Paper Title* (use style: paper title)

Abstract - Apple counting in orchards traditionally relies on manual labor, which is time-consuming and error prone. Computer vision technology provides a new possibility for the automated management of orchards, through image algorithms to achieve automatic detection and counting of apples, thereby improving efficiency and reducing labor costs. This paper compares the detection performance of traditional methods based on color extraction and edge detection with the YOLOv5 algorithm in an orchard environment. The traditional method has low computational cost but poor robustness to light and complex background, while YOLOv5 shows high accuracy and strong robustness under occlusion, overlapping and complex background conditions through multi-layer feature extraction and optimization techniques. The experimental results verify the advantages of YOLOv5 and explore its applicability under the condition of limited hardware resources, which provides an important reference for the automation of orchard management.

i. INTRODUCTION

Orchards are central sites for fruit production, where the management and yield assessment of apple trees is critical to fruit growers and the agricultural economy. Traditionally, counting apples in the orchard relies on manual methods, which are both time-consuming and susceptible to human error. In recent years, computer vision technology has provided a whole new opportunity for automated orchard management, automatically identifying and counting apples through image-based algorithms, which not only improves efficiency but also significantly reduces labor costs.

The research hypothesis includes consistent ambient light conditions in the orchard, and under the problem of possible occlusion and overlapping of apples, YOLOv5 will recognize apples better than the traditional computer vision recognition approach based on extracting apple color and performing edge detection.

The main goal of the research is to explore how to achieve high accuracy in apple identification and counting in complex orchard environments. Specifically, the research will utilize YOLOv5, a target detection framework in deep learning, to compare with traditional image processing methods in order to demonstrate the advantages of YOLOv5 in orchard applications. Also analyze the difference in performance between traditional methods and deep learning techniques when dealing with the below challenges.

Shading and Overlap: apples may be partially invisible in the shade of leaves or branches, or overlap may occur due to dense distribution.

Background Complexity: the background in an orchard may include sky, soil, tree trunks, and other complex elements that may interfere with the algorithm.

Computational Resources: deploying an inspection system for orchards may be limited by hardware resources and requires a trade-off between model accuracy and real-time performance.

ii. RELATED WORKS

In recent years, apple counting and detection has become a popular research direction in computer vision and agricultural automation. The related literature mainly focuses on two categories: traditional image processing-based methods and deep learning methods.

A. Traditional Methods

Fruit visual inspection systems typically operate through five stages: fruit image acquisition, fruit image preprocessing, fruit feature extraction, fruit image segmentation, and fruit image recognition [1]. Traditional computer vision recognition methods mainly rely on image processing techniques such as color segmentation, shape analysis and edge detection. These methods usually segment apples based on their color features, mainly such as red, green and yellow. This research just uses based on extracting the color of the apple and removing the noise by canny edge detection to finally achieve the recognition of the apple.

However, these methods are sensitive to light variations and background complexity, and the detection accuracy decreases significantly when there are strong lights, shadows, or background objects with colors close to apples in the orchard environment, especially when there is occlusion and overlap between leaves and apples. The difficulty and complexity of research on apple recognition using computer vision is due to the difficulty in establishing the best method for extracting certain image features. Apples on a fruit tree cannot avoid occlusion and overlap with the leaves, so there is no real way to model the best way to recognize apples. Besides, as discussed by Simões and Costa, the automation of processes based on digital images presents as main difficulties: "(1) the nonexistence of a formal description of patterns; (2) the nonexistence of computational tools and consolidated models for classification; (3) dependence on the conditions of illumination of the environment" [2].

B. Deep Learning Methods

With the rise of deep learning, target detection algorithms show great potential in apple detection. Convolutional neural networks (CNNs) can better adapt to complex scenes and lighting conditions by learning multi-level features of apples. You Only Look Once (YOLO) is a real-time deep learning algorithm widely used in target detection tasks, first proposed by Joseph Redmon et al. in 2016 [3]. YOLO treats target detection as a single neural network regression problem, dividing the image into regions and predicting the bounding box and probability of each region, enabling efficient target localization and classification [4]. The YOLO series of algorithms is widely used in agriculture as a representative of real-time target detection.

YOLOv5, the newer version of the YOLO series used in this research, achieves a better balance between detection speed and accuracy by introducing mosaic enhancement techniques and an improved loss function [5]. In addition, deep learning-based methods are able to incorporate data enhancement techniques such as random

cropping and luminance adjustment to further improve the robustness of the model.

C. Comparison and Discussion

The advantages of traditional methods are low computational cost and simple implementation, but their robustness to environmental conditions is poor, especially in complex lighting and background scenes. Deep learning methods, especially YOLOv5, on the other hand, have high accuracy, strong robustness and good real-time performance, and are capable of handling complex problems such as occlusion and overlapping. However, limitations of deep learning methods include the need for large-scale labeled data and potentially higher computational resource requirements.

In summary, YOLOv5 is chosen as the core model in this research to verify its superiority in detecting apples in orchard environments by comparing it with traditional computer vision recognition methods based on color extraction and edge detection, as well as to explore its applicability under the condition of limited hardware resources.

iii. DATA PREPARE

The dataset used in this project is the MinneApple dataset published by the Horticultural Research Center at the University of Minnesota [1].The main purpose of this dataset is to provide a standardized data resource for the development of automated harvesting systems, fruit counting systems, and detection algorithms, as well as to meet the requirements of the task of calculating apple counts in this project. The MinneApple dataset was created by using a Samsung Galaxy S4 cell phone recording video at approximately 1 m/s from which training and test images were extracted. The data collection covered different areas of the orchard, including different weather and lighting conditions and backgrounds, and contained 1,000 images and more than 41,000 instances of labeled apples and was divided into Counting, Detection, and Test datasets, and the Detection data portion was selected for training and testing. For the MinneApple test dataset there are two labeled files attached, ground_truth.json and mapping.json, which are mainly used for model evaluation.

- The ground_truth.json file records the real labeling information of all the targets in the test set, including the target category, bounding box coordinates, and target ID of each image, which is the basis for evaluating the model's detection accuracy, and is used for calculating the model's average accuracy and other performance metrics.
- The mapping.json file establishes the mapping relationship between the image file names and the labeled files in the test set, which ensures that the prediction results can be correctly matched to the corresponding labeled data to avoid data confusion during the evaluation.

D. Data quality

The Detection part of the MinneApple dataset consists of captured images in Figure 1, and corresponding labeled mask images (Figure 1), which have a uniform resolution of 720 × 1280.However, since the dataset is captured in the context of a real environment, where the color and contrast of the target changes due to the weather and illumination, and a large amount of the apples can be heavily obscured by branches and leaves, which increases the likelihood of false detection,

and there is an imbalance in the number of apples in each piece of data, these challenges increase the difficulty of detection.

Figure 1 - Detect image instances in a dataset and masks instance with annotations

iv. METHOD

E. Data preprocessing

Since the MinneApple dataset used for this project does not directly give the label format data needed to train YOLOv5, the project needs to manually extract the location of the apple target in each training image from the mask file provided in the dataset and generate the corresponding label file. Specifically, the project utilized different pixel values (1 to 255) in the mask image to label the apple regions, this is because the pixel value of the background part is 0, that is, each non-zero pixel value corresponds to one apple uniquely, so that the project can guarantee that all apples in each training image can be labeled. Secondly for each apple the minimum and maximum coordinates of the bounding box are extracted and the coordinates of the center point of the bounding box and its width and height are calculated. Subsequently, this information is normalized to a range of image sizes and all the training set masked images are traversed and saved as YOLOv5 usable label format files ready for model training, overwriting the label format data back to its corresponding image instances are shown in Figure 2.

Figure 2 -The label data is overwritten back to the corresponding training image

Secondly, to effectively train and validate the yolo model as well as to objectively evaluate the performance of the model, the project divides the dataset into a training set and a validation set, where 80% of the images are used for training and 20% of the images are used for validation. Each image and its corresponding label file are randomly divided and stored in predefined folders

F. Traditional

G. Machine Vision

1) Deep learn yolov5

YOLO (You Only Look Once) is a real-time object detection algorithm that was first proposed by Joseph Redmon et al. in 2015[2] and trained and validated on the COCO dataset. At its core, YOLO processes the entire image using a single neural network, dividing the image into multiple grid regions and predicting both the target class probability and the bounding box position for each region. In this project, the fifth version of YOLO (YOLOv5) was selected, which has higher

detection accuracy and efficiency, and is suitable for complex Apple object detection tasks.

1) Training

When training YOLOV5m, the project selected several key parameters to optimize the training results. The pre-training weights of YOLOv5m were used to speed up the convergence rate. The initial learning rate was set to be set to 0.01, which was combined with a stochastic gradient descent optimizer (SGD) to ensure the stability and efficiency of weight adjustment. The training batch size is set to 16 to balance the memory usage and computational speed. The input image size is set to 640 pixels, and the project is trained for 50 rounds, because 50 rounds were found to be sufficient for the model to fully learn the features of the dataset and will not lead to the occurrence of overfitting.

2) Evaluation methods

The project used two ways to evaluate the performance of the model, which are Target Detection Accuracy Evaluation and Apple Counting Accuracy Evaluation are used to compare with the traditional methods. Firstly, the AP (Average Precision) is evaluated as shown in Equation 1.

$$AP = \int_0^1 Precision(Recall)dRecall$$

Equation 1 - Average precision

Precision denotes the proportion of correctly detected targets to the total predicted targets, and Recall denotes the proportion of correctly detected targets to all real targets. By integrating the PR curve, the AP value can be obtained, demonstrating the performance of the model under different thresholds.

$$Accuracy_{count} = 1 - \frac{\left|N_i - \widehat{N}_i\right|}{N_i}$$

Equation 2 - For the evaluation of apple counting accuracy, the project used the counting accuracy formula.

In the Equation 2, where Ni denotes the true apple count of the number of i test image and $\widehat{N}_i$ denotes the predicted apple count of the number of I image. The formula derives the counting accuracy of a single image by calculating the relative error between the predicted and true values. Ultimately, the average of all test images is taken as the overall counting accuracy of the model.

v. RESULT

Figure 3 - All result in dataset

Figure 3 shows two plots of the results after re-testing the trained model in the test set, where the blue box means that the apple was detected and boxed out, while the number next to it indicates the confidence level, meaning how likely the boxed-out apple is to be recognized as an apple.

The LOSS curves during training and validation are shown in Figure 4, which focuses only on the bounding box regression (Box Loss) of the target and does not incorporate a classification loss because the project has only one target with only

one category, Apple. This Loss curve graph shows how the target localization error of the model changes with the number of training rounds in the training and validation phases, the horizontal coordinate is the number of training rounds, and the vertical coordinate is the value of the loss, Box Loss mainly measures the error between the model-predicted bounding box and the real labeled box, and the lower and lower Box Loss proves that the model's target localization accuracy is higher. In the figure the project can see that the red line is the Box Loss of training and the yellow line is the validation, in the initial 10 rounds of rapid decline in the training of the loss from 0.13 to 0.065 validation of the loss from 0.10 to 0.06 shows that the model in the initial stage of rapid learning of the characteristics of the apple target, the localization error decreases rapidly, and the mid-term of the 10 to 30 rounds of training The loss of training and validation in the middle stage from 10 to 30 rounds is still decreasing slowly, and in the last stage from 30 to 50 rounds the loss is almost stabilized in the range of 0.04-0.05, the model is close to convergence, and it also indicates that the number of training rounds of the project is enough to satisfy the learning of the model.

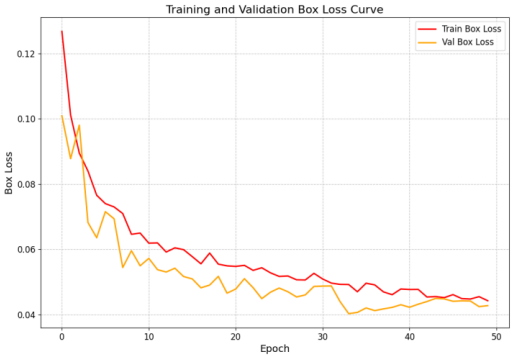

Figure 4 Training and Validation Box Loss Convergence Curve

In Figure 5, the item can see the trend of the Precision-Recall curve of the model during the training process, which illustrates the model's ability to detect apples under different training rounds, the precision rate is the green fold, the recall rate is the red fold, the horizontal coordinate is the training rounds, the vertical coordinate is the Metrics, in the pre-training period (0-5 rounds) both Precision and Recall are both low and fluctuate significantly because the model has not effectively learned the features of apples at the early stage of training, resulting in more misdetections and omissions, while in rounds 5-20, both Precision and Recall are rapidly increasing, the model starts to reduce the target of misdetections, and the increase in Recall indicates that the model's ability to detect apples is gradually increasing. And in the later rounds 20-50 Precision and Recall both leveled off to around 0.8, indicating that the model has converged. Secondly, it can be seen in the figure that Precision is higher than Recall, which indicates that the model performs better in reducing false detections, but there are leakage detections for some obscured or small-sized targets.

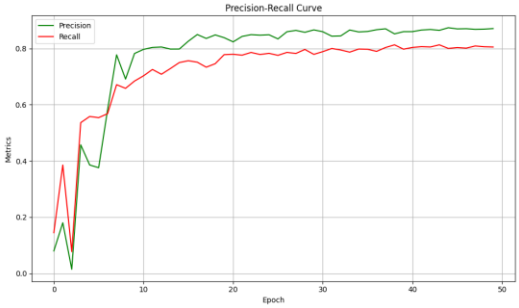

Figure 5 The curve for precision-recall in training process

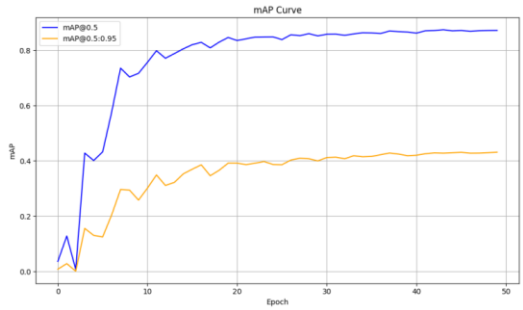

Figure 6 mAP curve in training

Figure 6 shows the mAP (mean Average Precision) curve in training, which has two metrics: mAP-0.5 and mAP-0.5:0.95. The mAP-0.5 indicates the mean Average Precision under the threshold of IoU (Intersection over Union) = 0.5, while mAP-0.5:0.95 is more stringent. mAP-0.5 represents the mean Average Precision at the IoU (Intersection over Union) = 0.5 threshold. IoU is a metric to evaluate the degree of overlap between the predicted bounding box and the real bounding box, whereas mAP-0.5:0.95 is more stringent in that it synthesizes the average precision of multiple IoU thresholds from 0.5 to 0.95. mAP-0.5 is the average precision of the predicted bounding box and the real bounding box. The mAP-0.5 is shown by the blue line and the mAP-0.5:0.95 is shown by the yellow line. The project can clearly see that the average accuracy stabilizes around 0.85 when the IoU is 0.5, indicating that the model performs well under looser detection criteria and can accurately detect most targets. However, as the IoU threshold is raised from 0.5 to 0.95, the evaluation criteria gradually become stricter, and the model not only needs to predict the target location, but also needs to accurately fit the target bounding box, which eventually stabilizes at around 0.4. This also shows the limitations of the model in high accuracy requirements, such as the difficulty of accurately fitting the bounding box when dealing with data where the targets are densely distributed, overlapped, or heavily occluded by foliage.

IoU Threshold	Area	Average Precision
0.20:0.95	Small	0.268
	Medium	0.625
	Large	0.767
0.50:0.95	Small	0.185
	Medium	0.513
	Large	0.657

Table 1 Average precision in different threshold

Table 4 demonstrates the average accuracy for different scale targets in the test set results, when the IoU threshold is 0.20:0.95, the average accuracy is 0.268 in the small target region, 0.625 in the medium target region, and the highest AP in the large target region, which is 0.767. This suggests that the model performs best in detecting large-size apples, and poorly in detecting apples of small sizes. In the apple detection task of this project, the large apples were those that were clearly visible and not obscured by branches and leaves, while the small apples were targets that were heavily obscured by leaves or other apples, or appeared to be small because they were far away from the camera. The model achieves higher accuracy in the large target region and lower accuracy in the small target region due to occlusion, overlap, and image resolution limitations, making it difficult for the model to accurately detect these apples.

When the IoU threshold is increased to 0.50:0.95, the overall average accuracy decreases, with the AP decreasing to 0.185 for the small-target region, 0.513 for the medium-target region, and 0.657 for the large-target region. This suggests that the more stringent IoU threshold requires that the model needs to be more accurate in its prediction of apple

bounding box locations and sizes. By comparing the average accuracy of large target apples is still high, indicating that the model in this project is robust in detecting unobstructed and well-distributed apples.

Figure 7 - Detection effect

The project can also observe the test set result examples in Figure 7, where the green box indicates the real labeled box provided by the test set, and the red box indicates the detection results predicted by the model, the project sets the two boxes together to show the results, by pairing the two detection result diagrams the project can see that ID-2 has a better detection effect, the main reason is that most of the apple targets in the diagram are large and clearly visible, and the red box and the green box overlap, and less occlusion, the background is relatively simple, so that the model can more easily capture the boundary features of the apple.

While ID-323 has a poor detection effect, with more omissions and false detections. The project can be seen in ID-323 has many leaves, most of the apples are heavily occluded by leaves and branches and shadows resulting in a large number of small targets. Combined with the results in Table 1, the project can see that the Average Precision value for small targets is low, which also confirms the model's deficiency in detecting small targets.

vi. REFERENCES

[1] Xiao, F. et al. (2023) 'Fruit Detection and Recognition Based on Deep Learning for Automatic Harvesting: An Overview and Review', Agronomy, 13(6), p. 1625. doi:10.3390/agronomy13061625.

[2] Gomes, J.F. and Leta, F.R. (2012) 'Applications of computer vision techniques in the Agriculture and Food Industry: A Review', European Food Research and Technology, 235(6), pp. 989–1000. doi:10.1007/s00217-012-1844-2.

[3] Redmon, J. et al. (2016) 'You only look once: Unified, real-time object detection', 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788. doi:10.1109/cvpr.2016.91.

[4] Afonso, M. et al. (2020) 'Tomato fruit detection and counting in greenhouses using Deep Learning', Frontiers in Plant Science, 11. doi:10.3389/fpls.2020.571299.

[5] Khanam, R. and Hussein, M. (2024) 'What is YOLOv5: A deep look into the internal features of the popular object detector', arXiv, Available at: https://arxiv.org/html/2407.20892 (Accessed: 13 December 2024).

[6] Häni, N., Roy, P., & 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

[7] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016

[8] Wang, C. Y., Liao, H. Y. M., Yeh, I. H., Wu, Y. H., Chen, P. Y., & Hsieh, J. W. (2019). *CSPNet: A New Backbone that can EnCSPNet: A New Backbone that can Enhance Learning Capability of CNN. arXiv. https://arxiv.org/abs/1911.11929

[9] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. arXiv. https://arxiv.org/abs/1512.03385

[10] He, K., Zhang, X., Ren, S., & Sun, J. (2014). Spatial pyramid pooling in deep convolutional networks for visual recognition. In Computer Vision – ECCV 2014 (pp. 346–361). Springer Internathttps://doi.org/10.1007/978-3-319-10578-9_23

[11] Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J. (2018). Path Aggregation Network for Instance Segmentation. arXiv preprint arXiv:1803.01534. Retrieved fhttps://arxiv.org/abs/1803.01534