Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.058409

ARTICLE

Enhancing Security in Distributed Drone-Based Litchi Fruit Recognition and Localization Systems

Liang Mao[1,2], Yue Li[1,2], Linlin Wang[1,*], Jie Li[1], Jiajun Tan[1], Yang Meng[1] and Cheng Xiong[1]

[1]Guangdong-Hong Kong-Macao Greater Bay Area Artificial Intelligence Application Technology Research Institute, Shenzhen Polytechnic University, Shenzhen, 518055, China

[2]School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, 114051, China

*Corresponding Author: Linlin Wang. Email: wanglinlin@szpu.edu.cn

Received: 11 September 2024 Accepted: 04 November 2024 Published: 17 February 2025

ABSTRACT

This paper introduces an advanced and efficient method for distributed drone-based fruit recognition and localization, tailored to satisfy the precision and security requirements of autonomous agricultural operations. Our method incorporates depth information to ensure precise localization and utilizes a streamlined detection network centered on the RepVGG module. This module replaces the traditional C2f module, enhancing detection performance while maintaining speed. To bolster the detection of small, distant fruits in complex settings, we integrate Selective Kernel Attention (SKAttention) and a specialized small-target detection layer. This adaptation allows the system to manage difficult conditions, such as variable lighting and obstructive foliage. To reinforce security, the tasks of recognition and localization are distributed among multiple drones, enhancing resilience against tampering and data manipulation. This distribution also optimizes resource allocation through collaborative processing. The model remains lightweight and is optimized for rapid and accurate detection, which is essential for real-time applications. Our proposed system, validated with a D435 depth camera, achieves a mean Average Precision (mAP) of 0.943 and a frame rate of 169 FPS, which represents a significant improvement over the baseline by 0.039 percentage points and 25 FPS, respectively. Additionally, the average localization error is reduced to 0.82 cm, highlighting the model's high precision. These enhancements render our system highly effective for secure, autonomous fruit-picking operations, effectively addressing significant performance and cybersecurity challenges in agriculture. This approach establishes a foundation for reliable, efficient, and secure distributed fruit-picking applications, facilitating the advancement of autonomous systems in contemporary agricultural practices.

KEYWORDS

Objective detection; deep learning; machine learning

1 Introduction

In recent years, the integration of robotics in agriculture has gained attention, with target detection methods based on image processing and machine learning becoming more prevalent in these environments. Early research applied methods such as local binary patterns and color histograms, combining these features using score fusion algorithms for fruit species recognition. Similarly, Yamamoto [1] developed a segmentation approach using traditional RGB cameras and machine learning techniques, while Zawbaa et al. [2] utilized shape and color features, along with Scale Invariant Feature Transform (SIFT), to build fruit species recognition systems. However, these methods struggled to handle the complexities of real-world environments, where lighting changes, occlusion, and overlapping fruits pose significant challenges to accurate recognition.

Deep learning methods, specifically convolutional neural networks (CNNs), have since advanced the field of fruit recognition, offering more robust and scalable solutions. For example, Zhang et al. [3] introduced an RGB-D depth camera to fuse depth and image data for accurate localization of tomato bunches, while Lin et al. [4] used RGB-D images and Euclidean clustering for guava detection and segmentation. Research has also focused on real-time localization in more complex environments, with deep learning models for navel oranges and citrus fruits. While deep learning has shown promise in these controlled scenarios, there are still significant challenges when it comes to handling small, densely packed objects like litchis, particularly in natural orchard environments where occlusion, lighting variability, and jitter affect performance.

The need for secure and efficient recognition and localization in such environments becomes even more critical when deployed in distributed drone networks. Drones, used for automated fruit picking, operate in decentralized systems, introducing vulnerabilities to data tampering, unauthorized access, and system disruptions. Ensuring the integrity of the data used for fruit recognition, as well as the communication between drones and central processing systems, is essential for reliable operation.

This paper builds on previous deep learning approaches and introduces a new method to enhance both performance and security in drone-based fruit recognition systems. The proposed solution leverages the YOLOv8 model, modified with the RepVGG network for lightweight processing to reduce computational complexity. Additionally, the model incorporates a small target detection layer and SKAttention module to enhance its ability to detect small objects such as litchis, even in complex environments. Furthermore, we address the security challenges by implementing a secure distributed framework that protects against data manipulation and ensures robust communication across drones. By combining these advancements with RGB-D depth camera technology, the system achieves precise localization while ensuring the reliability and security of the distributed architecture.

Experimental results demonstrate that the proposed method achieves high accuracy and performance in detecting litchi fruits in complex orchard environments, with improvements in detection speed and localization precision. Additionally, the integration of security protocols ensures that the system is resilient to cyber threats, making it suitable for large-scale, automated drone-based fruit harvesting operations.

The contribution of this paper can be summarized as follows:

- We propose an improved YOLOv8 model by integrating the RepVGG network structure for lightweight processing, Selective Kernel Attention for better feature extraction, and a small target detection layer to improve the recognition of small fruits, such as litchis, in complex environments.
- We introduce a secure framework for distributing the fruit recognition and localization process across multiple drones, ensuring data integrity and protection against tampering, which is crucial for reliable and safe operations in decentralized systems.
- Our system leverages RGB-D depth cameras to enhance the accuracy of fruit localization, achieving an average localization error of 0.82 cm, which meets the precision demands for autonomous drone operations in agricultural settings.

2 Related Work

2.1 *Objective Detection*

Objective detection is fundamental in computer vision, particularly for applications that involve identifying and localizing objects in diverse environments. This capability is crucial in agriculture, where precise detection of fruits can significantly improve automated tasks such as harvesting and yield monitoring [5,6]. Historically, traditional techniques like local binary patterns and color histograms were employed for fruit recognition, extracting texture and color features to classify various fruit types [7,8]. While these methods laid the groundwork, they often fell short in robustness under real-world conditions, challenged by issues such as varying lighting, occlusion, and overlapping fruits [9].

The advent of deep learning has been transformative in the field of objective detection. Specifically, Convolutional Neural Networks (CNNs) have revolutionized feature extraction and processing, facilitating the automatic learning of complex patterns from data [10]. The YOLO (You Only Look Once) series, including the advanced YOLOv8 model, exemplifies this progress by providing a real-time detection framework that efficiently predicts object boundaries and classifications directly from image data [11,12]. These models have achieved significant gains in speed and accuracy, rendering them ideal for dynamic and complex environments [13].

In agricultural settings, the integration of deep learning with depth sensing technologies has further enhanced detection capabilities. RGB-D cameras, capturing both color and depth information, offer a richer dataset that improves object localization accuracy [14,15]. This method has proven effective in projects aimed at detecting tomatoes, guavas, and passion fruits, where depth data helps resolve ambiguities caused by overlapping fruits and varying object sizes [16]. By merging depth information with deep learning models, precise 3D localization is achieved, enabling accurate detection even in cluttered and occluded environments [17,18].

Despite these advancements, challenges remain, especially in detecting small, densely packed objects. Factors such as lighting variations, shadows, and natural occlusions can impede detection performance. Current research is directed towards improving feature extraction techniques and enhancing model efficiency. Techniques like Selective Kernel Attention, which refine the model's focus and filter out irrelevant noise, show promise in overcoming these obstacles [19]. As these technologies evolve, they are expected to further refine objective detection systems, broadening their applicability to a range of real-world scenarios [20,21].

2.2 *Fruit Recognition*

Fruit recognition plays a pivotal role in agricultural automation, facilitating tasks such as harvesting, sorting, and quality control [22]. Initially, traditional image processing methods that relied on handcrafted features like color histograms, Local Binary Patterns (LBP), and Scale Invariant Feature Transform (SIFT) were predominant [23,24]. These methods, effective in controlled conditions, struggled in natural settings characterized by variable lighting, occlusion, and overlapping fruits. Techniques combining color and texture features, followed by classification with k-nearest neighbors or support vector machines, often faced limitations under complex conditions [25,26].

The adoption of deep learning has significantly advanced fruit recognition capabilities, particularly through Convolutional Neural Networks [27,28]. These models autonomously learn and extract relevant features from extensive datasets, adapting better to varying conditions. YOLO (You Only Look Once) models, known for real-time object detection, frame detection as a regression problem, simultaneously predicting bounding boxes and class probabilities [29,30]. YOLO's efficacy in agricultural contexts, such as apple detection in orchards, illustrates its capability to handle challenges like variable lighting and partial occlusions [31,32].

Enhancements in fruit recognition have also been achieved through the integration of depth information from RGB-D cameras. This multimodal approach, combining RGB images with depth data, significantly improves the ability to accurately recognize and localize fruits, even in cluttered and occluded environments [33]. These developments have bolstered fruit recognition systems, making them more robust and suitable for diverse and challenging agricultural settings [34,35].

Challenges persist, particularly in detecting small, densely packed fruits in varying environmental conditions. Fluctuating lighting, shadows, and foliage occlusions continue to impact the performance of recognition systems. Future research is likely to focus on enhancing the robustness and adaptability of deep learning models, improving network architectures, and employing sophisticated data augmentation techniques. Addressing these challenges will further advance the development of fully automated, efficient, and accurate fruit recognition systems, significantly benefiting agricultural operations [36,37].

3 Methodology

3.1 Fruit Identification Methods

The network structure of YOLOv8 is mainly composed of backbone network, neck network and head network. Because of the consideration of high real-time and accuracy in the process of litchi picking, the method in this paper is improved in terms of improving the detection efficiency of the algorithm, while improving the accuracy of the model. Specifically: 1) Use RepVGG module to replace the c2f module as the network model of YOLOv8 to realize the lightweight of the model. 2) Add the SKAttention attention mechanism to improve the lychee target detection accuracy of the network model of YOLOv8. 3) Add the small target detection layer to improve the model's ability of detecting the dense small lychee target. The network structure of the method in this paper is shown in Fig. 1.

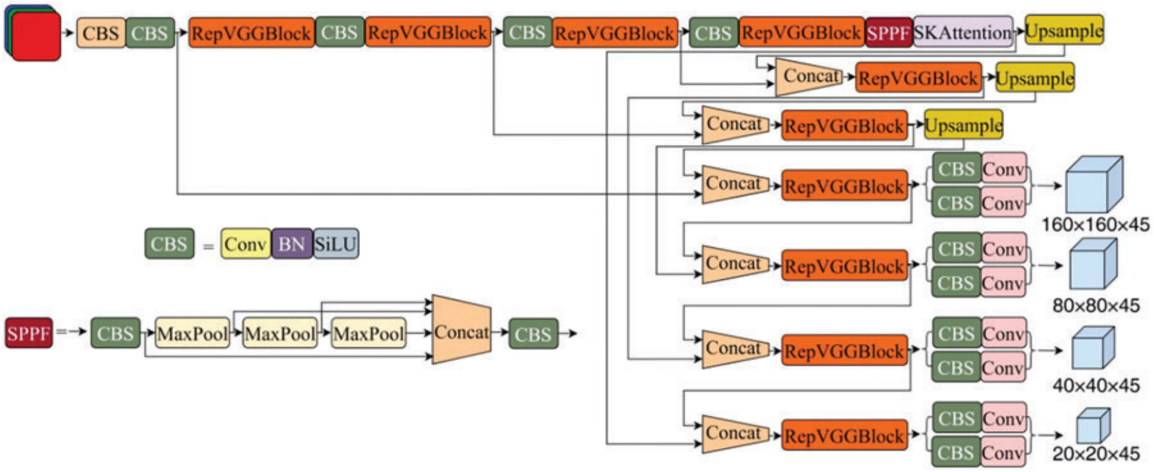


Figure 1: Network structure of our method

RepVGG was selected for this study due to its balance between simplicity, lightweight architecture, and high inference speed, which are critical for real-time applications like drone-based fruit detection. Unlike more complex models like EfficientNet and Transformer-based architectures, RepVGG uses a structural re-parameterization approach, allowing the model to switch between a multi-branch structure during training and a more efficient single-path structure during inference. This results in faster computations while maintaining accuracy, which is essential in dynamic environments where quick responses are required. While EfficientNet and Transformer-based models have demonstrated superior performance in handling highly complex and variable datasets, they often come with higher computational costs, which could slow down real-time detection on resource-constrained devices like drones. In future work, we plan to explore these advanced architectures to assess their potential benefits in terms of accuracy and robustness in more diverse environments. However, for the current study, RepVGG offers an optimal trade-off between speed, efficiency, and detection accuracy for real-time fruit-picking operations.

The RepVGG network uses a structural re-parameterization approach, which results in a substantial increase in both speed and accuracy. During training, a multi-branch model is used, where multiple branches generally increase the model's representational power, while inference is converted to a single-path model, which is faster, memory efficient and more flexible. As shown in Fig. 2, Fig. 2A represents the network structure used for RepVGG training, while the network structure of Fig. 2B is used for inference.

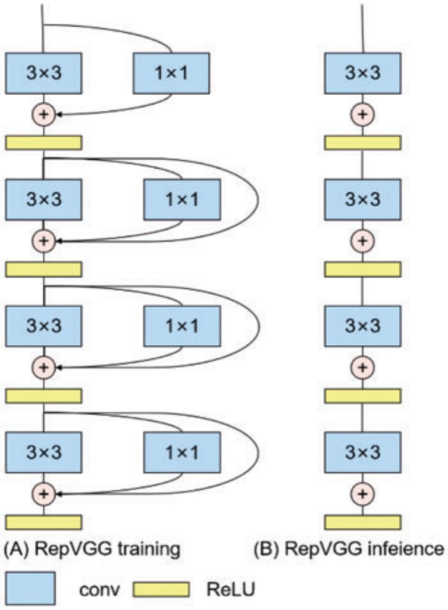


Figure 2: RepVGG network structure

RepVGG has three branches in each layer during training, which are IDENTIFY, 1 × 1, 3 × 3. When the model is trained, the output, for each layer, requires 3 parameter blocks, and for an n-layer network, 3n parameter blocks are required. Therefore, the method in this paper performs structural reparameterization, which makes the number of model parameters small when inference is performed. repVGG converts the 3-branch network equivalently and simplifies it into a single-branch network. The structural reparameterization is divided into three main steps: (1) fusing Conv2d and BN; (2) converting 1 × 1 convolution to 3 × 3 convolution; (3) multi-branch fusion, the process is shown in Fig. 3.

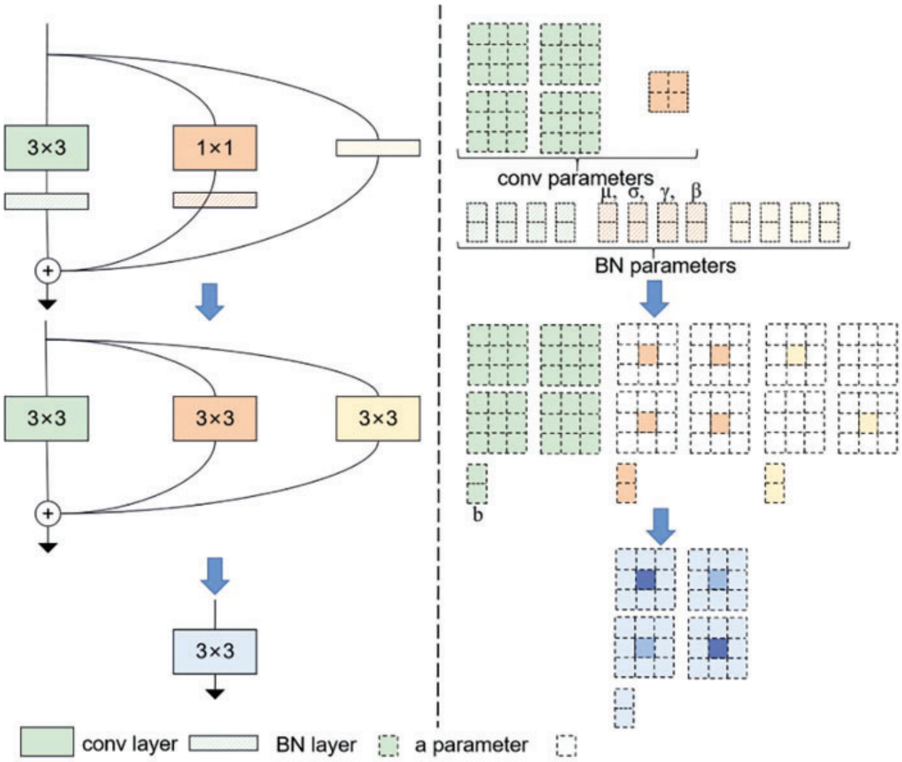


Figure 3: Structure reparameterization process

3.2 *Small Target Detection Layer*

The original YOLOv8 network model has a relatively large downsampling multiplier, and during the downsampling process with a trunk step of 2, the network model can obtain more semantic information, but a large amount of detail feature information is lost, and the lack of shallow network information is a problem [38]. Among them, the detail information contains mass features of small-sized objects, which may be ignored in the downsampling process. The deep feature map is difficult to learn the feature information of small targets. Therefore, the method in this paper adds a small target detection layer, which detects the shallower feature maps spliced with the deeper feature maps, so that the network model pays more attention to the detection of small targets of litchi fruits, reduces the leakage rate of small targets, and improves the detection accuracy of litchi fruits, as shown in Fig. 4.

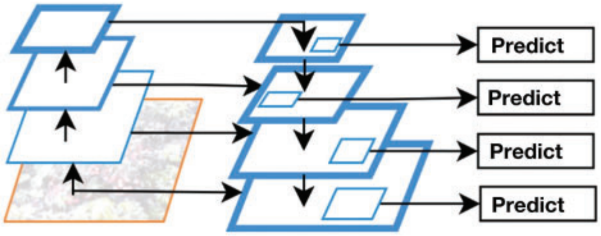


Figure 4: Small target detection layer

3.3 *Incorporating Selective Kernel Attention Attention Mechanisms*

SKAttention, called Selective Kernel Attention, is an attention mechanism that introduces different kernel sizes to capture multi-scale contextual information in convolutional neural networks (CNNs). In traditional CNNs, the sensory field size is fixed, limiting their ability to efficiently capture both local and global contextual information [39]. The SKAttention attention mechanism addresses this limitation by introducing multiple parallel convolutional branches, each using a different kernel size. These branches can capture information at different spatial scales, allowing the network model to have a better understanding of the input features. The key idea of SKAttention is to utilize channel attention across different kernel sizes. The attention mechanism learns the importance of each channel for each kernel size, allowing the network to selectively focus on the most informative kernel size. This adaptivity allows the model to dynamically adjust the receptive field and gather relevant information from different scales. By introducing SKAttention into the CNN architecture, the model is able to capture both the local details of fine-grained litchi fruits and the larger global context, thus improving the accuracy of litchi recognition, as shown in Fig. 5.

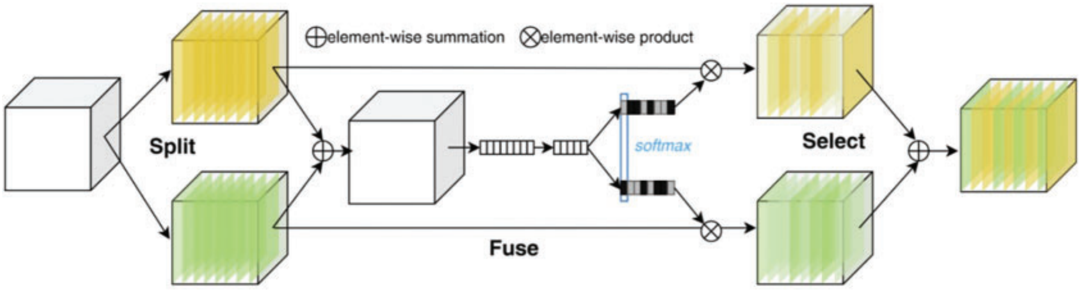


Figure 5: SKAttention structure

Notably, **Split** performs a complete convolution operation with different convolution kernel sizes on the input tensor X; **Fuse** performs element-wise summation on the above two outputs to obtain the output feature map U, followed by global average pooling with a fully connected layer to obtain the attentional information of the feature maps; Select: re-divides the compact vectors into two (the structure of the above figure) or Select: re-split the compact vector into two (in the above structure) or more (in more cases) feature vectors, and then perform the multiplication operation with the corresponding channels with the feature maps after the split, and finally sum them up to form the tensor for input to the next neuron.

The integration of SKAttention and the small target detection layer plays a crucial role in improving the model's performance. SKAttention enhances the model's ability to capture multi-scale contextual information by employing different kernel sizes, allowing it to focus on both fine-grained details and broader patterns. This mechanism significantly improves the detection of small, overlapping, or partially occluded fruits in dense environments. Meanwhile, the small target detection layer addresses the challenge of detecting small objects by fusing shallow and deep feature maps, which ensures that fine details are preserved throughout the network's downsampling process. Together, these components boost the model's precision and recall, particularly for small and hard to detect targets, while maintaining high computational efficiency.

4 Experiments

4.1 Settings

4.1.1 Collection of Data

The experimental data were gathered in June 2023 at the litchi garden on the campus of Shenzhen Vocational and Technical University, Nanshan District, Shenzhen, China. Mature and immature litchis were photographed from various angles using an Intel RealSense D435 camera under varying climatic conditions, including cloudy or sunny days. The collected images represent a range of real-world fruit growth conditions such as smooth light, backlight, sparse and dense fruit arrangements, and shaded or overlapping fruits, ensuring a representative sample set. A total of 7686 sets of color images were collected, consisting of 96,400 mature litchis and 27,213 immature litchis. The data were divided into training and test sets in an 8:2 ratio, with 6149 images for training and 1537 images for testing, manually labeled using Labelme. The lightweight YOLOv8 model used in this study has approximately 25.3 million parameters, significantly reduced from over 40 million parameters in the standard YOLOv8 model. An example of the data used in our research is given in Fig. 6.

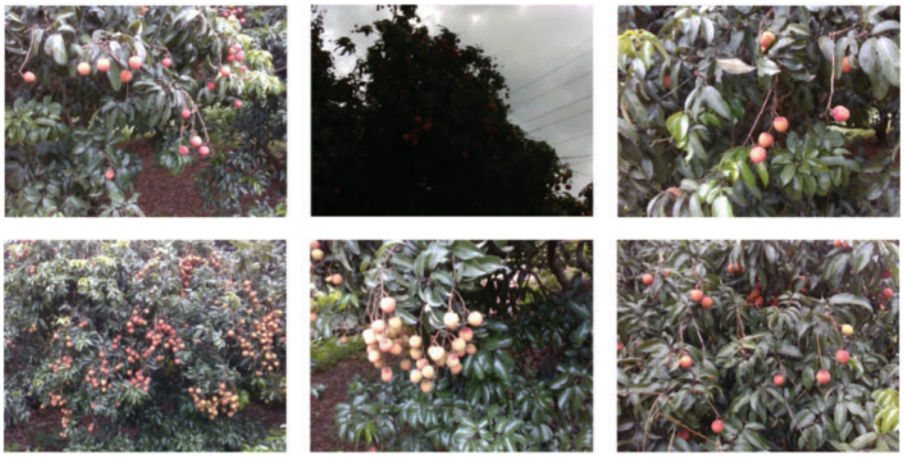


Figure 6: Examples of the collected data

Our dataset primarily focuses on litchi fruits, captured under various environmental conditions that challenge the detection system, such as varying lighting, occlusion by leaves, and varying fruit densities. However, the dataset is limited as it concentrates on a single fruit type in a specific agricultural setting. Future expansions of the dataset to include various fruit types and environmental conditions are planned to assess the model's applicability and robustness in broader agricultural scenarios.

4.1.2 Experimental Parameters

Training parameters were set as follows: batch size at 8, number of iterations at 400, and input image size at 640 × 640. Default values were used for other parameters. The experimental setup is detailed in Table 1. For testing, the batch size was set to 1, and the input image size remained at 640 × 640, optimizing the conditions for detailed, image-by-image evaluation of model performance.

Table 1: Experimental environment

Configure	Parameters
CPU	AMD EPYC 7742 64-Core Processor
RAM	128
GPU	A100-SXM4-40GB
Operating system	Windows 10
CUDA	CUDA Version: 11.2
Image processing language	Python 3.8
Deep learning framework	Pytorch 1.7

4.1.3 Evaluation Metrics

Evaluation metrics for object detection include precision, recall, mean average precision (mAP), and frames per second (FPS). Precision and recall are defined respectively by the equations:

$$P = \frac{TP}{TP + FP} \tag{1}$$

$$R = \frac{TP}{TP + FN} \tag{2}$$

mAP provides a comprehensive measure of model performance across different classes and is calculated as the mean of the area under the precision-recall curve. FPS is defined as:

$$FPS = \frac{1000}{\text{inference time (ms)}} \tag{3}$$

indicates the speed at which the model processes images, vital for real-time applications.

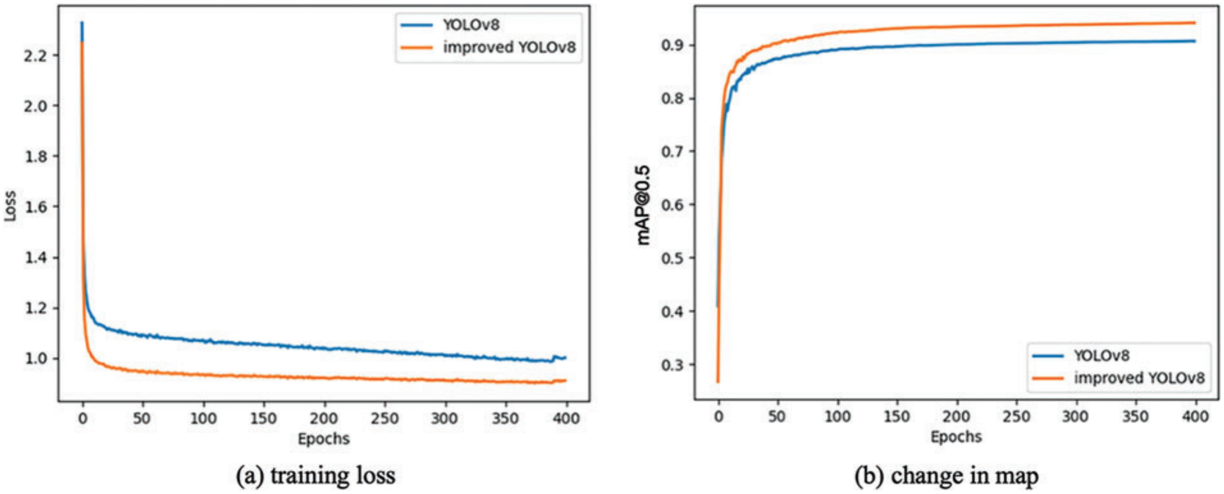
4.2 Results and Analysis

4.2.1 Model Training Results

Training results show that the modified YOLOv8 model demonstrates a faster convergence in loss reduction compared to the original model, stabilizing around 50 iterations, with an evident improvement in mAP over time, as shown in Fig. 7.

Detection results in various conditions confirm the effectiveness of the improvements. The model accurately identifies litchi fruits under different lighting and occlusion scenarios, crucial for autonomous picking robots, as shown in Fig. 8.

Backbone network comparisons reveal that RepVGG provides the best balance of accuracy, recall, mAP, and detection speed among the networks tested, as detailed in Table 2 and Fig. 9.



(a) training loss (b) change in map

Figure 7: Training results comparison

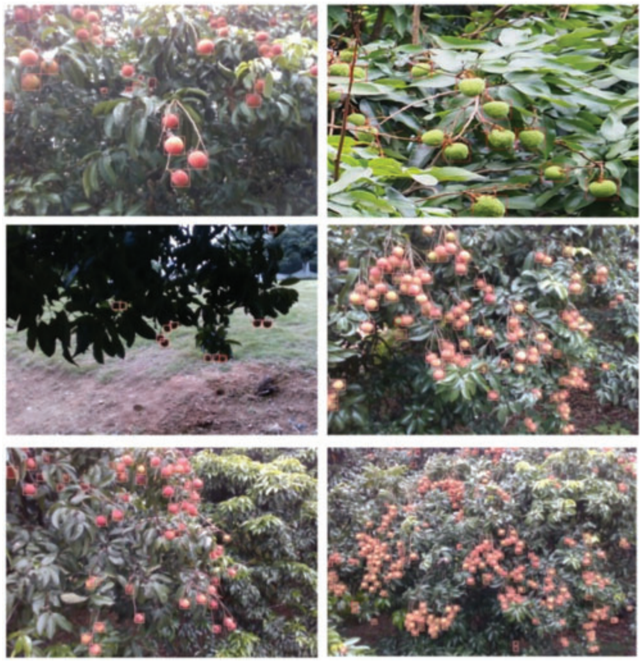


Figure 8: Detection results under various conditions

Table 2: Results of different backbone experiments

Id	Backbone network	mAP@0.5	Precious	Recall	FPS
1	YOLOv8	0.904	0.901	0.836	144
2	Shufflenet v2	0.863	0.883	0.787	144

(Continued)

Table 2 (continued)

Id	Backbone network	mAP@0.5	Precious	Recall	FPS
3	Mobilenet v3	0.864	0.886	0.786	117
4	Efficientnetv2	0.802	0.873	0.721	90
5	Mobile ViT	0.784	0.867	0.692	85
6	Mobilile One	0.896	0.896	0.823	62
7	PP-LCNet	0.852	0.876	0.769	169
8	RepVGG	0.906	0.901	0.841	181

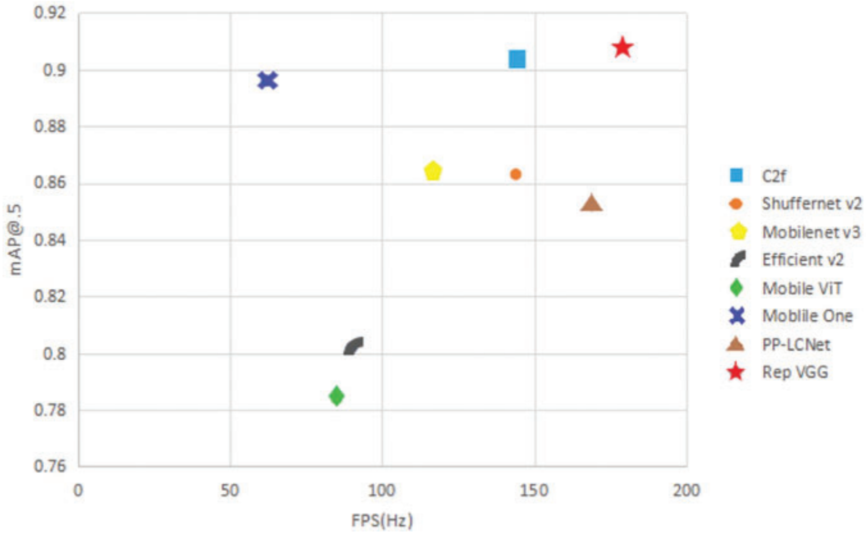


Figure 9: Comparison of experimental results for different backbone networks

These results underline the robustness and effectiveness of the YOLOv8 model modified with RepVGG and SKAttention, enhancing both the accuracy and operational efficiency, making it highly suitable for real-time applications in agricultural settings.

4.3 Discussion

From the results we can observe that, the proposed security framework in our system enhances protection by distributing the fruit recognition and localization processes across multiple drones in a decentralized manner, which reduces the risk of single points of failure and limits exposure to tampering or manipulation. Each drone operates independently, and communication between drones and the central system is encrypted using secure communication protocols, ensuring data integrity and preventing interception by unauthorized parties. Additionally, we incorporate blockchain-based logging, where each transaction and data transfer is recorded on a tamper-proof ledger.

In particular, the decentralized recognition framework, encryption, and blockchain-based logging work in synergy to provide robust security for the proposed system. In the decentralized framework, the fruit recognition and localization tasks are distributed across multiple drones, eliminating the reliance on a central system and reducing the risk of a single point of failure. This decentralized approach makes it more difficult for an attacker to compromise the entire system, as each drone operates independently while sharing secure data. Encryption is employed for all communication between the drones and the central processing system, ensuring that data transmissions are protected from interception or tampering during flight. Additionally, blockchain-based logging provides an immutable and verifiable record of all data interactions within the system. Each drone logs its activities and data exchanges on a blockchain, making it nearly impossible for any tampering to go unnoticed, as the blockchain ledger ensures transparency and traceability. Together, these security mechanisms create a highly robust system that is resistant to common threats such as data manipulation, unauthorized access, and system disruption.

5 Conclusion

This paper introduces a method for the rapid recognition and localization of litchi fruits in orchard environments, which significantly enhances the original YOLOv8 network model. By incorporating the lightweight RepVGG network and integrating both a small target detection layer and the SKAttention attention module, this approach achieves more effective and efficient processing. Depth information, crucial for precise localization, is gathered using a depth camera, with the system's effectiveness validated through a comprehensive dataset of litchi fruits. The method entails several improvements. Firstly, the adoption of the RepVGG network structure reduces the model's size and memory usage, facilitating faster processing suitable for real-time applications in orchards. The introduction of a small target detection layer and SKAttention module enhances the model's ability to concentrate on relevant features and resist noise interference, improving both the accuracy and the generalization capacity of litchi fruit detection. In comparative tests with the standard YOLOv8 model, the enhanced YOLOv8 model exhibited a 3.9% improvement in mean Average Precision (mAP) and a 25 FPS increase, indicating boosted performance in rapid recognition tasks. Additionally, these modifications have improved the model's compactness and recall metrics, decreased memory usage during training, and made the system more adaptable for implementation on fruit-picking robots. Stereo matching techniques are utilized to fuse images with depth data, obtaining accurate depth measurements of litchi fruits. Experimental results show that within a range of 313 cm, the maximum error in 3D localization by this method is 1.7 cm, with an average error of 0.82 cm. These findings confirm that the localization precision satisfies the operational requirements of litchi-picking robots.

Acknowledgement: The authors thank all individuals and organizations whose contributions and support were instrumental to the success of this research.

Funding Statement: The work was supported by Guangdong Province Rural Science and Technology Commissioner Project, Zen Tea Reliable Traceability and Intelligent Planting Key Technology Research and Development, Promotion and Application (KTP20210199), Special Project of Guangdong Provincial Education Department, Research on Abnormal Behavior Recognition Technology of Pregnant Sows Based on Graph Convolution (2021ZDZX1091), Guangdong Basic and Applied Basic Research Foundation under Grant 2023A1515110729, Shenzhen Science and Technology Program under Grant 20231128093642002 and the Research Foundation of Shenzhen Polytechnic University under Grant 6023312007K.

Author Contributions: The authors confirm contribution to the paper as follows: Liang Mao: Conceptualization, methodology development, and implementation of security framework. Yue Li: Data collection, analysis, and experiments, including data labeling. Linlin Wang: Supervision, funding

acquisition, and manuscript editing. Jie Li: Model optimization and design of RepVGG-based architecture. Jiajun Tan: Integration of SKAttention and small-target detection components. Yang Meng: Experimental validation and evaluation metric analysis. Cheng Xiong: Data visualization and final manuscript review. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the corresponding author, upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

[1] M. Yamamoto, “A segmentation method based on motion from image sequence and depth,” in *[1990] Proc. 10th Int. Conf. Pattern Recog.*, 1990, vol. 1, pp. 230–232. doi: 10.1109/ICPR.1990.118100.

[2] H. M. Zawbaa, M. Abbass, M. Hazman, and A. E. Hassenian, “Automatic fruit image recognition system based on shape and color features,” in *Adv. Mach. Learn. Technol. Appli.: Second Int. Conf., AMLTA 2014*, Cairo, Egypt, Springer, 2014, pp. 278–290.

[3] Y. Zhang and T. Funkhouser, “Deep depth completion of a single RGB-D image,” in *Proc. IEEE Conf. Comput. Visi. Pattern Recognit.*, 2018, pp. 175–185.

[4] G. Lin, Y. Tang, X. Zou, J. Xiong, and J. Li, “Guava detection and pose estimation using a low-cost RGB-D sensor in the field,” *Sensors*, vol. 19, no. 2, 2019, Art. no. 428. doi: 10.3390/s19020428.

[5] S. A. Nawaz, J. Li, U. A. Bhatti, M. U. Shoukat, and R. M. Ahmad, “AI-based object detection latest trends in remote sensing, multimedia and agriculture applications,” *Front. Plant Sci.*, vol. 13, 2022, Art. no. 1041514. doi: 10.3389/fpls.2022.1041514.

[6] Y. -Y. Zheng, J. -L. Kong, X. -B. Jin, X. -Y. Wang, T. -L. Su and M. Zuo, “CropDeep: The crop vision dataset for deep-learning-based classification and detection in precision agriculture,” *Sensors*, vol. 19, no. 5, 2019, Art. no. 1058. doi: 10.3390/s19051058.

[7] K. Lee, C. Lee, S. -A. Kim, and Y. -H. Kim, “Fast object detection based on color histograms and local binary patterns,” in *TENCON 2012 IEEE Region 10 Conf.*, Cebu, Philippines, IEEE, 2012, pp. 1–4.

[8] Y. Su, D. Tao, X. Li, and X. Gao, “Texture representation in aam using gabor wavelet and local binary patterns,” in *2009 IEEE Int. Conf. Syst. Man and Cyber.*, IEEE, 2009, pp. 3274–3279.

[9] Y. Dong *et al.*, “Benchmarking robustness of 3D object detection to common corruptions,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 1022–1032.

[10] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, “A review of yolo algorithm developments,” *Procedia Comput. Sci.*, vol. 199, no. 11, pp. 1066–1073, 2022. doi: 10.1016/j.procs.2022.01.135.

[11] Y. Tian, G. Yang, Z. Wang, H. Wang, E. Li and Z. Liang, “Apple detection during different growth stages in orchards using the improved YOLO-V3 model,” *Comput. Electron. Agric.*, vol. 157, pp. 417–426, 2019. doi: 10.1016/j.compag.2019.01.012.

[12] C. Liu, Y. Tao, J. Liang, K. Li, and Y. Chen, “Object detection based on YOLO network,” in *2018 IEEE 4th Inform. Technol. Mechat. Eng. Conf. (ITOEC)*, IEEE, 2018, pp. 799–803.

[13] T. Diwan, G. Anirudh, and J. V. Tembhurne, “Object detection using YOLO: Challenges, architectural successors, datasets and applications,” *Multimed. Tools Appl.*, vol. 82, no. 6, pp. 9243–9275, 2023. doi: 10.1007/s11042-022-13644-y.

[14] O. Wosner, G. Farjon, and A. Bar-Hillel, “Object detection in agricultural contexts: A multiple resolution benchmark and comparison to human,” *Comput. Electron. Agric.*, vol. 189, 2021, Art. no. 106404. doi: 10.1016/j.compag.2021.106404.

[15] J. S. Dvorak, M. L. Stone, and K. P. Self, "Object detection for agricultural and construction environments using an ultrasonic sensor," *J. Agric. Saf. Health*, vol. 22, no. 2, pp. 107–119, 2016. doi: 10.13031/jash.22.11260.

[16] W. Zhao, W. Yamada, T. Li, M. Digman, and T. Runge, "Augmenting crop detection for precision agriculture with deep visual transfer learning a case study of bale detection," *Remote Sens.*, vol. 13, no. 1, 2020, Art. no. 23. doi: 10.3390/rs13010023.

[17] J. -W. Chen, W. -J. Lin, H. -J. Cheng, C. -L. Hung, C. -Y. Lin and S. -P. Chen, "A smartphone-based application for scale pest detection using multiple-object detection methods," *Electronics*, vol. 10, no. 4, 2021, Art. no. 372. doi: 10.3390/electronics10040372.

[18] M. Kragh, R. N. Jørgensen, and H. Pedersen, "Object detection and terrain classification in agricultural fields using 3D lidar data," in *Int. Conf. Comput. Vis. Syst.*, Springer, 2015, pp. 188–197.

[19] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 510–519.

[20] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *Adv. Neural Inform. Process. Syst.*, 2013, vol. 26.

[21] P. Zhou, B. Ni, C. Geng, J. Hu, and Y. Xu, "Scale-transferrable object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 528–537.

[22] R. Khan and R. Debnath, "Multi class fruit classification using efficient object detection and recognition techniques," *Int. J. Image, Graph. Signal Process.*, vol. 11, no. 8, pp. 1–18, 2019. doi: 10.5815/ijigsp.2019.08.01.

[23] T. Roddick, A. Kendall, and R. Cipolla, "Orthographic feature transform for monocular 3D object detection," 2018, *arXiv:1811.08188*.

[24] A. R. Jiménez, A. K. Jain, R. Ceres, and J. L. Pons, "Automatic fruit recognition: A survey and new results using range/attenuation images," *Pattern Recognit.*, vol. 32, no. 10, pp. 1719–1736, 1999. doi: 10.1016/S0031-3203(98)00170-8.

[25] S. Jana, S. Basak, and R. Parekh, "Automatic fruit recognition from natural images using color and texture features," in *2017 Dev. Integ. Circ. (DevIC)*, IEEE, 2017, pp. 620–624.

[26] S. Chaivivatrakul and M. N. Dailey, "Texture-based fruit detection," *Precis. Agric.*, vol. 15, no. 6, pp. 662–683, 2014. doi: 10.1007/s11119-014-9361-x.

[27] L. Hou, Q. Wu, Q. Sun, H. Yang, and P. Li, "Fruit recognition based on convolution neural network," in *2016 12th Int. Conf. Natural Comput., Fuzzy Syst. Know. Disc. (ICNC-FSKD)*, IEEE, 2016, pp. 18–22.

[28] H. S. Gill, O. I. Khalaf, Y. Alotaibi, S. Alghamdi, and F. Alassery, "Multi-model CNN-RNN-LSTM based fruit recognition and classification," *Intell. Autom. Soft Comput.*, vol. 33, no. 1, pp. 638–650, 2022.

[29] Y. J. Wu, Y. Yang, X. Wang, J. Cui, and X. Y. Li, "Fig fruit recognition method based on YOLO v4 deep learning," in *2021 18th Int. Conf. Elect. Eng./Elect. Comput. Telecommun. Inform. Technol. (ECTI-CON)*, IEEE, 2021, pp. 303–306.

[30] Z. Wang, L. Jin, S. Wang, and H. Xu, "Apple stem/calyx real-time recognition using YOLO-v5 algorithm for fruit automatic loading system," *Postharvest Biol. Technol.*, vol. 185, no. 2, 2022, Art. no. 111808. doi: 10.1016/j.postharvbio.2021.111808.

[31] N. Mamdouh and A. Khattab, "YOLO-based deep learning framework for olive fruit fly detection and counting," *IEEE Access*, vol. 9, pp. 84 252–84 262, 2021. doi: 10.1109/ACCESS.2021.3088075.

[32] Q. An, K. Wang, Z. Li, C. Song, X. Tang and J. Song, "Real-time monitoring method of strawberry fruit growth state based on yolo improved model," *IEEE Access*, vol. 10, pp. 124 363–124 372, 2022. doi: 10.1109/ACCESS.2022.3220234.

[33] F. Xiao, H. Wang, Y. Xu, and R. Zhang, "Fruit detection and recognition based on deep learning for automatic harvesting: An overview and review," *Agronomy*, vol. 13, no. 6, 2023, Art. no. 1625. doi: 10.3390/agronomy13061625.

[34] M. H. Junos, A. S. Mohd Khairuddin, S. Thannirmalai, and M. Dahari, "An optimized YOLO-based object detection model for crop harvesting system," *IET Image Process.*, vol. 15, no. 9, pp. 2112–2125, 2021. doi: 10.1049/ipr2.12181.

[35] Y. Wang, G. Yan, Q. Meng, T. Yao, J. Han and B. Zhang, "DSE-YOLO: Detail semantics enhancement YOLO for multi-stage strawberry detection," *Comput. Electron. Agric.*, vol. 198, no. 3, 2022, Art. no. 107057. doi: 10.1016/j.compag.2022.107057.

[36] Y. Bai, J. Yu, S. Yang, and J. Ning, "An improved YOLO algorithm for detecting flowers and fruits on strawberry seedlings," *Biosyst. Eng.*, vol. 237, no. 13, pp. 1–12, 2024. doi: 10.1016/j.biosystemseng.2023.11.008.

[37] D. Wu, S. Lv, M. Jiang, and H. Song, "Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments," *Comput. Electron. Agric.*, vol. 178, no. 4, 2020, Art. no. 105742. doi: 10.1016/j.compag.2020.105742.

[38] E. Cetinic, T. Lipic, and S. Grgic, "Fine-tuning convolutional neural networks for fine art classification," *Expert. Syst. Appl.*, vol. 114, no. 6, pp. 107–118, 2018. doi: 10.1016/j.eswa.2018.07.026.

[39] M. Ma, C. Xia, C. Xie, X. Chen, and J. Li, "Boosting broader receptive fields for salient object detection," *IEEE Trans. Image Process.*, vol. 32, pp. 1026–1038, 2023. doi: 10.1109/TIP.2022.3232209.