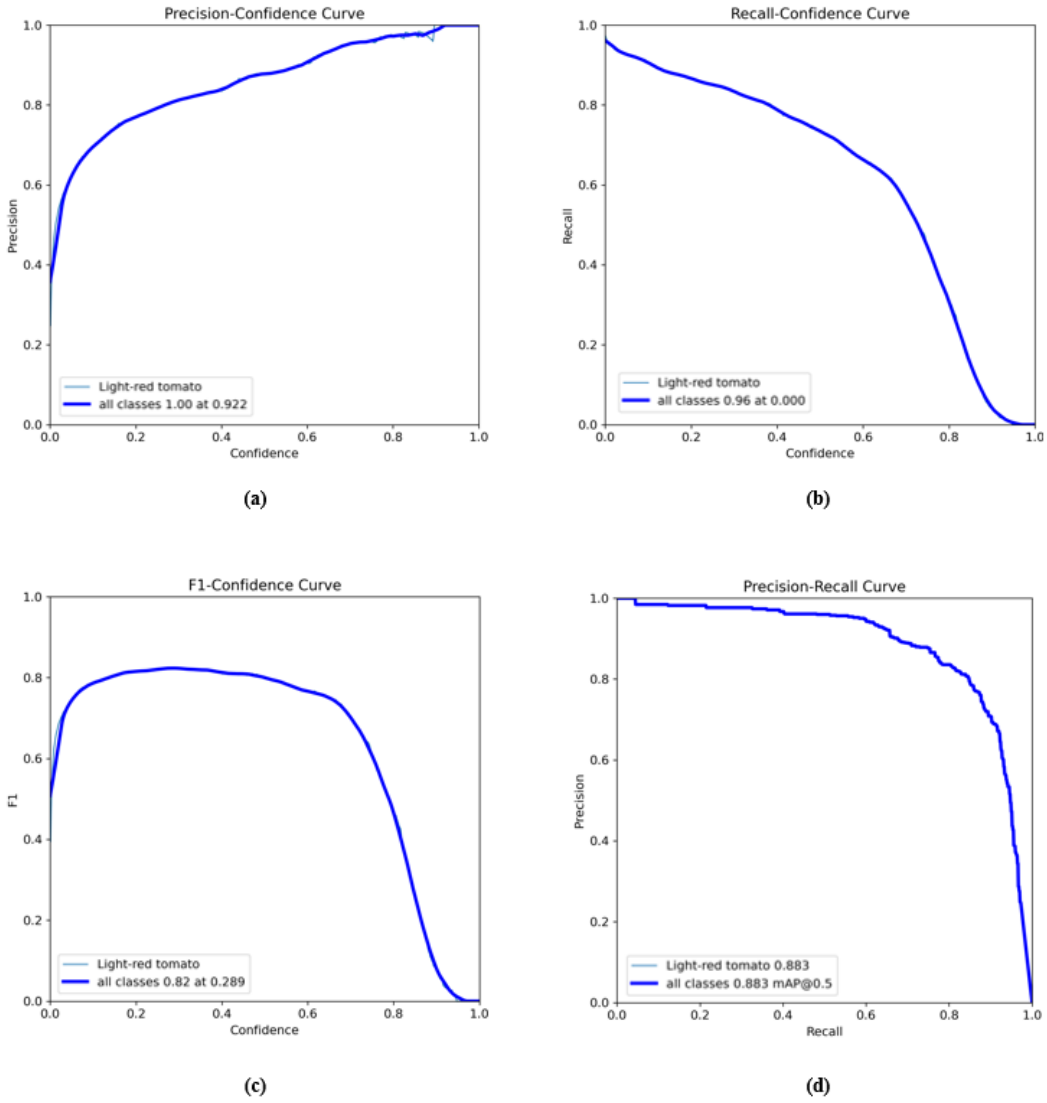


Figure E2. Light-Red Cherry Tomato Detection Model's Performance Parameters: (a) P-Curve; (b) R-Curve; (c) F1-Curve; and (D) PR-Curve

doi:10.6342/NTU202401594

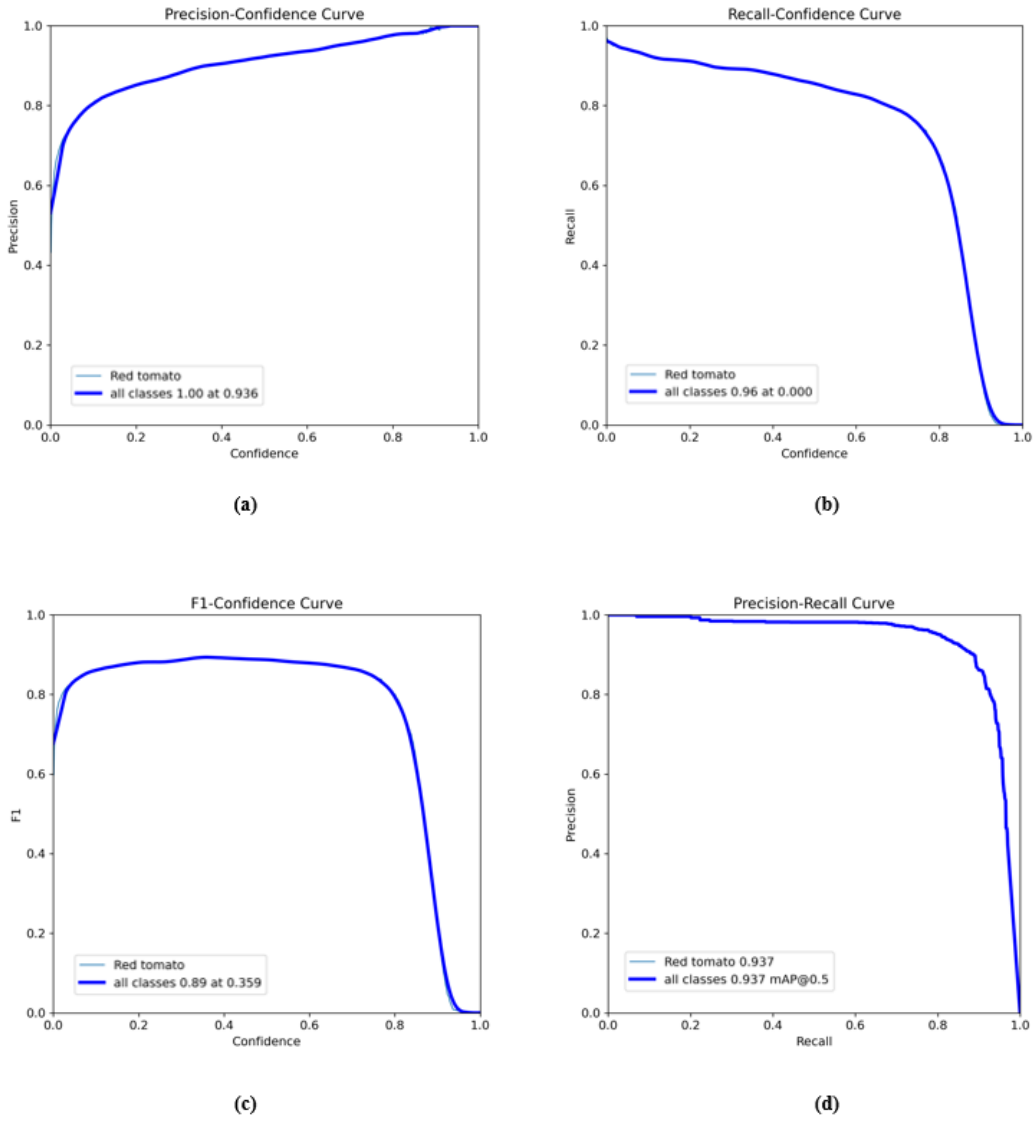


Figure E3. Red Cherry Tomato Detection Model's Performance Parameters: (a) P-Curve; (b) R-Curve; (c) F1-Curve; and (D) PR-Curve

doi:10.6342/NTU202401594

Appendix F
Training and Validation Performance Curves

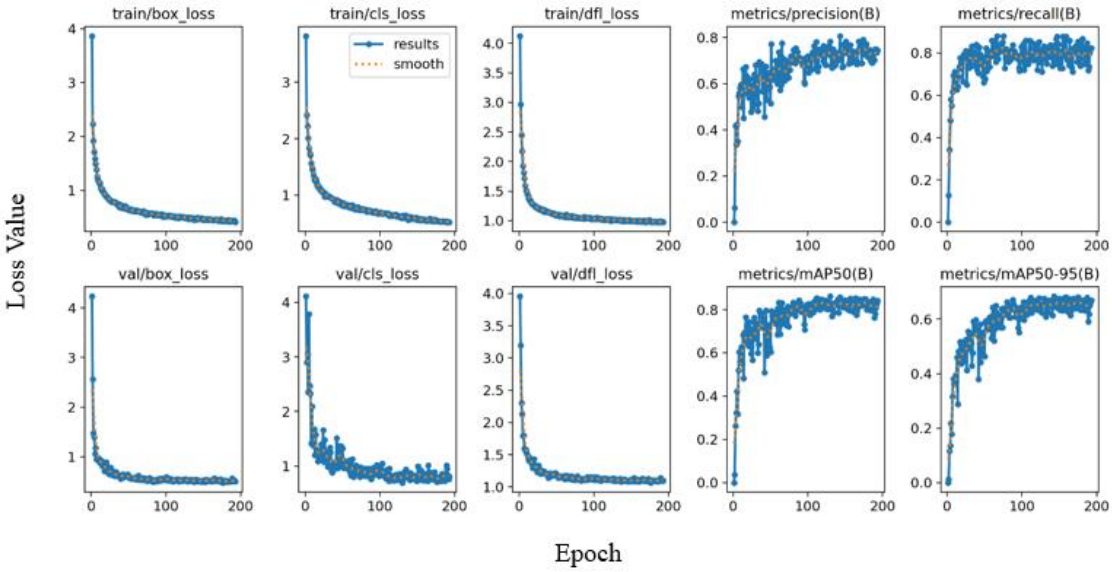


Figure F1. Light-Red Cherry Tomato Detection Model

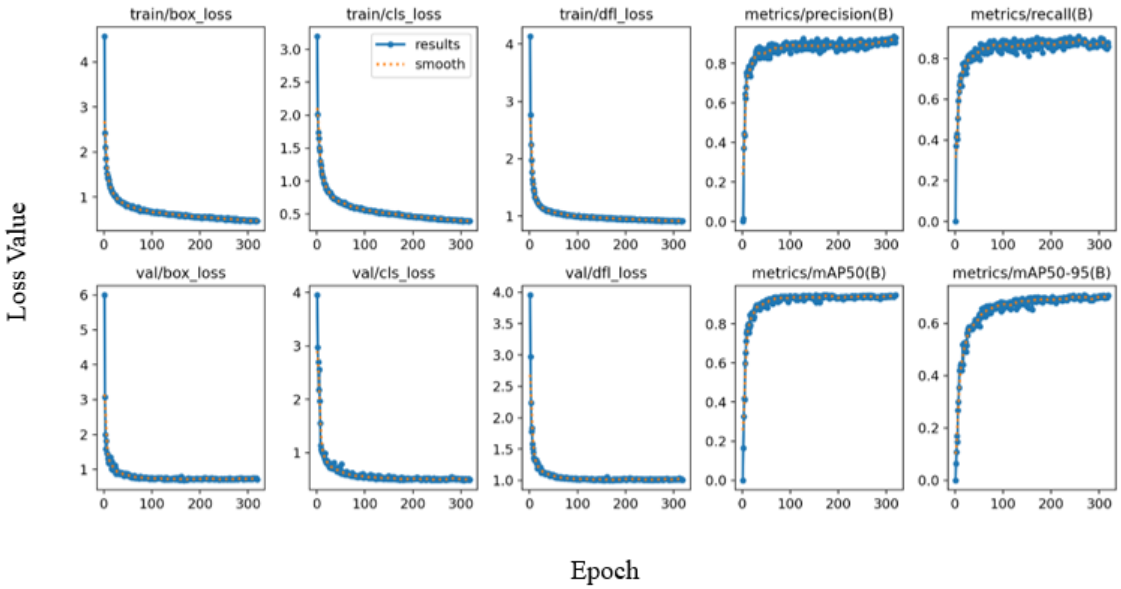


Figure F2. Red Cherry Tomato Detection Model

doi:10.6342/NTU202401594

Appendix G

Quantity of FN and FP across Different Confidence Thresholds

Table G1. False Negative

Video Sequences	VidSeq_1	VidSeq_2	VidSeq_3
0.4	75	194	79
0.5	51	184	51
0.6	49	200	50
0.7	68	230	53
0.8	182	301	93

Table G2. False Positive

Video Sequences	VidSeq_1	VidSeq_2	VidSeq_3
0.4	329	346	72
0.5	388	382	78
0.6	349	349	75
0.7	247	249	66
0.8	22	13	11

doi:10.6342/NTU202401594