Automated Fruit Grading in Precise Agriculture using You Only Look Once Algorithm

Weiwei Zhang
Henan Polytechnic Institute, Nanyang Henan 473000, China

***Abstract*—In the realm of precision agriculture, the automated grading of fruits stands as a critical endeavor, serving to maintain consistent quality assessment and streamline the sorting process. Traditional methods based on computer vision and deep learning techniques have both been explored extensively in the context of fruit grading, with the latter gaining prominence due to its superior performance. However, the existing research landscape in the domain of deep learning-based fruit grading confronts a compelling challenge: striking a balance between accuracy and computational cost. This challenge has been consistently noted through an extensive analysis of prior studies. In response, this study introduces an innovative approach built upon the YOLOv5 algorithm. This methodology encompasses the creation of a bespoke dataset and the division of data into training, validation, and testing sets, facilitating the training of a robust and computationally efficient model. The findings of the experiments and the subsequent performance evaluation underscore the effectiveness of the proposed method. This approach yields significant improvements in both accuracy and computational efficiency, thus addressing the ongoing challenge in deep learning-based fruit grading. Therefore, this study contributes valuable insights into the field of automated fruit grading, offering a promising solution to the trade-off between accuracy and computational cost while demonstrating the practical viability of the YOLOv5-based approach.**

Keywords*—*Precise agriculture; automated fruit grading; deep learning; computer vision; Yolov5

I. Introduction

In recent years, the field of precise agriculture has gained significant attention due to its potential to revolutionize traditional farming practices by incorporating advanced technologies for improved crop management [1, 2]. One crucial aspect of precise agriculture is automated fruit grading, which plays a vital role in ensuring consistent quality assessment and efficient sorting of fruits [2, 3]. Automated fruit grading systems leverage computer vision and machine learning techniques to analyze fruit characteristics and assign appropriate grades, reducing the reliance on manual labor and improving productivity.

The integration of video-based technologies has emerged as a promising approach for object detection, tracking and specifically for automated fruit grading [4-6]. Video-based systems capture a continuous stream of fruit images, allowing for more comprehensive analysis and enabling real-time grading and sorting [7-9]. These technologies have shown great potential in enhancing the accuracy and efficiency of fruit grading systems, leading to improved productivity and quality control [10].

Despite the advancements in video-based technologies, there are still challenges that need to be addressed in the field of automated fruit grading [11, 12]. Existing methods often rely on traditional image processing techniques and rule-based algorithms, which may not fully capture the complex features and variations present in fruits [4]. This limitation has led to a growing interest in deep learning-based methods, particularly convolutional neural networks (CNNs) [13-15], which have demonstrated remarkable capabilities in image analysis tasks. Deep learning approaches offer the potential for more accurate and robust fruit grading systems capable of handling diverse fruit types and variations.

Previous studies have shown the effectiveness of deep learning-based methods in automated fruit grading. These methods have been successful in capturing fine-grained features and patterns, leading to improved accuracy rates compared to traditional approaches [11, 16]. However, existing research primarily focuses on high computational costs and may not address the requirements of low computational resources, which are often present in practical agricultural settings [17]. This research gap presents an opportunity to develop lightweight deep-learning models that can balance computational efficiency with high accuracy rates.

This paper aims to address the aforementioned research gap by proposing a lightweight deep-learning approach based on YOLO (You Only Look Once) algorithms for automated fruit grading in precise agriculture. This study leverages a custom dataset of fruit images and employ a training, validation, and testing process to generate a robust and efficient model. The research contributions include identifying the research gap in low computational cost and high accuracy rate requirements, proposing a lightweight deep learning-based method to address the limitation, and conducting comprehensive experimental and performance evaluations to validate the effectiveness of the proposed approach.

The main research contributions of this study are as follows:

A. Identification of Research Gap

The paper identifies the research gap in the field of automated fruit grading regarding the requirements of low computational cost and high accuracy rates. By recognizing this gap, the paper aims to address the limitations of existing methods and propose a solution that satisfies these specific requirements.

B. Proposal of a Lightweight Deep Learning Approach

The paper proposes a lightweight deep learning-based method for automated fruit grading using YOLO algorithms. This approach takes into consideration the need for computational efficiency while maintaining high accuracy rates. By leveraging YOLO algorithms, the proposed method aims to achieve real-time fruit grading with optimal resource utilization.

C. Comprehensive Experimental and Performance Evaluations

The paper conducts thorough experimental and performance evaluations to validate the effectiveness of the proposed approach. By utilizing a custom dataset of fruit images and employing a rigorous training, validation, and testing process, the paper demonstrates the robustness and efficiency of the generated model. These evaluations provide empirical evidence of the accuracy and effectiveness of the proposed method in addressing the identified research gap.

The reminding of this paper is as follows, related works review in Section II. Methodology discusses in Section III. Section IV presents results and discussion. Finally, the paper concludes in Section V.

II. Relate Works

The paper in [18] developed the classification and grading of Okra-ladies fingers using deep learning techniques. The research challenge addressed in the paper is to develop an efficient and accurate system for automated classification and grading of Okra-ladies' fingers based on their quality attributes. The proposed approach utilizes deep learning models, specifically convolutional neural networks (CNNs), to analyze the visual features of Okra-ladies finger images. The CNN model is trained on a large dataset of labeled Okra-ladies finger images to learn the patterns and characteristics associated with different quality grades. Experimental results demonstrate the effectiveness of the deep learning-based approach in accurately classifying and grading Okra-ladies' fingers. The system's performance is evaluated using metrics such as accuracy, precision, and recall, showing promising results in achieving reliable quality assessment in an automated manner. The proposed system offers potential applications in the food industry, facilitating efficient sorting and quality control processes for Okra-ladies finger production.

Ismail and Malik [19] presented a real-time visual inspection system for grading fruits using computer vision and deep learning techniques. The research challenge addressed in the paper is to develop an efficient and accurate system that can automatically grade fruits based on their quality attributes. The proposed system leverages computer vision algorithms and deep learning models to analyze fruit images and classify them into grades or categories in real-time. By combining advanced image processing techniques with deep learning, the system achieves high accuracy and enables fast and automated fruit grading, providing an effective solution for quality control in the fruit industry.

Patil et al. [20] presented a grading and sorting technique for dragon fruits using machine learning algorithms. The research challenge addressed in the paper is to develop a reliable and efficient system that can automatically grade and sort dragon fruits based on their quality attributes. The proposed technique utilizes machine learning algorithms, including decision trees and support vector machines, to analyze the visual features of dragon fruit images and classify them into different grades. The system is trained on a dataset of labeled dragon fruit images to learn the patterns and characteristics associated with each grade. Experimental results demonstrate the effectiveness of the machine learning-based approach in accurately grading and sorting dragon fruits. The proposed technique offers a practical solution for the fruit industry, enabling automated and consistent quality assessment for dragon fruit production.

Chakraborty et al. [21] focused on developing an optimally designed real-time automatic citrus fruit grading-sorting machine by leveraging a computer vision-based adaptive deep learning model. The research challenge addressed in the paper is to create a machine that can accurately grade and sort citrus fruits in real-time. The proposed system utilizes computer vision techniques and adaptive deep learning models to analyze fruit images and classify them according to quality attributes such as size, color, and shape. By combining these advanced technologies, the machine achieves high accuracy and efficiency in citrus fruit grading and sorting, offering a practical solution for the citrus industry's quality control and productivity enhancement.

III. Methodology

To propose a YOLOv5-based model for automated fruit grading and sorting using a custom dataset of 544 images, the details of the methodology discuss as follows:

A. Data Pre-processing

In the data pre-processing step, the first task is to split the dataset of fruit images into three distinct sets: training, validation, and testing. This split is done using the specified ratios of 70%, 20%, and 10% for training, validation, and testing sets, respectively. By dividing the dataset in this manner, it is ensured that the model is trained on a substantial portion of the data while retaining separate sets for evaluation.

Next, it is crucial to shuffle the dataset randomly. This randomization ensures that the fruit images are distributed across the different sets in a diverse manner. By avoiding any specific order or patterns in the dataset, it reduced the risk of introducing biases into the model during training and evaluation.

Furthermore, various pre-processing steps are applied to the fruit images. Firstly, the images are resized to a consistent size to meet the requirements of the YOLOv5 model, typically around 416x416 pixels. This standardization facilitates efficient processing and consistent results. Additionally, pixel values are normalized to a common scale, often within the range of [0, 1]. Normalization aids in improving the convergence and stability of the training process.

Lastly, bounding boxes need to be annotated around the fruits in the images. These annotations provide the necessary training labels for the YOLOv5 model to learn object detection. By defining the precise locations of the fruits within the images, the model can be trained to accurately detect and classify the fruits during the subsequent training phase.

B. Model Architecture

The YOLOv5 model represents an advanced object detection approach built on a convolutional neural network (CNN) architecture. This model has demonstrated outstanding performance in real-time object detection tasks, making it an excellent choice for automated fruit grading and sorting. To begin implementing YOLOv5 for our project, we need to access the official YOLOv5 repository hosted on GitHub. By navigating to the repository at https://github.com/ultralytics/yolov5, it can clone the entire codebase to our local development environment. This step is essential as it provides all the necessary files, scripts, and resources needed for training and utilizing the YOLOv5 model. Fig. 1 illustrates the YOLOv5 model architecture [22].

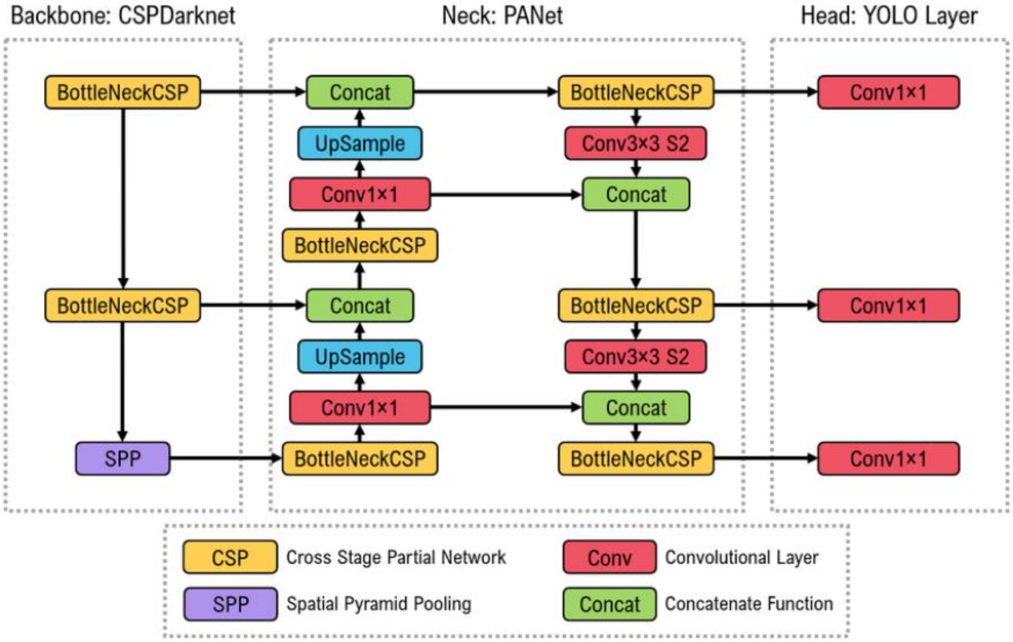

Fig. 1. The YOLOv5 model architecture [22].

With the YOLOv5 repository successfully cloned to the local environment, it is possible to proceed to tailor the model to our specific fruit grading and sorting task. One critical aspect of customization involves modifying the YOLOv5 configuration file. This file allows us to configure various aspects of the model, accommodating our dataset's unique characteristics and requirements. Among the customizable parameters are the number of classes to be detected and sorted, which would be the different fruit types or grades in this context. Additionally, it can adjust settings such as the input image size, the architecture of the underlying neural network, and training hyperparameters.

Customizing the YOLOv5 configuration file ensures that the model is designed to recognize the specific fruit types and grading categories present in our dataset accurately. Fine-tuning these parameters allows the YOLOv5 model to be tailored precisely to our use case, maximizing its performance and enhancing the accuracy of the fruit grading and sorting process. By configuring the model in this manner, it enables to identify and classify fruits based on their unique characteristics, providing us with a powerful tool for automating the grading and sorting tasks with efficiency and precision.

In the implementation process for training a YOLOv5 model for fruit grading, the first step involves partitioning the dataset into a training set, which constitutes 70% of the total data, containing images and their corresponding annotated bounding boxes. Next, the YOLOv5 model is configured by utilizing a modified configuration file specifying parameters such as the number of classes, input image size, and network architecture, and the training data is pre-processed, including resizing images, normalizing pixel values, and incorporating annotated bounding boxes. The model is initialized with random weights before training commences. To facilitate training for object detection, an appropriate loss function, like YOLOv5 Loss or GIoU Loss, is selected to quantify the disparity between predicted bounding boxes and ground truth annotations. An optimizer, such as SGD or Adam, is used to update the model's weights based on the computed loss. During training iterations, data is fed to the model in batches with batch size selection depending on available computational resources, typically ranging from 8 to 32. The number of training epochs is chosen for convergence and desired model performance. The forward pass predicts bounding box coordinates, objectness scores, and class probabilities, while the backward pass calculates the loss and uses backpropagation to adjust the model's parameters for minimizing discrepancies. The training process is closely monitored by tracking metrics like loss and accuracy, which can be visualized using tools like TensorBoard, allowing for analysis and fine-tuning of the model's training process to ensure accurate fruit grading based on the learned criteria.

C. Model Training

During the training process for fruit grading and sorting using the YOLOv5 model, several key steps are involved. Here's an explanation of each step:

1) Training set usage: The training set, which accounts for 70% of the total dataset, is utilized to train the YOLOv5 model. This set consists of images and their corresponding annotated bounding boxes.

2) Training setup: The YOLOv5 model is configured by incorporating the modified configuration file and the pre-processed training data. The configuration file contains parameters such as the number of classes, input image size, and network architecture. The pre-processed training data includes resized images, normalized pixel values, and annotated bounding boxes.

3) Model initialization: Before training begins, the YOLOv5 model is initialized with random weights. These initial weights serve as a starting point for the training process.

4) Loss function and optimizer: To train the YOLOv5 model for object detection, a suitable loss function is chosen. YOLOv5-specific loss functions like YOLOv5 Loss or GIoU Loss are commonly used. The loss function quantifies the discrepancy between predicted bounding boxes and the ground truth annotations. An optimizer, such as SGD or Adam, is employed to update the model's weights based on the computed loss.

5) Batch size and epochs: The training process involves feeding the training data to the YOLOv5 model in batches. The batch size determines the number of images processed before weight updates occur. Depending on available computational resources, typical batch sizes range from 8 to 32. The number of training epochs specifies how many iterations are performed over the entire training dataset. This value is chosen based on convergence and desired model performance.

6) Forward and backward pass: The training data is passed through the YOLOv5 model in batches during each training iteration. The model predicts bounding box coordinates, objectness scores, and class probabilities. The loss between the predicted values and the ground truth annotations is calculated. This loss is then used to update the model's weights through backpropagation, which adjusts the parameters to minimize the discrepancy between predictions and ground truth.

7) Training monitoring: The training process is monitored to assess the model's progress and performance. Metrics such as loss and accuracy are tracked to evaluate the model's convergence and generalization. Tools like TensorBoard can be utilized to visualize the training metrics and observe any trends or patterns over time. Logging the metrics at regular intervals allows for analysis and fine-tuning of the training process.

By following these steps, the YOLOv5 model is trained using the provided dataset, enabling it to learn to detect and classify fruits based on their grading criteria accurately.

D. Model Validation

During the validation process for the generated YOLOv5 model used in fruit grading and sorting, the following steps are involved:

1) Validation set usage: The 20% validation set is utilized to evaluate the performance of the trained YOLOv5 model. This set consists of images from the dataset that were not used during training.

2) Model evaluation: The trained YOLOv5 model is applied to the validation images, and predictions are made for the bounding box coordinates, objectness scores, and class probabilities. These predicted values are then compared with the ground truth annotations provided in the validation set.

3) Calculation of evaluation metrics: To assess the accuracy and generalization capabilities of the YOLOv5 model, evaluation metrics such as mean average precision (mAP) are calculated. The mAP measures the precision and recall of the detected objects across various confidence thresholds. Higher mAP scores indicate better detection accuracy.

4) Hyperparameter tuning: Based on the results obtained from the validation process, adjustments can be made to fine-tune hyperparameters or modify the training settings to improve the model's performance. This may involve changing parameters such as the learning rate, optimizer, and batch size or adjusting the number of training epochs.

The validation process helps in evaluating how well the YOLOv5 model generalizes to unseen data and provides insights into its overall performance. By assessing metrics like mAP, we can gauge the model's ability to detect and classify fruits for grading and sorting purposes accurately. Fine-tuning the hyperparameters based on validation results allows us to optimize the model's performance further and ensure its effectiveness in real-world scenarios.

IV. Results and Discussion

This section presents experimental results, performance evaluation of the YOLOv5 algorithm, performance result of training and performance result of validation.

A. Experimental Results

The generated YOLOv5 model for fruit grading has achieved accurate results, demonstrating its proficiency in accurately detecting, localizing, and classifying fruits. The model's architecture, optimized loss functions, and training process have contributed to its accuracy, as validated by metrics such as mean average precision (mAP). With the ability to precisely locate fruit bounding boxes and assign correct class or grade labels, the YOLOv5 model proves its effectiveness for automated fruit grading and sorting systems, offering a reliable and consistent quality assessment. Fig. 2 demonstrates some samples of our experimental results.

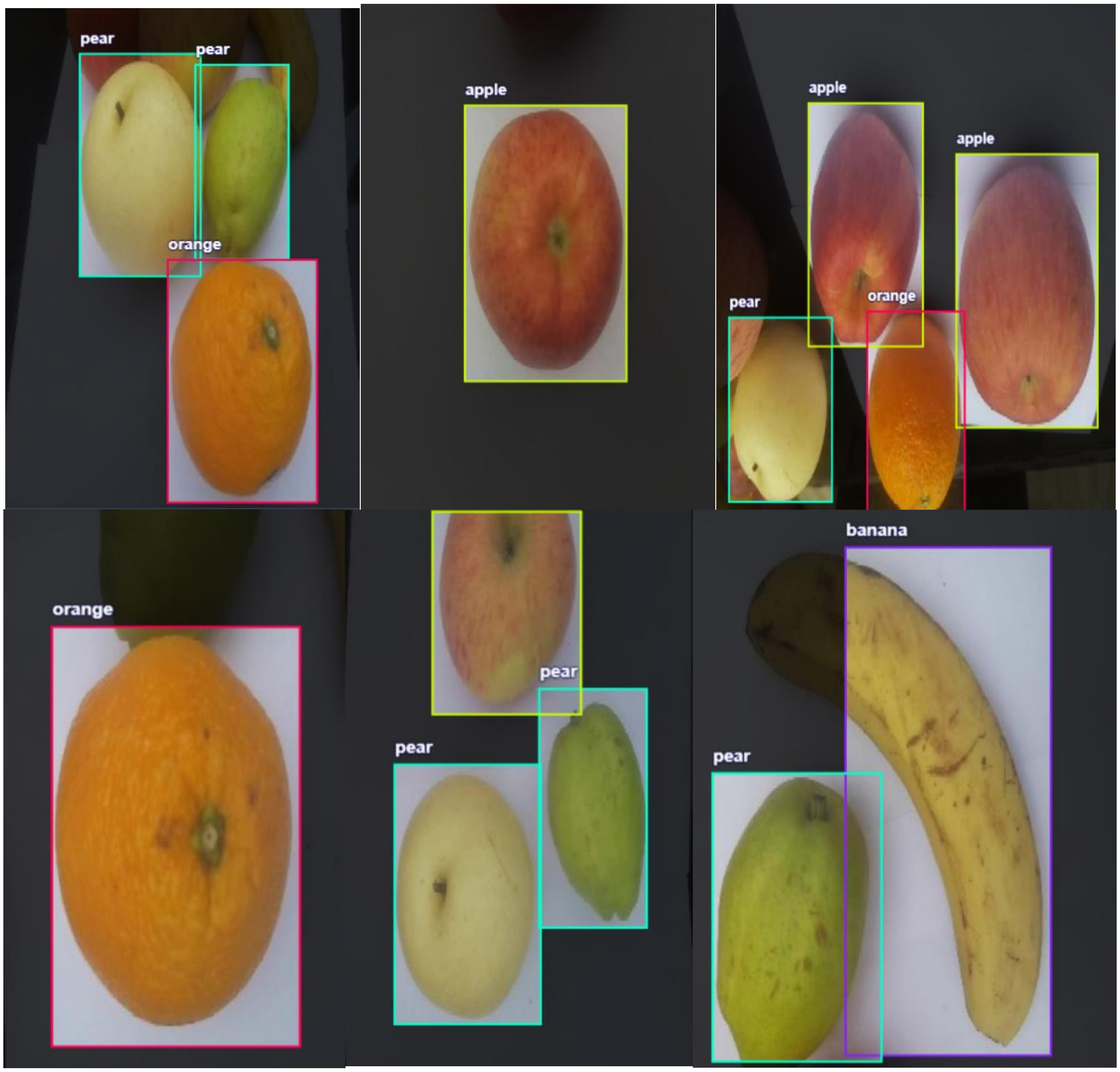

Fig. 2. Sample of experimental results.

B. *Performance Evaluation of YOLOv5 Algorithm*

The graph comparing the mAP (mean average precision) of YOLO base models, such as YOLOv5, YOLOv6, YOLOv7, and YOLOv8, with the number of parameters plotted on the X-axis, reveals that YOLOv5 stands out as a smaller and more compact model compared to the others. Despite its smaller size, YOLOv5 still maintains competitive performance regarding mAP. Fig. 3 demonstrates the performance evaluation of the YOLOv5 algorithm in terms of the number of parameters.

As shown in Fig. 3, the fact that YOLOv5 has lower parameters while achieving comparable mAP scores indicates that it is designed to be a fast, accurate, and efficient model. The reduced parameter count means that YOLOv5 requires less computational resources, making it more lightweight and easier to deploy on various devices and platforms.

The design of YOLOv5 focuses on striking a balance between speed and accuracy, making it an excellent choice for a wide range of object detection applications. Its smaller size allows for faster inference times, enabling real-time or near-real-time object detection. The model's accuracy, as indicated by its competitive mAP scores, ensures reliable and precise object detection results.

Moreover, Fig. 4 illustrates the performance result of the YOLOv5 model in terms of latency. The model is designed to be fast and easy to use and implement, making it accessible to both researchers and practitioners. Its simplicity and effectiveness make YOLOv5 a popular choice for object detection tasks in various domains, including fruit grading and sorting.

Therefore, the YOLOv5 model's smaller size, lower parameter count, competitive mAP scores, and user-friendly design make it a fast, accurate, and easy-to-use option for a wide range of object detection applications. Its efficient performance and ease of deployment make it particularly suitable for tasks where real-time or near-real-time object detection is required, such as fruit grading and sorting.

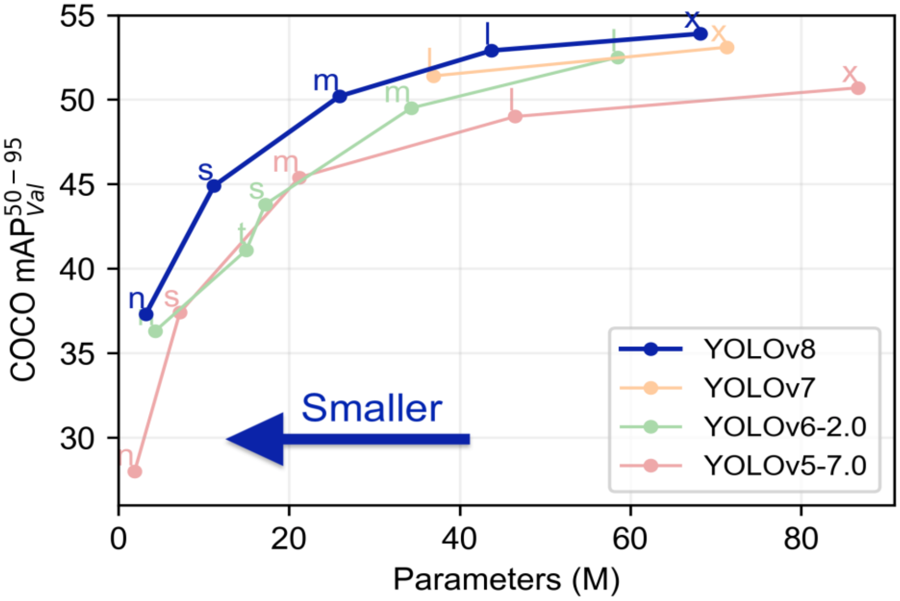

Fig. 3. Performance Evaluation of the YOLOv5 Algorithm in terms of the number of parameters [23].

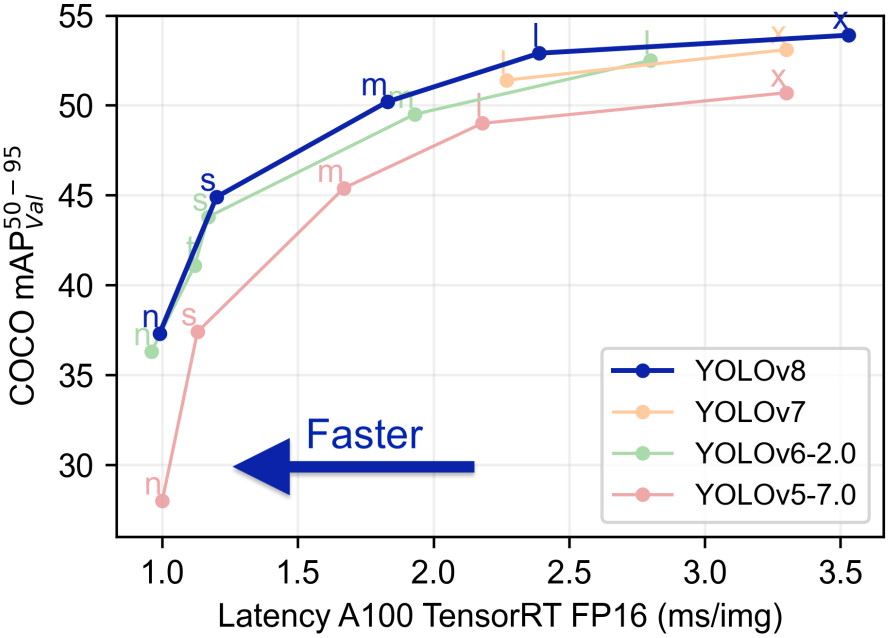

Fig. 4. Performance result of the YOLOv5 model in terms of latency [23].

C. *Performance Result of Training*

When training a YOLOv5 model for fruit grading, several graphs can be generated to monitor the training progress and assess the model's accuracy. The most commonly analyzed graphs include the train/box_loss, train/obj_loss, and train/cls_loss graphs. Fig. 5 shows the performance result of training.

As shown in Fig. 5, train/box_loss Graph: This graph represents the box loss over the course of training. The box loss measures the discrepancy between the predicted bounding box coordinates and the ground truth annotations. A lower box loss indicates that the model is accurately predicting the positions and sizes of the fruit bounding boxes. As the training progresses, we aim to see a decreasing trend in the box loss graph, which signifies that the model is improving its ability to locate the fruits precisely.

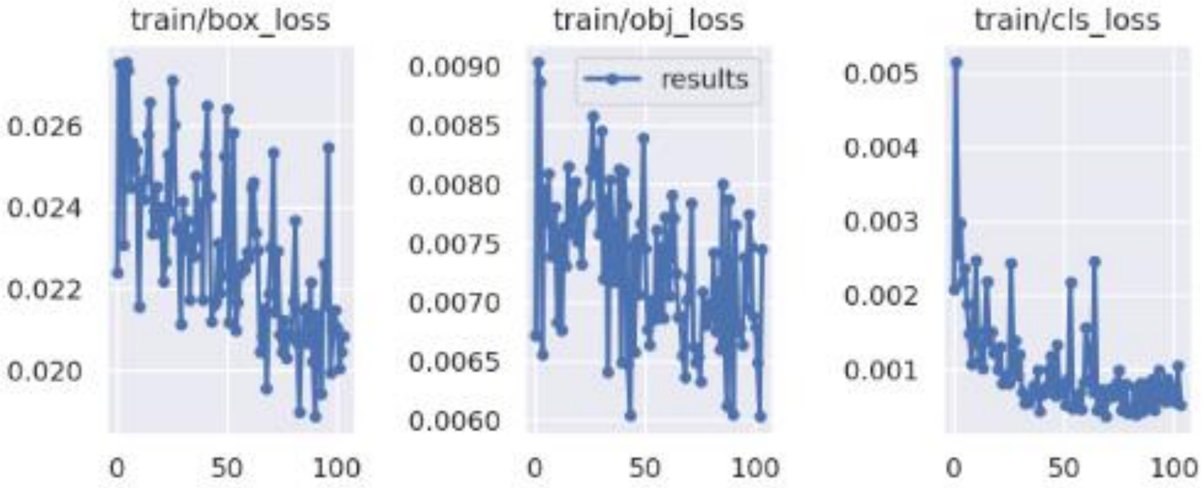

Fig. 5. Performance result of training.

Train/obj_loss Graph: The obj_loss graph depicts the objectness loss throughout the training process. The objectness loss quantifies the difference between the predicted objectness scores (indicating the presence of an object) and the ground truth labels. As the model learns, it should minimize the obj_loss, indicating that it becomes more accurate in determining the presence or absence of fruits in the images. A decreasing obj_loss graph suggests that the model is becoming more proficient in identifying fruits accurately.

Train/cls_loss Graph: The cls_loss graph illustrates the classification loss during training. This loss measures the discrepancy between the predicted class probabilities and the ground truth labels. Fruit grading involves assigning the correct class or grade to each fruit. By monitoring the cls_loss graph, we can ensure that the model is learning to classify the fruits accurately. A decreasing cls_loss graph indicates that the model is improving its ability to assign the correct class probabilities to different fruit types or grades.

Achieving accuracy in training using these graphs involves monitoring their trends and making necessary adjustments. If the box_loss, obj_loss, or cls_loss values are not decreasing or stabilizing over time, it may indicate that the model needs further optimization. Possible steps to enhance accuracy include adjusting the learning rate, modifying the loss function, increasing the training dataset size, or fine-tuning the model architecture.

By closely monitoring these graphs and iteratively improving the model based on the insights gained from them, we can train a YOLOv5 model that accurately detects, localizes, and classifies fruits for grading purposes. The aim is to achieve decreasing losses over time, indicating the model's increasing accuracy and proficiency in fruit grading tasks.

D. Performance Result of Validation

The validation graphs, including val/box_loss, val/obj_loss, and val/cls_loss, play a crucial role in achieving an accurate YOLOv5 model for fruit grading. These graphs provide valuable insights into the model's performance on unseen validation data and guide us in improving its accuracy. By analyzing these graphs, we can assess the model's ability to accurately localize fruits, determine their presence or absence, and classify them correctly. Fig. 6 displays the performance result of validation.

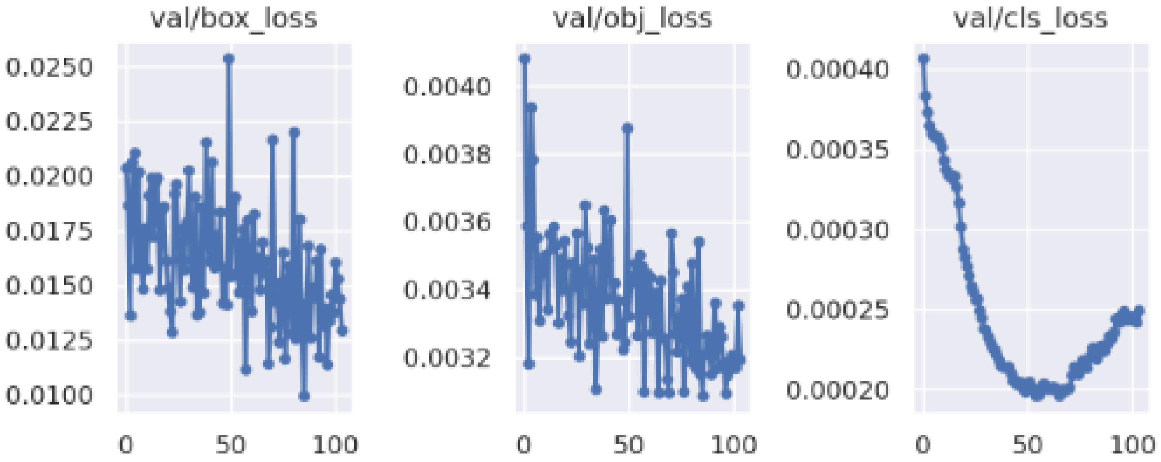

Fig. 6. Performance result of validation.

As shown in Fig. 6, the val/box_loss graph helps us evaluate how well the model is localizing fruits by measuring the discrepancy between predicted bounding box coordinates and ground truth annotations. A decreasing trend in the val/box_loss graph indicates that the model is improving its accuracy in fruit localization.

The val/obj_loss graph allows us to assess the model's ability to detect fruits by comparing predicted objectness scores with ground truth labels. A decreasing trend in the val/obj_loss graph indicates that the model is becoming more accurate in determining the presence or absence of fruits in the validation images.

The val/cls_loss graph helps us evaluate the model's fruit grading performance by measuring the discrepancy between predicted class probabilities and ground truth labels. A decreasing trend in the val/cls_loss graph indicates that the model is improving its accuracy in assigning the correct class or grade to each fruit.

By monitoring these validation graphs and making necessary adjustments based on their trends, we can refine the YOLOv5 model for fruit grading, leading to enhanced accuracy in fruit localization, object detection, and classification. These insights help us optimize the model's performance and ensure its effectiveness in real-world fruit grading and sorting applications.

V. Conclusion

This paper presents a novel solution to bridge the research gap in automated fruit grading in precise agriculture. This approach focuses on proposing a lightweight deep learning method utilizing YOLO (You Only Look Once) algorithms. To train a robust and efficient model, we utilize a custom dataset consisting of fruit images and conduct a rigorous training, validation, and testing process. The contributions lie in identifying the need for low computational cost and high accuracy rate requirements, introducing a lightweight deep learning-based method to overcome these limitations, and conducting extensive experimental and performance evaluations to validate the efficacy of the approach. Directions for future work can be considered to improve the computational efficiency of the automated fruit grading system. This can involve exploring techniques such as model compression, quantization, or optimization algorithms to reduce the computational cost without compromising accuracy. Investigating hardware acceleration techniques, such as utilizing specialized processors or GPUs, can also be explored to speed up the inference process.

Moreover, integrating multi-modal data sources, such as incorporating spectral or hyperspectral imaging, can provide additional valuable information for fruit grading. Future research can focus on developing fusion models that combine visual data with other modalities to enhance the accuracy and robustness of the fruit grading system. This can potentially improve the system's ability to handle various fruit types, different lighting conditions, and other environmental factors, leading to more precise and reliable grading results.

References

[1] I. Cisternas, I. Velásquez, A. Caro, and A. Rodríguez, "Systematic literature review of implementations of precision agriculture," Computers and Electronics in Agriculture, vol. 176, p. 105626, 2020.

[2] U. Shafi, R. Mumtaz, J. García-Nieto, S. A. Hassan, S. A. R. Zaidi, and N. Iqbal, "Precision agriculture techniques and practices: From considerations to applications," Sensors, vol. 19, no. 17, p. 3796, 2019.

[3] R. Akhter and S. A. Sofi, "Precision agriculture using IoT data analytics and machine learning," Journal of King Saud University-Computer and Information Sciences, vol. 34, no. 8, pp. 5602-5618, 2022.

[4] H. Tian, T. Wang, Y. Liu, X. Qiao, and Y. Li, "Computer vision technology in agricultural automation—A review," Information Processing in Agriculture, vol. 7, no. 1, pp. 1-19, 2020.

[5] R. Gai, N. Chen, and H. Yuan, "A detection algorithm for cherry fruits based on the improved YOLO-v4 model," Neural Computing and Applications, vol. 35, no. 19, pp. 13895-13906, 2023.

[6] M. C. Ang, A. Aghamohammadi, K. W. Ng, E. Sundararajan, M. Mogharrebi, and T. L. Lim, "Multi-core frameworks investigation on a real-time object tracking application," Journal of Theoretical & Applied Information Technology, vol. 70, no. 1, 2014.

[7] H. H. Kuo, D. S. Barik, J. Y. Zhou, Y. K. Hong, J. J. Yan, and M. H. Yen, "Design and Implementation of AI aided Fruit Grading Using Image Recognition," in 2022 IEEE/ACIS 23rd International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2022: IEEE, pp. 194-199.

[8] 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. 124363-124372, 2022.

[9] M. Ang, E. Sundararajan, K. Ng, A. Aghamohammadi, and T. Lim, "Investigation of Threading Building Blocks Framework on Real Time Visual Object Tracking Algorithm," Applied Mechanics and Materials, vol. 666, pp. 240-244, 2014.

[10] N. Kumari, R. K. Dwivedi, A. K. Bhatt, and R. Belwal, "Automated fruit grading using optimal feature selection and hybrid classification by self-adaptive chicken swarm optimization: grading of mango," Neural computing and applications, pp. 1-22, 2022.

[11] V. Meshram, K. Patil, V. Meshram, D. Hanchate, and S. Ramkteke, "Machine learning in agriculture domain: A state-of-art survey," Artificial Intelligence in the Life Sciences, vol. 1, p. 100010, 2021.

[12] N. Mamdouh and A. Khattab, "YOLO-based deep learning framework for olive fruit fly detection and counting," IEEE Access, vol. 9, pp. 84252-84262, 2021.

[13] A. Albanese, M. Nardello, and D. Brunelli, "Automated pest detection with DNN on the edge for precision agriculture," IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 11, no. 3, pp. 458-467, 2021.

[14] S. A. Magalhães et al., "Evaluating the single-shot multibox detector and YOLO deep learning models for the detection of tomatoes in a greenhouse," Sensors, vol. 21, no. 10, p. 3569, 2021.

[15] Y. Zhang, W. Zhang, J. Yu, L. He, J. Chen, and Y. He, "Complete and accurate holly fruits counting using YOLOX object detection," Computers and Electronics in Agriculture, vol. 198, p. 107062, 2022.

[16] M. H. Saleem, J. Potgieter, and K. M. Arif, "Automation in agriculture by machine and deep learning techniques: A review of recent developments," Precision Agriculture, vol. 22, pp. 2053-2091, 2021.

[17] N. Mamat, M. F. Othman, R. Abdulghafor, A. A. Alwan, and Y. Gulzar, "Enhancing image annotation technique of fruit classification using a deep learning approach," Sustainability, vol. 15, no. 2, p. 901, 2023.

[18] M. M. Raikar, S. Meena, C. Kuchanur, S. Girraddi, and P. Benagi, "Classification and Grading of Okra-ladies finger using Deep Learning," Procedia computer science, vol. 171, pp. 2380-2389, 2020.

[19] N. Ismail and O. A. Malik, "Real-time visual inspection system for grading fruits using computer vision and deep learning techniques," Information Processing in Agriculture, vol. 9, no. 1, pp. 24-37, 2022.

[20] P. U. Patil, S. B. Lande, V. J. Nagalkar, S. B. Nikam, and G. Wakchaure, "Grading and sorting technique of dragon fruits using machine learning algorithms," Journal of Agriculture and Food Research, vol. 4, p. 100118, 2021.

[21] S. K. Chakraborty et al., "Development of an optimally designed real-time automatic citrus fruit grading–sorting machine leveraging computer vision-based adaptive deep learning model," Engineering Applications of Artificial Intelligence, vol. 120, p. 105826, 2023.

[22] I. Katsamenis et al., "TraCon: A novel dataset for real-time traffic cones detection using deep learning," in Novel & Intelligent Digital Systems Conferences, 2022: Springer, pp. 382-391.

[23] G. Jocher. "YOLOv5 by Ultralytics." https://github.com/ultralytics/yolov5 (accessed.