Contents lists available at Science-Gate

International Journal of Advanced and Applied Sciences

Journal homepage: http://www.science-gate.com/IJAAS.html

Efficient batch size detection of apple fruit in a plantation environment

Wahyu Pebrianto [1], Ahmad Hoirul Basori [2,*], Hendra Yufit Riskiawan [1], Andi Besse Firdausiah Mansur [2], Taufiq Rizaldi [1], Nouf Atiahallah Alghanmi [2], Alanoud Subahi [2], Hermawan Arief Putranto [1], Hanadi Alkhudhayr [2], Arwa Mashat [2], Yogiswara Yogiswara [1]

[1]*Information Technology Department, Politeknik Negeri Jember, Jember, Indonesia*
[2]*Faculty of Computing and Information Technology in Rabigh, King Abdulaziz University, Rabigh 21911, Saudi Arabia*

ARTICLE INFO

Article history:
Received 16 May 2024
Received in revised form
6 September 2024
Accepted 8 September 2024

Keywords:
Deep learning
Object recognition
Apple farming
Batch size
Orchard environments

ABSTRACT

There is growing interest in using deep learning for object recognition in robots to enhance the efficiency of apple farming. While deep learning-based object detection has shown promising results in various visual tasks, more research is needed to accurately recognize apples in orchard environments. During the training phase, it is important to determine the optimal values of hyperparameters. This research aims to develop a deep learning model, YOLOv7, to reliably identify apples in orchards, using four different batch size values for training. The MinneApple dataset, trained with these batch sizes, serves as our reference model. To assess the model's ability to work in different situations, we evaluate it using test data with varying input scales. Our results show that the optimal batch size for detecting apples in orchards is 16, achieving a mean average precision (mAP) of 50%. Furthermore, our findings suggest that increasing the batch size does not improve the efficiency of apple detection in orchard environments.

© 2024 The Authors. Published by IASE. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

1. Introduction

Over the course of the past several years, there has been a significant trend toward the implementation of object identification technologies in conjunction with robots for the purpose of automating fruit farming activities (Xiao et al., 2023). For example, monitoring the health of the fruit (Kaplun et al., 2024), monitoring the quality of the fruit (Apostolopoulos et al., 2023), and making automatic harvesting decisions (Onishi et al., 2019; Yoshida et al., 2022). The presence of different lighting conditions, varying image quality, and complex backgrounds makes it challenging to develop a system that can automatically detect fruit in images or videos taken directly in an orchard environment. Traditional machine learning methods for object detection, such as Histogram of Oriented Gradients (HOG) (Sun et al., 2023), Viola-Jones Detector (Huang et al., 2019), and Scale Invariant Feature Transform (SIFT) (Lin et al., 2021), need further adjustments to overcome these challenges. This is because these methods rely on hand-crafted features to achieve the desired results. Utilizing deep learning-based object detection algorithms (LeCun et al., 2015), which have demonstrated promising performance in a variety of visual tasks (Chen et al., 2022; Voulodimos et al., 2018), particularly in fruit farming (Xiao et al., 2023), is currently a prominent option that is being used to address the inadequacies of standard machine learning. Apples are one of the most extensively consumed and produced fruits all over the world because of the ease with which they can be grown and harvested (Arnold and Gramza-Michalowska, 2023). Apples are widely regarded as an important agricultural crop due to their high nutritional value. In this study, we aim to identify different types of apples. The paper is divided into several sections. The first section covers the background of the research, followed by a review of related work in Section 2. Section 3 explains the research methodology, and Section 4 presents the study's results. Finally, Section 5 provides the conclusions and offers recommendations for future research.

2. Related works

2.1. Object detection in various apple farming

Currently, the attention of practitioners and researchers in the field of deep learning-based object

* Corresponding Author.
Email Address: abasori@kau.edu.sa (A. H. Basori)
https://doi.org/10.21833/ijaas.2024.09.017
Corresponding author's ORCID profile:
https://orcid.org/0000-0001-9684-490X
2313-626X/© 2024 The Authors. Published by IASE.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

detection on various apple farming problems is increasing, and much has been done to improve the efficiency of apple farming processes in complex environments (Chen et al., 2021; Kuznetsova et al., 2020; Wang and He, 2022; Xuan et al., 2020), which on average adopt YOLO as one-stage-detector method (Ge et al., 2021; Lin et al., 2020; Liu et al., 2016; Redmon et al., 2016; Redmon and Farhadi, 2017; 2018; Wang et al., 2021), which generally involves regression in the detection process, instead of using the two-stage-detector method (Girshick, 2015; Girshick et al., 2014; Ren et al., 2017), which adopt with region proposal technique, that tends to require larger computational cost in the training process and slower in the detection process (Chen et al., 2022).

YOLOX-Tiny (Ge et al., 2021) is adopted by Shufflenetv2-YOLOX (Ji et al., 2022), which is a lightweight backbone network consisting of Shufflenetv2 (Ma et al., 2018) with the inclusion of Convolutional Block Attention Module (CBAM) (Woo et al., 2018) for the purpose of Apple identification in orchard situations. This was done by Ji et al. (2022). When compared to YOLOX-Tiny, this research achieved an increase in Average Precision (AP) of 6.24% during the test phase. During the training phase, the research obtained an Average Precision (AP) of 96.76% with a speed of 65 Frames Per Second (FPS).

Additionally, this research demonstrates that YOLOv5-s (Pebrianto et al., 2024), Efficientdet-d0 (Tan et al., 2020), YOLOv4-Tiny, and Mobilenet-YOLOv4-Lite (Bochkovskiy et al., 2020) are incapable of achieving the same level of detection and speed as this research. YOLOv5-PRE is a product that was proposed by Sun et al. (2022). Its primary objective is to enhance the speed and accuracy of YOLOv5 when it comes to apple-detecting tasks in orchard environments. The lightweight structure of ShuffleNet (Zhang et al., 2018) and GhostNet (Han et al., 2020) was employed to reduce the size of the model. CA (Coordinate Attention) (Hou et al., 2021) and CBAM (Woo et al., 2018) were utilized to improve the accuracy of detection. According to the findings of this study, the YOLOv5-PRE model is both more accurate and faster than the YOLOv5s model. Lin et al. (2021) focused on enhancing the YOLOv4 model by utilizing data augmentation. They also replaced the Cross Stage Partial Darknet53 (CSPDarknet53) as a backbone network (Bochkovskiy et al., 2020) with EfficientNet (Tan and Le, 2019). Additionally, they added a convolutional layer (Conv2D) to the final three outputs to reduce the computational complexity. The findings of this study demonstrate that the YOLOv4 model may achieve superior performance on test results compared to YOLOv3 (Pebrianto et al., 2022; Redmon and Farhadi, 2018), YOLOv4 and Faster Region Convolutional Neural Network (Faster R-CNN) (Ren et al., 2017) with Residual Network (ResNet) (He et al., 2016) respectively. Unfortunately, a number of studies on the current Apple detection task (Chen et al., 2021; Kuznetsova et al., 2020; Wang and He, 2022; Xuan et al., 2020; Ji et al., 2022; Sun et al., 2022; Wu et al., 2021; Zhao et al., 2023). Typically, there is a tendency to prioritize modifying the model architecture to enhance performance rather than giving attention to the precise selection of the batch size hyperparameter. However, it is important to note that the batch size hyperparameter directly impacts the performance of optimization techniques like Stochastic Gradient Descent (SGD) (Qian and Klabjan, 2020) and its variants (Abdulkadirov et al., 2023) while searching for the most effective parameters and reducing loss values throughout the training of the model.

2.2. Batch size in various visual tasks and research problem

Several studies have investigated how to choose the right batch size for various visual tasks. Kandel and Castelli (2020) examined the effect of batch size on the performance of the VGG-16 model. Their findings showed that increasing the batch size does not necessarily lead to better accuracy. This is because the learning rate and optimization strategy also influence the model's accuracy. Additionally, other studies have shown that larger batch sizes can negatively affect the model's ability to generalize (LeCun et al., 2012; Keskar et al., 2016). Other studies emphasize the importance of analyzing batch size during the training of deep learning models. For example, Sato and Iiduka (2023) explored the training of Generative Adversarial Networks (GANs) and highlighted how batch size can affect the number of steps needed for training. Moreover, batch normalization techniques, which are key in many image classification models, are sensitive to batch size. These techniques may perform poorly when the batch size is too small (Brock et al., 2021), leading to underfitting and resulting in less effective models (Yong et al., 2020). This is different from what was focused on by Goyal et al. (2017) and You et al. (2019), which showed that choosing a high batch size during the training process can minimize the amount of time required during model training. Ahmad et al. (2024) and Stapor et al. (2022) similarly demonstrated that increasing batch size can potentially enhance accuracy while also reducing the time required for the training process. A significant factor that must be taken into consideration is the selection of the batch size hyperparameter, as demonstrated by the findings of the research that was presented earlier. A further complicated matter is the fact that the vast amount of data, which is then followed by the large model parameters, can be exceedingly difficult to compute. Stapor et al. (2022) also necessitated the right modification of batch values. Therefore, it is of the utmost importance to do further batch size study on specific visual tasks, particularly those tasks that involve the identification of apples in surroundings that are reminiscent of orchards. Performing batch size analysis on the YOLOv7 model is the objective that we have set for ourselves within the context of

this inquiry. On the other hand, to the best of our knowledge, there has not been any research done to test the batch size value that concentrates on YOLOv7 (Wang et al., 2023a) for the purpose of detecting apples directly around the orchard.

2.3. Objective of the research

To develop appropriate batch hyperparameters for the apple detection task in an orchard setting, it is important to address the issue discussed earlier. Our main goal is to thoroughly analyze the batch size hyperparameter in YOLOv7, a state-of-the-art object detection method, rather than focusing on modifying the model's architecture to improve performance. We selected YOLOv7 because it is the most advanced and effective object detection technology available, outperforming its previous versions. By examining the batch size, we aim to identify the optimal hyperparameter for apple detection directly in the orchard environment. The specific contributions of this research are as follows:

- We propose the YOLOv7 method to address direct apple detection in orchard environments with different batch sizes.
- We show the appropriate batch size for training the YOLOv7 in the case of apple detection in an orchard environment.
- We show that different input scales can influence the generalization level of the YOLOv7 model.

3. Material and methods

This section provides a comprehensive description of the materials and procedures employed in this investigation. Initially, we will discuss the distinguishing features of YOLOv-7, followed by an examination of the batch size and dataset employed in this study.

3.1. YOLOv7

The basis of the YOLO method (Redmon et al., 2016) in the detection process is to map the input pixels in the image into an S x S grid. Each grid cell is tasked with predicting the B bounding box and the confidence score, which is explained by the following Eqs. 1-3,

$$\text{Confidence} = \text{P}_\text{r}(\text{Object})*\text{IoU}\left(\frac{\text{truth}}{\text{predict}}\right) \quad (1)$$

$$\text{Class probability} = \text{Pr}(\text{Class}_\text{i}|\text{Object}) \quad (2)$$

$$\text{Pr}(\text{Class}_\text{i}|\text{Object}) * \text{P}_\text{r}(\text{Object}) * \text{IoU}\left(\frac{\text{truth}}{\text{predict}}\right) = \text{Pr}(\text{Class}_\text{i}) * \text{IoU}\left(\frac{\text{truth}}{\text{predict}}\right) \quad (3)$$

As represented by Eq. 1, $\text{P}_\text{r}(\text{Object})$ denotes the probability of the object in the bounding box and $\text{IoU}_{\text{predict}}^{\text{truth}}$ denotes the Intersection over Union (IoU) of the ground truth and the prediction box. Each bounding box consists of 5 parameters: (x, y, w, h, confidence). The width and height of the bounding box are represented by w, h, and x, y as the center coordinates. Confidence will be 0 if there are no objects in the cell and 1 if there are objects. In the end, the confidence prediction result will represent the IoU between the predicted box and the ground truth box. At the same time, as represented by Eq. 2, each grid cell also predicts "C," the conditional class probability in each grid cell, which is conditioned if there is an object in the grid cell. At the end of the process, as represented by Eq. 3, the testing process will multiply the conditional class probability and the confidence prediction value of the individual box to get a specific class based on the confidence score of each box so that the result encodes the probability of the class appearing in the box and represents how to match the predicted box to the object. YOLOv7 (Wang et al., 2023b) is a state-of-the-art object detection method that is a development of the previous version (Bochkovskiy et al., 2020; Wang et al., 2021), which is represented in detail in Fig. 1a represent overall architecture), Fig. 1b represents the basic part of overall architecture).

As can be shown in Fig. 1, YOLOv7 involves the utilization of a backbone layer for the purpose of feature extraction. This backbone layer is comprised of many convolutional layers, and the head layer is accountable for the generation of detection. There are various component developments that distinguish YOLOv7 from earlier architectures. These developments include the Extended efficient layer aggregation network (E-ELAN) and feature scaling. YOLOv7 is significantly different from earlier architectures. ELAN, which was developed by Wang et al. (2023), is a method that controls the shortest longest gradient route to make it possible for a deep model to learn and converge in a more effective manner. The process of scaling involves modifying certain model properties to produce models of varying sizes. As a result of the implementation of this new design, YOLOv7 delivers superior accuracy and efficiency compared to its predecessor. As part of this investigation, we will conduct a Hyperparameter batch Analysis of the YOLOv7 model during the training process. Our goal is to find the most suitable training hyperparameters to solve the issue of apple detection in the orchard setting.

3.2. Batch size

The hyperparameter under analysis is the batch size in the YOLOv7 method (Wang et al., 2023a). YOLOv7, which is a one-stage object detection algorithm based on Convolutional Neural Networks (CNNs), is the method that we take into consideration (Bishop and Bishop, 2023). During its training process, YOLOv7 employs Stochastic Gradient Descent (SGD) for optimization to minimize the loss value $L(w)$, which is represented by the following equation:

$$L(w) = \frac{1}{|X|}\sum_{x\in X} l(x, w) \quad (4)$$

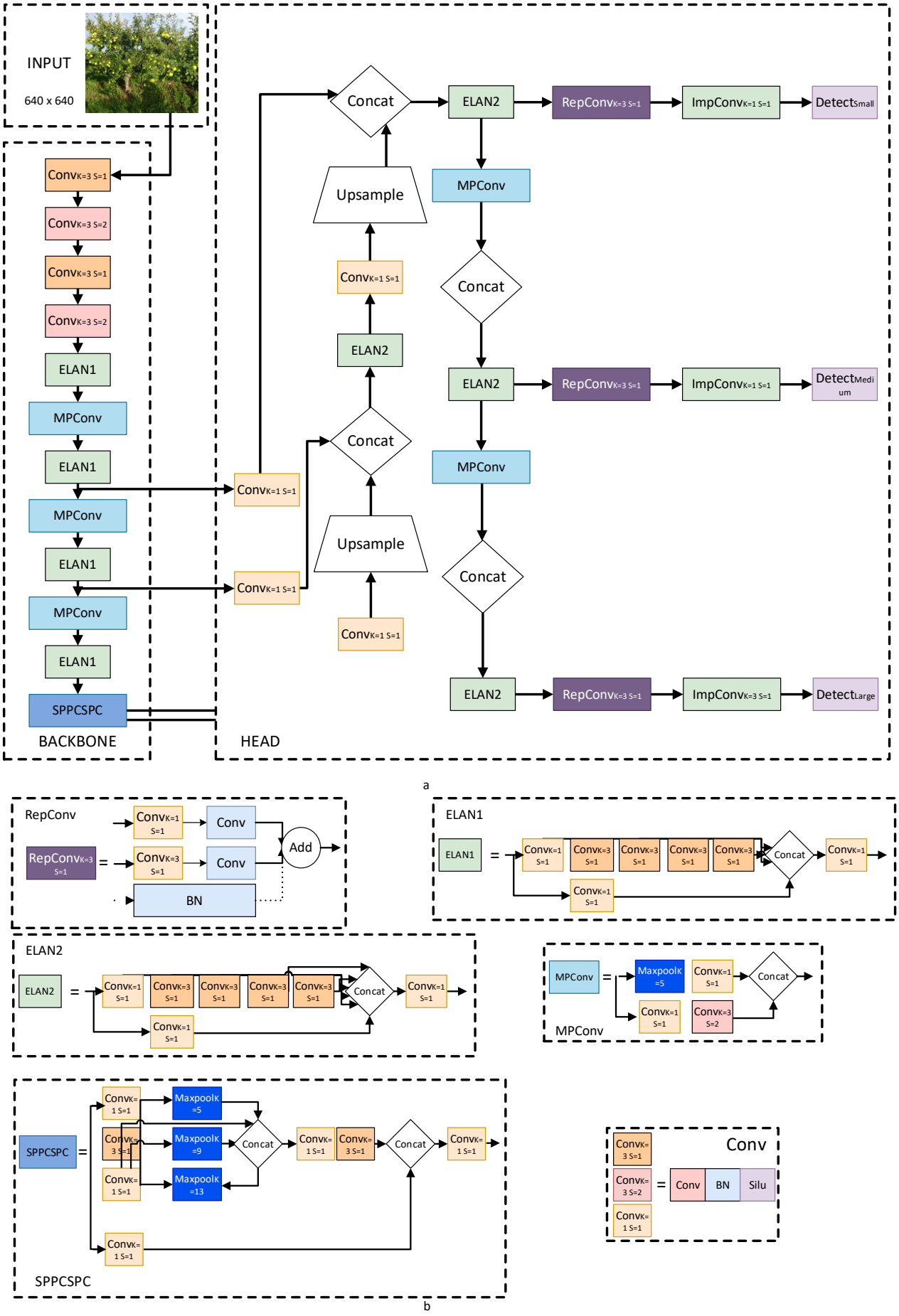

Fig. 1: Yolov7 architectural details: (a) represent overall architecture and (b) represent basic part of architecture of (a)

In Eq. 4, w represents the weight parameters of the YOLOv7 model architecture, X denotes the training dataset that has been labeled with $|X|$ samples, and $l(x, w)$ calculates the loss value from samples $x \in X$. The batch size $|B|$ determines the amount of data processed in one forward and backward pass through the network with SGD optimization, as shown in the following equation:

$$w_{t+1} = w_t - \eta \frac{1}{|B|} \sum_{x \in B} \nabla l(x, w_t) \quad (5)$$

In Eq. 5, B represents a batch of data samples from X, and $|B|$ denotes the batch size used during the training process, η represents the learning rate, and t represents the iteration index during the training process. In this study, to determine the optimal batch size for the apple detection task in the orchard, we used four different batch sizes $B_k \in \{2, 4, 8, 16\}$ during the YOLOv7 model training process.

3.3. Dataset

Regarding the objectives of this investigation, we make use of the MinneApple dataset that was acquired from Hani et al. (2020). In consideration of the fact that the MinneApple dataset is collected directly from the apple orchard environment, which involves a variety of complexities, one of the most crucial things to do in this work is to make sure that the batch size hyperparameter is adjusted correctly. On the other hand, Fig. 2c has less light than Figs. 2a-2b, which is followed by dynamic backdrop differences in each photograph. This is illustrated in Fig. 2a, which can be found below. Fig. 2b is a representation of very bright lighting. Specifically, the MinneApple dataset is made up of 670 training data, which includes ground truth and a single prediction category. On the other hand, the test data is made up of 331 image data that does not include ground truth elements. To obtain validation data, we make use of all the training data, which we then divide into two distinct categories: eighty percent, or 536 photos, serves as training data, and twenty percent, or 134 images, serves as validation data. We trained the model using training data during the training phase, and then we used validation data at the same time. Both sets of data were used simultaneously throughout the training phase. We combined test data that did not include ground truth with test data that did include ground truth and was provided by the author with a total of 331 photos of data to test the model after the training and validation method had been completed. For the aim of carrying out the examination, this was carried out.

Fig. 2: Image data of apples in an orchard environment (a) represents very bright lighting compared to (b), (c) has less light compared to (a-b), followed by dynamic background differences in each image

3.4. Evaluations metrics

To evaluate the model during the training, validation, and testing processes, we use several parameter metrics (Padilla et al., 2020). The parameters we use include Precision (P), Recall (R), Average Precision (AP), and mean Average Precision (mAP), which are measured based on 0.5 intersections over union (IoU). The parameters P and R are described by the following equation,

$$\text{Precision (P)} = \frac{\text{TP}}{\text{TP+FP}} \quad (6)$$

$$\text{Recall (R)} = \frac{\text{TP}}{\text{TP+FN}} \quad (7)$$

Based on Eqs. 6-7, TP is a true positive, which means correct detection of the ground truth bounding box, and FP is a false positive, which means the object is detected but in the wrong place. FN is a false negative, which means that the ground truth bounding box is not detected. The AP and mAP parameters are described by the following equation,

$$\text{AP} = \int_0^1 \text{P(R)dR} \quad (8)$$

$$\text{mAP} = \frac{1}{\text{N}} \sum_{\text{i=1}}^{\text{N}} \int_0^1 \text{P}_\text{i}(\text{R}_\text{i})\text{dR}_\text{i} \quad (9)$$

Based on Eq. 8, AP represents the average value of P and R, which is between 1 and 0. mAP in the Eq. 9, R_i is the recall from class i, $P_i(R_i)$ is the precision of the recall from the class is R_i, and N is the total number of classes evaluated. mAP is represents the average of AP used to measure all categories in the dataset and is a metric used to measure the accuracy of the object detection model.

4. Result and discussion

Pre-trained YOLOv7 (Wang et al., 2023b) from the Common Objects in Context (COCO) dataset (Lin et al., 2014) was utilized for the fine-tuning step during the experimental procedure that we carried out. During this time, we trained the model using 100 epochs, an input size of 640x640, a learning rate value of 0.1, a momentum of 0.9, and an IoU of 0.2. In addition, we chose to train the model with an input size of 640x640. To determine the effect of batch size, we carried out training with a variety of batch size values, including 2, 4, 8, and 16 (you can find more information about this in section 2.2). PyTorch (Paszke et al., 2019), which is a Google-Colab application powered by a Tesla T4 Graphics Processing Unit (GPU), was the framework that we utilized for the experimental approach. Several explanation sections are included in this part, which is where we describe the outcomes of the experiment. The outcomes of the training and validation procedure are displayed in (the section material and methods), which includes a variety of batch size settings. With a particular emphasis on the influence of various model input scales, the results of the model generalization analysis on test data are presented in the following section.

4.1. Training results with different batch sizes

In the following section, we will discuss the training outcomes of the YOLOv7 model with a variety of batch-size values, which are outlined in Table 1. On the other hand, the results of YOLOv7-B16 acquired a total mAP value of 50%, which is a superior difference of 1.9% compared to YOLOv7-B2, 8.2% compared to YOLOv7-B4, and 12.5% compared to YOLOv7-B1. Based on these findings, it can be concluded that the batch size value of sixteen is the most effective value for training the YOLOv7 model to perform the apple detection job in an orchard setting. However, according to the findings of the experiments, we also discovered that YOLOv7-B2, which has the smallest batch size, is the second highest by creating a total mAP of 48.1%. This is 6.3% higher than YOLOv7-B4, and it is 10.5% higher than YOLOv7-B8. According to these findings, there is a phenomenon in which a greater batch size value does not necessarily guarantee that the model will also be more accurate. This is because the results obtained for batch sizes 4 and 8 are falling at an increasing pace in comparison to size 2. Fig. 3 provides a visual representation of the detection findings.

We analyzed the model using a test set with various input scales to validate the experimental results. These scales show the model's ability to generalize when faced with real-world challenges. As shown in Table 2, the YOLOv7-B16 model achieved the highest mean Average Precision (mAP) across three different input scales, outperforming other models. In addition to achieving the highest training accuracy, the YOLOv7-B16 model also demonstrated strong generalization on unseen data. The results of YOLOv7-B4 and YOLOv7-B8 were compared to those of YOLOv7-B2, which ranked second across the three input scales. This indicates that a model trained with a batch size of 2 can be efficient and may serve as a solution for training the YOLOv7 model with limited computational resources. It is well known that larger batch sizes require more computational power. We also found that the YOLOv7-B8 model, despite having the smallest batch size, demonstrated better generalization compared to the YOLOv7-B4 model at input scales of 416x416 and 1280x1280. These results suggest that the YOLOv7-B8 model shows stronger generalization performance at both the highest and lowest scales compared to the YOLOv7-B4 model.

Table 1: The training results with different batch values

Model	Batch	Precision	Recall	mAP
YOLOv7-B2	2	52.4	54.1	48.1
YOLOv7-B4	4	48.1	47.9	41.8
YOLOv7-B8	8	45.1	45	37.5
YOLOv7-B16	16	51.4	58.1	50

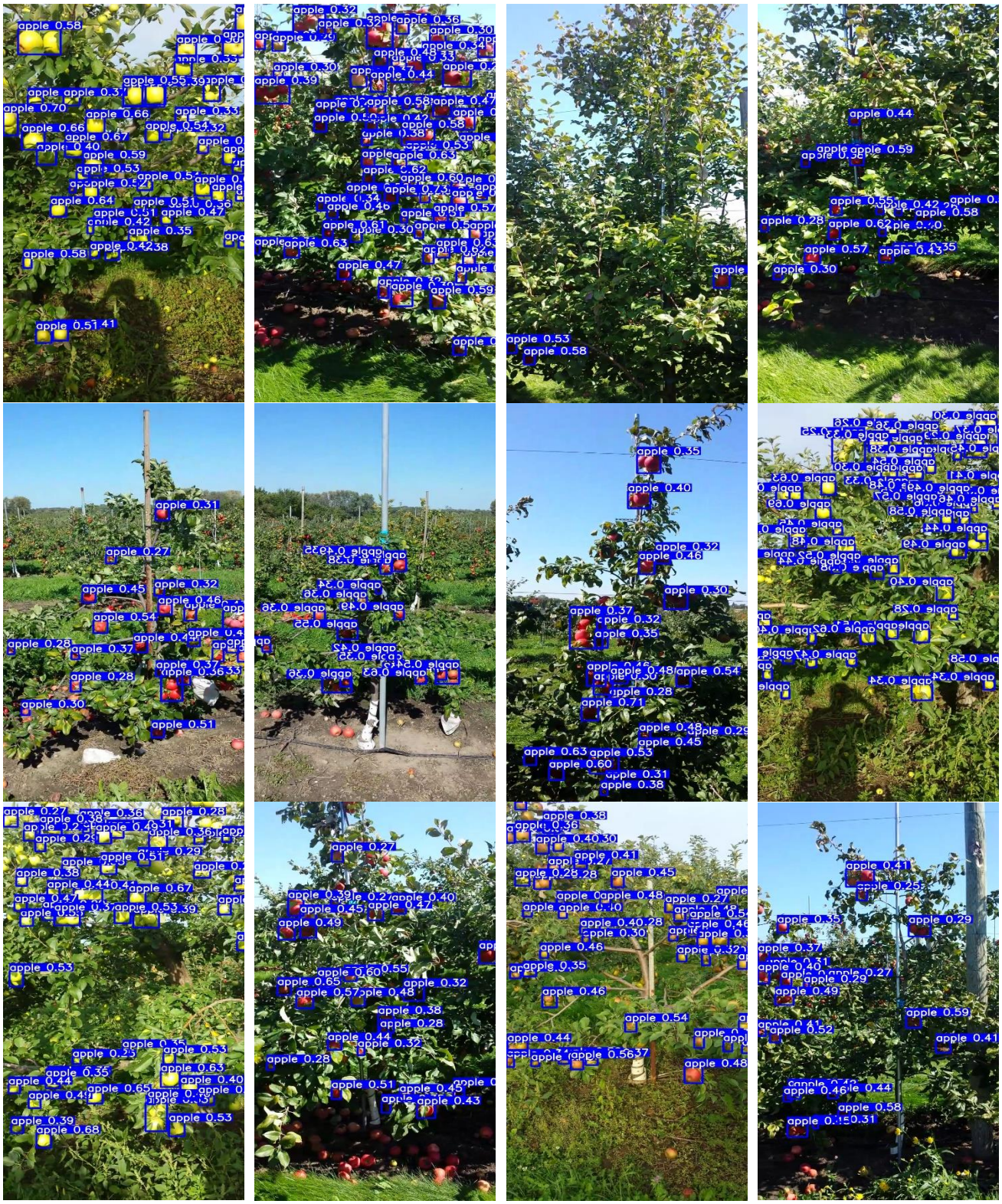

Fig. 3: Results of apple detection in the orchard environment

5. Conclusions

The objective of this study was to create a YOLOv7 model that possesses batch size values that are appropriate for apple identification in conditions that take place in orchards. To achieve this goal, we conducted experimental analysis on YOLOv7 utilizing several different batch values on numerous occasions. We utilize the MinneApple dataset inside the context of the analytic approach that we have developed. Over the course of the training phase, we provide evidence that YOLOv7-B16 is 1.9% more successful than YOLOv7-B2, 8.2% more effective than YOLOv7-B4, and 12.5% more effective than YOLOv7-B8. It can be deduced from this that the batch size value of sixteen is currently the most efficient number. Additionally, we show that increasing the batch size parameter does not necessarily guarantee that the model will be more accurate. This is something that we illustrate through our investigations. Following that, we proceeded to evaluate the level of generalization that

the model held by subjecting it to a series of real-world tasks that utilized test data and a range of input scales. In addition to obtaining the highest training accuracy, we show that YOLOv7-B16 also has the best model generalization. This is something that we have demonstrated. After this comes YOLOv7-B2, which, according to our findings, has the lowest batch value and can surpass YOLOv7-B4 and YOLOv7-B8. This is the next step in the process. A model that is trained with a batch size value of two can be a solution when the YOLOv7 model is being trained with restricted computational resources. This suggests that a model might be a solution. One further thing that this brings to light is the fact that a higher batch size number does not necessarily imply that the model will likewise be more accurate on its own.

Table 2: Results of model generalization testing with different input scales on test data

Model	Input	Precision	Recall	mAP
YOLOv7-B2	1280x1280	42.2	50	39.9
	640x640	55.4	55.4	53.1
	416x416	53.7	50.4	46.6
YOLOv7-B4	1280x1280	39.9	43.8	32.5
	640x640	50.9	48.9	45.2
	416x416	49.8	44.1	38.7
YOLOv7-B8	1280x1280	39.1	43.8	35.2
	640x640	47.6	47.3	42.8
	416x416	45.2	46.8	39.2
YOLOv7-B16	1280x1280	46.2	52.3	45.5
	640x640	56.4	55.4	54.6
	416x416	59.4	55.5	53.4

Acknowledgment

This research work was funded by Institutional Fund Projects under grant no (IFPIP:901-830-1443). The authors gratefully acknowledge the technical and financial support provided by the Ministry of Education and King Abdulaziz University, DSR, Jeddah, Saudi Arabia.

Compliance with ethical standards

Conflict of interest

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

Abdulkadirov R, Lyakhov P, and Nagornov N (2023). Survey of optimization algorithms in modern neural networks. Mathematics, 11(11): 2466. https://doi.org/10.3390/math11112466

Ahmad HM, Rahimi A, and Hayat K (2024). Capacity constraint analysis using object detection for smart manufacturing. Arxiv Preprint Arxiv:2402.00243. https://doi.org/10.48550/arXiv.2402.00243

Apostolopoulos ID, Tzani M, and Aznaouridis SI (2023). A general machine learning model for assessing fruit quality using deep image features. AI, 4(4): 812–830. https://doi.org/10.3390/ai4040041

Arnold M and Gramza-Michalowska A (2023). Recent development on the chemical composition and phenolic extraction methods of apple (*Malus domestica*)—A review. Food and Bioprocess Technology, 17: 2519-2560. https://doi.org/10.1007/s11947-023-03208-9

Bishop CM and Bishop H (2023). Convolutional networks. In: Bishop CM and Bishop H (Eds.), Deep learning: Foundations and concepts: 287-324. Springer International Publishing, Cham, Switzerland. https://doi.org/10.1007/978-3-031-45468-4_10

Bochkovskiy A, Wang CY, and Liao H-YM (2020). YOLOv4: Optimal speed and accuracy of object detection. Arxiv Preprint Arxiv:2004.10934. https://doi.org/10.48550/arXiv.2004.10934

Brock A, De S, Smith SL, and Simonyan K (2021). High-performance large-scale image recognition without normalization. Proceedings of the 38th International Conference on Machine Learning, PMLR 139: 1059-1071.

Chen G, Wang H, Chen K, Li Z, Song Z, Liu Y, Chen W, and Knoll A (2022). A survey of the four pillars for small object detection: Multiscale representation, contextual information, super-resolution, and region proposal. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 52(2): 936–953. https://doi.org/10.1109/TSMC.2020.3005231

Chen W, Zhang J, Guo B, Wei Q, and Zhu Z (2021). An apple detection method based on Des-YOLO v4 algorithm for harvesting robots in complex environment. Mathematical Problems in Engineering, 2021(1): 7351470. https://doi.org/10.1155/2021/7351470

Ge Z, Liu S, Wang F, Li Z, and Sun J (2021). YOLOX: Exceeding YOLO series in 2021. Arxiv Preprint Arxiv:2107.08430. https://doi.org/10.48550/arXiv.2107.08430

Girshick R (2015). Fast R-CNN. In the Proceedings of the IEEE International Conference on Computer Vision, IEEE, Santiago, Chile: 1440-1448. https://doi.org/10.1109/ICCV.2015.169

Girshick R, Donahue J, Darrell T, and Malik J (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Columbus, USA: 580–587. https://doi.org/10.1109/CVPR.2014.81

Goyal P, Dollár P, Girshick R, Noordhuis P, Wesolowski L, Kyrola A, Tulloch A, Jia Y, and He K (2017). Accurate, large minibatch SGD: Training ImageNet in 1 hour. Arxiv Preprint Arxiv:1706.02677. https://doi.org/10.48550/arXiv.1706.02677

Han K, Wang Y, Tian Q, Guo J, Xu C, and Xu C (2020). GhostNet: More features from cheap operations. In the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, Seattle, USA: 1580-1589. https://doi.org/10.1109/CVPR42600.2020.00165

Hani N, Roy P, and Isler V (2020). MinneApple: A benchmark dataset for apple detection and segmentation. IEEE Robotics and Automation Letters, 5(2): 852–858. https://doi.org/10.1109/LRA.2020.2965061

He K, Zhang X, Ren S, and Sun J (2016). Deep residual learning for image recognition. In the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Las Vegas, USA: 770–778. https://doi.org/10.1109/CVPR.2016.90

Hou Q, Zhou D, and Feng J (2021). Coordinate attention for efficient mobile network design. In the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, Nashville, USA: 13708–13717. https://doi.org/10.1109/CVPR46437.2021.01350

Huang J, Shang Y, and Chen H (2019). Improved Viola-Jones face detection algorithm based on HoloLens. EURASIP Journal on Image and Video Processing, 2019: 41. https://doi.org/10.1186/s13640-019-0435-6

Ji W, Pan Y, Xu B, and Wang J (2022). A real-time apple targets detection method for picking robot based on ShufflenetV2-YOLOX. Agriculture, 12(6): 856. https://doi.org/10.3390/agriculture12060856

Kandel I and Castelli M (2020). The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. ICT Express, 6(4): 312–315. https://doi.org/10.1016/j.icte.2020.04.010

Kaplun D, Deka S, Bora A, Choudhury N, Basistha J, Purkayastha B, Mazumder IZ, Gulvanskii V, Sarma KK, and Misra DD (2024). An intelligent agriculture management system for rainfall prediction and fruit health monitoring. Scientific Reports, 14(1): 512. https://doi.org/10.1038/s41598-023-49186-y

Keskar NS, Mudigere D, Nocedal J, Smelyanskiy M, and Tang PTP (2016). On large-batch training for deep learning: Generalization gap and sharp minima. Arxiv Preprint Arxiv:1609.04836. https://doi.org/10.48550/arXiv.1609.04836

Kuznetsova A, Maleva T, and Soloviev V (2020). Using YOLOv3 algorithm with pre-and post-processing for apple detection in fruit-harvesting robot. Agronomy, 10(7): 1016. https://doi.org/10.3390/agronomy10071016

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

LeCun Y, Bottou L, Orr GB, and Müller KR (2002). Efficient backprop. In: Orr GB and Müller KR (Eds.), Neural networks: Tricks of the trade: 9-50. Springer Berlin Heidelberg, Berlin, Germany. https://doi.org/10.1007/3-540-49430-8_2

Lin TY, Goyal P, Girshick R, He K, and Dollar P (2020). Focal loss for dense object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 42(2): 318–327. https://doi.org/10.1109/TPAMI.2018.2858826

Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, and Zitnick CL (2014). Microsoft COCO: Common objects in context. In the Computer Vision–ECCV 2014: 13th European Conference, Springer International Publishing, Zurich, Switzerland: 740-755. https://doi.org/10.1007/978-3-319-10602-1_48

Lin Z, Gao W, Jia J, and Huang F (2021). CapsNet meets SIFT: A robust framework for distorted target categorization. Neurocomputing, 464(24): 290–316. https://doi.org/10.1016/j.neucom.2021.08.087

Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, and Berg AC (2016). SSD: Single shot multibox detector. In the Computer Vision–ECCV 2016: 14th European Conference, Springer International Publishing, Amsterdam, Netherlands: 21-37. https://doi.org/10.1007/978-3-319-46448-0_2

Ma N, Zhang X, Zheng HT, and Sun J (2018). ShuffleNet V2: Practical guidelines for efficient CNN architecture design. In: Ferrari V, Hebert M, Sminchisescu C, and Weiss Y (Eds), Computer vision – ECCV 2018. Lecture notes in computer science: 122–138. Volume 11218, Springer, Cham, Switzerland. https://doi.org/10.1007/978-3-030-01264-9_8

Onishi Y, Yoshida T, Kurita H, Fukao T, Arihara H, and Iwai A (2019). An automated fruit harvesting robot by using deep learning. ROBOMECH Journal, 6(1): 2–9. https://doi.org/10.1186/s40648-019-0141-2

Padilla R, Netto SL, and Da Silva EA (2020). A survey on performance metrics for object-detection algorithms. In the International Conference on Systems, Signals and Image Processing, IEEE, Niteroi, Brazil: 237-242. https://doi.org/10.1109/IWSSIP48289.2020.9145130

Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, Desmaison A, Köpf A, Yang E, DeVito Z, Raison M, Tejani A, Chilamkurthy S et al. (2019). Pytorch: An imperative style, high-performance deep learning library. Advances in Neural Information Processing Systems, 32: 8024–8035.

Pebrianto W, Mudjirahardjo P, and Pramono S H (2022). YOLO method analysis and comparison for real-time human face detection. In the 11th Electrical Power, Electronics, Communications, Controls and Informatics Seminar, IEEE, Malang, Indonesia: 333-338. https://doi.org/10.1109/EECCIS54468.2022.9902919

Pebrianto W, Mudjirahardjo P, and Pramono SH (2024). Partial half fine-tuning for object detection with unmanned aerial vehicles. IAES International Journal of Artificial Intelligence (IJ-AI), 13(1): 399-407. https://doi.org/10.11591/ijai.v13.i1.pp399-407

Qian X and Klabjan D (2020). The impact of the mini-batch size on the variance of gradients in stochastic gradient descent. Arxiv Preprint Arxiv:2004.13146. https://doi.org/10.48550/arXiv.2004.13146

Redmon J and Farhadi A (2017). YOLO9000: Better, faster, stronger. In the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Honolulu, USA: 6517–6525. https://doi.org/10.1109/CVPR.2017.690

Redmon J and Farhadi A (2018). YOLOv3: An incremental improvement. Arxiv Preprint Arxiv:1804.02767. https://doi.org/10.48550/arXiv.1804.02767

Redmon J, Divvala S, Girshick R, and Farhadi A (2016). You only look once: Unified, real-time object detection. In the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Las Vegas, USA: 779-788. https://doi.org/10.1109/CVPR.2016.91

Ren S, He K, Girshick R, and Sun J (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(6): 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031

Sato N and Iiduka H (2023). Existence and estimation of critical batch size for training generative adversarial networks with two time-scale update rule. In the International Conference on Machine Learning, PMLR, Honolulu, USA: 30080-30104.

Stapor P, Schmiester L, Wierling C, Merkt S, Pathirana D, Lange BMH, Weindl D, and Hasenauer J (2022). Mini-batch optimization enables training of ODE models on large-scale datasets. Nature Communications, 13: 34. https://doi.org/10.1038/s41467-021-27374-6

Sun L, Hu G, Chen C, Cai H, Li C, Zhang S, and Chen J (2022). Lightweight apple detection in complex orchards using YOLOV5-PRE. Horticulturae, 8(12): 1169. https://doi.org/10.3390/horticulturae8121169

Sun Z, Caetano E, Pereira S, and Moutinho C (2023). Employing histogram of oriented gradient to enhance concrete crack detection performance with classification algorithm and Bayesian optimization. Engineering Failure Analysis, 150: 107351. https://doi.org/10.1016/j.engfailanal.2023.107351

Tan M and Le Q (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. In the International Conference on Machine Learning, PMLR, Long Beach, USA: 6105-6114.

Tan M, Pang R, and Le QV (2020). EfficientDet: Scalable and efficient object detection. In the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, Seattle, USA: 10778–10787. https://doi.org/10.1109/CVPR42600.2020.01079

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

Wang CY, Bochkovskiy A, and Liao HYM (2021). Scaled-YOLOv4: Scaling cross stage partial network. In the IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, Nashville, USA: 13024–13033. https://doi.org/10.1109/CVPR46437.2021.01283

Wang CY, Bochkovskiy A, and Liao HYM (2023a). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In the IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, Vancouver,

Canada: 7464–7475. https://doi.org/10.1109/CVPR52729.2023.00721

Wang CY, Liao HY M, and Yeh IH (2023b). Designing network design strategies through gradient path analysis. Journal of Information Science and Engineering, 39(2): 975–995.

Wang D and He D (2022). Apple detection and instance segmentation in natural environments using an improved mask scoring R-CNN model. Frontiers in Plant Science, 13: 1016470. https://doi.org/10.3389/fpls.2022.1016470

Woo S, Park J, Lee J, and Kweon IS (2018). CBAM: Convolutional block attention module. In: Ferrari V, Hebert M, Sminchisescu C, and Weiss Y (Eds.), Computer vision – ECCV 2018, Lecture notes in computer science: 3–19. Volume 11211, Springer, Cham, Switzerland. https://doi.org/10.1007/978-3-030-01234-2_1

Wu L, Ma J, Zhao Y, and Liu H (2021). Apple detection in complex scene using the improved YOLOv4 model. Agronomy, 11(3): 476. https://doi.org/10.3390/agronomy11030476

Xiao F, Wang H, Xu Y, and Zhang R (2023). Fruit detection and recognition based on deep learning for automatic harvesting: An overview and review. Agronomy, 13(6): 1625. https://doi.org/10.3390/agronomy13061625

Xuan G, Gao C, Shao Y, Zhang M, Wang Y, Zhong J, Li Q, and Peng H (2020). Apple detection in natural environment using deep learning algorithms. IEEE Access, 8: 216772-216780. https://doi.org/10.1109/ACCESS.2020.3040423

Yong H, Huang J, Meng D, Hua X, and Zhang L (2020). Momentum batch normalization for deep learning with small batch size. In the Computer Vision–ECCV 2020: 16th European Conference, Springer International Publishing, Glasgow, UK: 224-240. https://doi.org/10.1007/978-3-030-58610-2_14

Yoshida T, Kawahara T, and Fukao T (2022). Fruit recognition method for a harvesting robot with RGB-D cameras. ROBOMECH Journal, 9: 15. https://doi.org/10.1186/s40648-022-00230-y

You Y, Li J, Reddi S, Hseu J, Kumar S, Bhojanapalli S, Song X, Demmel J, Keutzer K, and Hsieh CJ (2019). Large batch optimization for deep learning: Training BERT in 76 minutes. Arxiv Preprint Arxiv:1904.00962. https://doi.org/10.48550/arXiv.1904.00962

Zhang X, Zhou X, Lin M, and Sun J (2018). ShuffleNet: An extremely efficient convolutional neural network for mobile devices. In the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Salt Lake City, USA: 6848-6856. https://doi.org/10.1109/CVPR.2018.00716

Zhao Z, Wang J, and Zhao H (2023). Research on apple recognition algorithm in complex orchard environment based on deep learning. Sensors, 23(12): 5425. https://doi.org/10.3390/s23125425