Author 1 (one author only)

First Name (or initial)	Middle Name (or initial)	Surname	Suffix (Jr., III, etc.)	Optional http://orcid.org ORCID	Email	Contact author? yes or no
Xuehai	XZ	Zhou	Mr.		xue-hai.zhou@mail.mcgill.ca	yes

ASABE use

Author 2 (one author only)

First Name (or initial)	Middle Name (or initial)	Surname	Suffix (Jr., III, etc.)	Optional http://orcid.org ORCID	Email	Contact author? yes or no
Yuyang	YZ	Zhang	Mr.		yuyang.zhang@mail.mcgill.ca	No

ASABE use

Author 3

First Name (or initial)	Middle Name (or initial)	Surname	Suffix (Jr., III, etc.)	Optional http://orcid.org ORCID	Email	Contact author? yes or no
Shangpeng Sun	SS	Sun	Dr.		shangpeng.sun@mcgill.ca	yes

ASABE use

Author 4

First Name (or initial)	Middle Name (or initial)	Surname	Suffix (Jr., III, etc.)	Optional http://orcid.org ORCID	Email	Contact author? yes or no
Phil	PR	Rosenbaum	Mr.		phil@vertite.ca	no

ASABE use

Paper number and page range

Paper number on the line below 2301450
Pages 1-Pages 9

2950 Niles Road, St. Joseph, MI 49085-9659, USA
269.429.0300 fax 269.429.3852 hq@asabe.org www.asabe.org

An ASABE Meeting Presentation
DOI: https://doi.org/10.13031/aim.202301450
Paper Number: 2301450

A Dynamic Object Counting Method for Strawberry Fruits using Vision Transformer Networks and Kalman Filter Tracking

Xuehai Zhou[1], Yuyang Zhang[1], Shangpeng Sun[1], Phil Rosenbaum[2]

[1]Department of Bioresource Engineering, McGill University, Montréal, Québec

[2]Vertité, Québec

Written for presentation at the
2023 ASABE Annual International Meeting
Sponsored by ASABE
Omaha, Nebraska
July 9-12, 2023

ABSTRACT. *Strawberries are an important economic crop. Precise counting of strawberry fruits can assist growers in estimating yield and arranging product flow into the market in advance. In this study, we proposed a dynamic object counting method for strawberry fruits from videos captured using a cell phone. Firstly, an improved YOLOv5s was developed by integrating a vision transformer layer to detect strawberry fruits in each frame from the input video. Secondly, we then proposed a dynamic counting algorithm to count the total number of fruits in the video. The counting algorithm consists of two modules: StrongSORT tracking and DynCount. The StrongSORT framework, a state-of-the-art tracking algorithm based on Kalman filter and Hungarian matching, is used to track the identities of strawberry fruits in the video data, while the DynCount algorithm maps the tracking IDs to counting numbers, and displays the counting numbers along with bounding boxes. Experimental results with ten videos showed that our approach achieved an average counting accuracy of 92.05%. Our method provides accurate and timely strawberry counting information, which can alleviate the labor burden and provide important reference indicators for yield prediction and market planning.*

Keywords. *YOLOv5, vision transformer, Kalman filter, Hungarian matching, fruit counting.*

Introduction

Yield prediction is one of the major concerns for growers and fruit counting is the fundamental quantification for yield prediction. There are three main techniques used for fruit counting, which are 3D data-based counting, counting-by-regression, and counting-by-detection. Due to the additional depth information, 3D data provides more details in farm scenes, but acquiring depth information for 3D scenes such as using LiDAR sensors is often expensive. In addition, rendering 3D scenes requires high computing power, and processing 3D data often needs more computing resources (Sun et al., 2020). Therefore, at the current stage, 3D counting methods are not widely applicable to large-scale farms. Counting-by-regression model, such as a single-output convolutional neural network (CNN), can predict a scalar counting number for an input image without localization information of objects (Segui, Pujol, Vitria, & Ieee, 2015). Unfortunately, in our scenario, even with a wide-angle lens, a single photo is not sufficient to cover all strawberry plants on our cultivation racks. For the application of this method, usually, multiple images need to be captured and stitched before counting. In this case, we need to consider the loss during image stitching, as well as the potential interpretation loss of the regression model on long images. The counting-by-detection algorithm is a variant derived from tracking-by-detection, which can display

bounding boxes for location information. With the prevalence of deep learning-based object detection methods, tracking-by-detection has emerged as a promising research area (Meimetis, Daramouskas, Perikos, & Hatzilygeroudis, 2023). However, the original research of tracking algorithms emphasizes the movement of crowds or vehicles under surveillance cameras, rather than counting problems. A customized counting module is needed if this method is applied. In terms of cost-efficiency and responsiveness, we believe that counting-by-detection is the best solution for large-scale applications.

Object detection algorithms serve as the cornerstone for counting-by-detection methods, and in recent years, deep learning-based object detection algorithms have demonstrated high levels of accuracy. Current deep learning-based object detection algorithms can be divided into two categories: one-stage and two-stage. In general, one-stage object detection algorithms, such as YOLO (Redmon, Divvala, Girshick, Farhadi, & Ieee, 2016) and SSD (Liu et al., 2016), predict the class and position all in once, while two-stage algorithms, such as Faster R-CNN (Ren, He, Girshick, & Sun, 2015) and Mask R-CNN (He, Gkioxari, Dollár, & Girshick, 2017), generate region proposals first and then perform classification and regression on them. Therefore, two-stage algorithms are more accurate but slower, while one-stage algorithms are faster but less accurate. Vision transformer was proposed to increase the detection accuracy for one-stage algorithms while maintaining high detection speed. The transformer framework emerged in the field of natural language processing, where researchers proposed the self-attention mechanism and made significant progress in feature extraction (Vaswani et al., 2017). Researchers have also applied self-attention mechanisms to the field of computer vision, giving rise to the vision transformer (Han et al., 2023).

Since object detection algorithms do not identify the same objects across adjacent frames in video data, tracking algorithms were introduced to record the same target identities. There have been significant breakthroughs for SORT family algorithms in multi-object tracking (MOT), where SORT stands for simple online and real-time tracking (Bewley, Ge, Ott, Ramos, & Upcroft). SORT predicts object positions in the next frame using a linear Kalman filter and matches the IoU (Intersection over Union) of position prediction results and detected bounding boxes by Hungarian matching for tracking bounding boxes. SORT has achieved good results in high frame rates and low occlusion environments. Its successor, DeepSORT (Wojke, Bewley, & Paulus), adopted the same paradigm and improved matching cascade in association by first matching through a ReID network based on feature cosine similarity and then through IoU. Including feature similarity and evaluating the Kalman filter prediction state using Mahalanobis distance so that DeepSORT leads to better performance in occlusion condition and long-term tracking. StrongSORT (Du et al., 2023) built on top of DeepSORT, which adds an exponential moving average (EMA) module to enhance the robustness of the detection noise when preserving long-term information. In addition, StrongSORT mitigates the camera jitter problem by introducing an enhanced correlation coefficient (ECC) maximization module.

In this study, the goal is to develop an end-to-end 2D dynamic counting method for strawberry fruits. The input of the method are videos captured using a cell phone camera, and the output displays accurate counting results and localization information onto the input videos. Specifically, we have three objectives: (1) to develop an improved YOLOv5s network for strawberry fruits detection from videos by integrating a vision transformer layer; (2) to develop a StrongSORT-based dynamic counting algorithm to count the total number of strawberry fruits. (3) to validate the proposed method using the videos collected on an indoor strawberry farm.

Materials and Methods

Infrastructure and Data Collection

The video data collection was conducted on an indoor farm located in Montréal, Canada. The strawberries were grown in densely arranged racks (Figure 1). Each rack measures 2.9 meters high, 3.0 meters long, and 0.6 meters wide, and is composed of five layers of strawberries. We conducted two separate sessions for the data collection on February 9th and March 2nd, 2023, respectively. During the February 9th session, most strawberries were unripe and characterized by a light green color. In contrast, most strawberries observed on March 2nd exhibited ripe, bright red features.

Keys	Values
Number of videos collected on Feb 9th	5
Number of videos collected on March 2nd	5
Average length	20 seconds
FPS (frames per second)	30
Resolution (pixels)	1920 × 1080
Average number of fruits per video	33

Table 1. Video dataset overview

A cell phone (Google Pixel 7 Pro) mounted on a DJI OSMO Mobile 6 gimbal was used for the video data collection (Figure 1). The videos were captured at a resolution of 1920×1080 at 30 FPS with a pace of around 0.15 m/s of the cell phone. A total of ten videos were captured with five videos collected from each session (Table 1). Each video had an average duration of 20 seconds and contained an average of 33 strawberry fruits.

Figure 1. Video data collection at the strawberry farm. A cell phone mounted on a DJI gimbal was used for the data collection.

Methodology

Data Preprocessing

While recording the strawberries, our phone camera inevitably captured some extraneous data that we are not interested in. This extraneous data included the strawberries in adjacent layers, and superfluous information about other strawberry racks in the background. We performed cropping preprocessing to eliminate the irrelevant information on a MacBook Pro running macOS Ventura 13.3.1, utilizing Final Cut Pro as the editing software. While ensuring the completeness of the strawberry information within the captured layer, we cropped redundant information out to the greatest extent possible.

Furthermore, we applied various data augmentation operations to our training dataset, facilitating the object detection network to acquire a more comprehensive understanding of strawberry characteristics. The augmentation methods comprised a broad spectrum of image manipulations such as color and brightness adjustments, image rotation, sharpness enhancement, blending, and conversion to greyscale.

Dynamic Strawberry Counting Framework

The proposed data processing framework mainly consists of two modules (An improved YOLOv5s object detection module and a SrongSORT-based counting module) (Figure 2). First, our dynamic counting method takes in strawberry videos, and through an object detection network, generates location coordinates for strawberry fruits by displaying bounding boxes in the output videos. Then, the SrongSORT tacking network is used to classify whether the strawberries in the bounding boxes of consecutive frames belong to the same individual. Once we have determined strawberry identities, the proposed DynCount algorithm is used to count strawberry fruits. Finally, we output the counting results along with the bounding boxes onto the original videos at the output end.

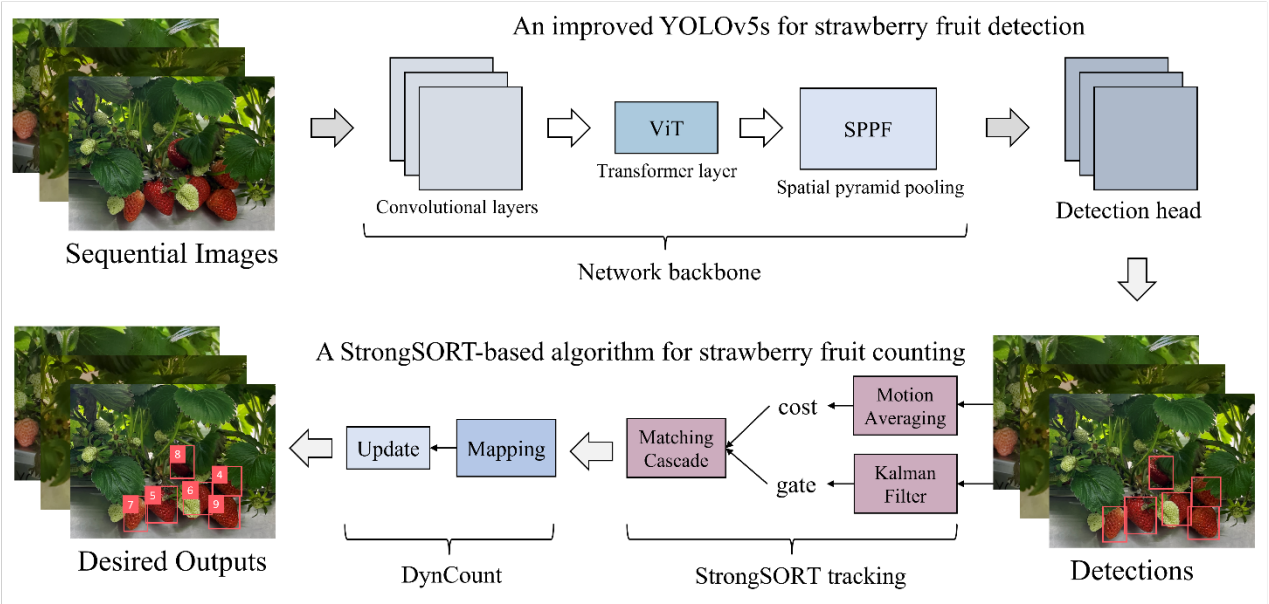


Figure 2. The modular design of dynamic counting for strawberry fruits demonstrates the process that uploads strawberry plant video data into an improved YOLOv5s network to generate bounding boxes for the location information of strawberry fruits. The video data, along with the obtained location information of strawberry fruits in each frame, is then fed into a StrongSORT-based strawberry fruit counting algorithm to acquire and display counting information on top of each bounding box.

Integrating a vision transformer layer into YOLOv5s for strawberry fruit detection

Precise strawberry detection is the prerequisite for fruit tracking and counting. YOLOv5 is a comprehensive and mature solution for object detection, whereas YOLOv5s is one of its backbone networks. However, due to the high standard for production improvement in our farm, our application scenario is complicated as the strawberry plants are densely arranged (Figure 1). Because of the dense canopy, some strawberry identities appeared in different states of being non-occluded, partially occluded, and severely occluded in different frames when capturing videos. Particularly, when the strawberries were unripe and had a light green color, some fruit identities may be omitted even by human eyes when they were obscured. Furthermore, focus loss is also a frequent problem in mobile video filming. With the movement of the lens, some individual frames in the video often experience focus loss and appear blurred.

Although the native YOLOv5s network can achieve a high detection accuracy in non-blurry and non-occluded conditions, we still expect our object detection algorithm to detect strawberry fruits in blurry and occluded frames. To do so, we integrated a vision transformer layer to the original YOLOv5s model for finer feature extraction (Figure 2). In the vision transformer layer, an input image is divided into a fixed number of non-overlapping patches, and each patch is linearly projected into a lower-dimensional embedding space to obtain a sequence of patch embeddings. The multi-head self-attention mechanism calculates a set of attention weights for each embedding based on its similarity to other embeddings in the sequence. In short, the vision transformer captures long-range dependencies in the image, increasing the confidence of detected objects that span multiple regions, thus increasing detection accuracy.

Mapping Track IDs from StrongSORT network to strawberry fruit counting numbers

In our pipeline, StrongSORT utilizes the improved YOLOv5s network to obtain a set of bounding boxes with corresponding confidence scores in each frame of the video. Once objects (strawberry fruits) are detected in the current frame, a Kalman filter is used to estimate the position and velocity of each object in the next frame. The Kalman filter incorporates the previous position and velocity of the object, the current bounding box detection, and the motion model of the object to estimate the new position of the object. The Hungarian algorithm is then employed to find the optimal matching between the detected objects and the estimated object positions based on their overlap and distance. After matching the objects, the data association is performed to link the tracks of the objects over time. It ensures that the same object is tracked over multiple frames by linking the detections in consecutive frames to form tracks. To ensure that the algorithm does not lose track of objects even in the presence of missed detections in some frames, StrongSORT assigns a unique track ID to every newly detected object and preserves all track IDs across all frames. In addition to track IDs, each newly detected object is given a tentative state label. As the video data plays, these object state labels will change to confirmed or deleted over time in the updates of each frame image, but track IDs remain unchanged.

Our DynCount algorithm takes advantage of track IDs and their associated state labels and then transforms them into

corresponding counting numbers. Once we obtain the confirmed target array, we can create a mapping function for these targets (Algorithm 1). In order to demonstrate the advantages of our DynCount algorithm over the native StrongSORT algorithm, we propose to write the confirmed IDs of StrongSORT and output results of DynCount, along with their corresponding bounding boxes, into the original video and conduct a comparison. To better illustrate the performance of DynCount, we select one frame from both StrongSORT and DynCount results in the same scene (Figure 3). As shown in the figure, the output of StrongSORT is a series of independent track IDs, rather than a continuous count of the targets, which is the number of strawberries in our case. Given that each frame in the video and each strawberry detected within a frame are independent of each other, StrongSORT distinguishes different strawberries by assigning them unique track IDs. The DynCount algorithm is designed to address the issue of repeated detections of the same strawberry in consecutive independent frames by updating the results of StrongSORT (Algorithm 1), and thus obtaining a dynamic count of the targets of interest. Specifically, we map the confirmed track IDs in Figure 3(a) (TrackID: 4, 6, 11, 12, 14, 15) to a precise count of six through eleven (Figure 3(b)), respectively.

(a) StrongSORT output (b) DynCount output

Figure 3. (a) Track IDs generated by the StrongSORT algorithm, (b) counting results output from the DynCount algorithm.

As shown in Algorithm 1, our mapping scheme extends the counting array range by one when there comes a new confirmed object. The main idea of this mapping method is to record and count the number of confirmed objects, rather than recording the sequence number of objects in all states. However, a mapping function alone is not sufficient to run the code, as the information from the previous frame and the current frame is different in video data. In our implementation, our DynCount array is updated in conjunction with the StrongSORT algorithm. If the identity of a strawberry in the next frame is already present in our array, it is not counted again. Otherwise, we store the track ID of the strawberry and its corresponding count in our DynCount array. At the output stage, when we need to retrieve the count for a specific strawberry, we directly retrieve the corresponding count number from the DynCount array based on its track ID.

Algorithm 1 A dynamic mapping scheme that projects tracking IDs onto counting numbers.

```
procedure DynMap(confirmed, buffer, reference)
    for id in confirmed do
        if id not in buffer then
            buffer.add(id)
    for id in buffer do
        if id in reference then
            continue
        count ← max(reference.values(), 0) + 1
        reference[id] ← count
    return reference
```

Algorithm 1. This algorithm performs a one-to-one mapping of the track IDs generated by StrongSORT to the proposed dynamic counting numbers.

Network Training and Validation

The algorithm development and validation were conducted on the Ubuntu 22.04 LTS operating system, and all the tasks from training to counting were carried out on a GeForce RTX 3060Ti graphics card. Thanks to the advantage of transfer learning (Zhuang et al., 2021) and the abundance of open-source datasets available online, we did not carry out any annotation work on our training dataset. Instead, we labeled a total of 40 images taken on February 9th using labelImg

(Tzutalin, 2015), with 20 images each reserved for the validation set and testing set. Our training set consists of a total of 440 images that have undergone data augmentation, and we published our training dataset at the following link: https://github.com/Xuehaiz/StrawberryDataset.git.

We validate the performance of the network through the loss function and mean Average Precision (mAP). During the training of YOLOv5, the model is optimized by minimizing the mean square error (MSE) loss function (Zhou & Bovik, 2009) between the predicted and ground truth bounding boxes' coordinates and object scores. The validation loss is used to evaluate the performance of the model during training. It is calculated by the same loss function as the training loss but applied to a separate validation set. The validation loss helps monitor the model's generalization ability and avoid the overfitting of training data. On the other hand, the metric of mAP is frequently employed in assessing the efficacy of object detection models. It computes the average precision of the model for all classes under specific IoU thresholds. In this study, we conducted an evaluation of the performance of the model on IoU of 0.5 and a spectrum of IoU thresholds ranging from 0.5 to 0.95. Generally, the utilization of mAP at IoU of 0.5 is more prevalent, while obtaining high scores on mAP at IoU values within the range of 0.5 to 0.95 is deemed more challenging.

Results and Discussion

Network Evaluation

The trained object detection network can accurately and quickly identify ripe and unripe strawberry fruits. We set the confidence threshold of our detection to 0.24, and the IoU threshold to 0.18. When the batch size was set to 16, our network's training time was less than 13 minutes.

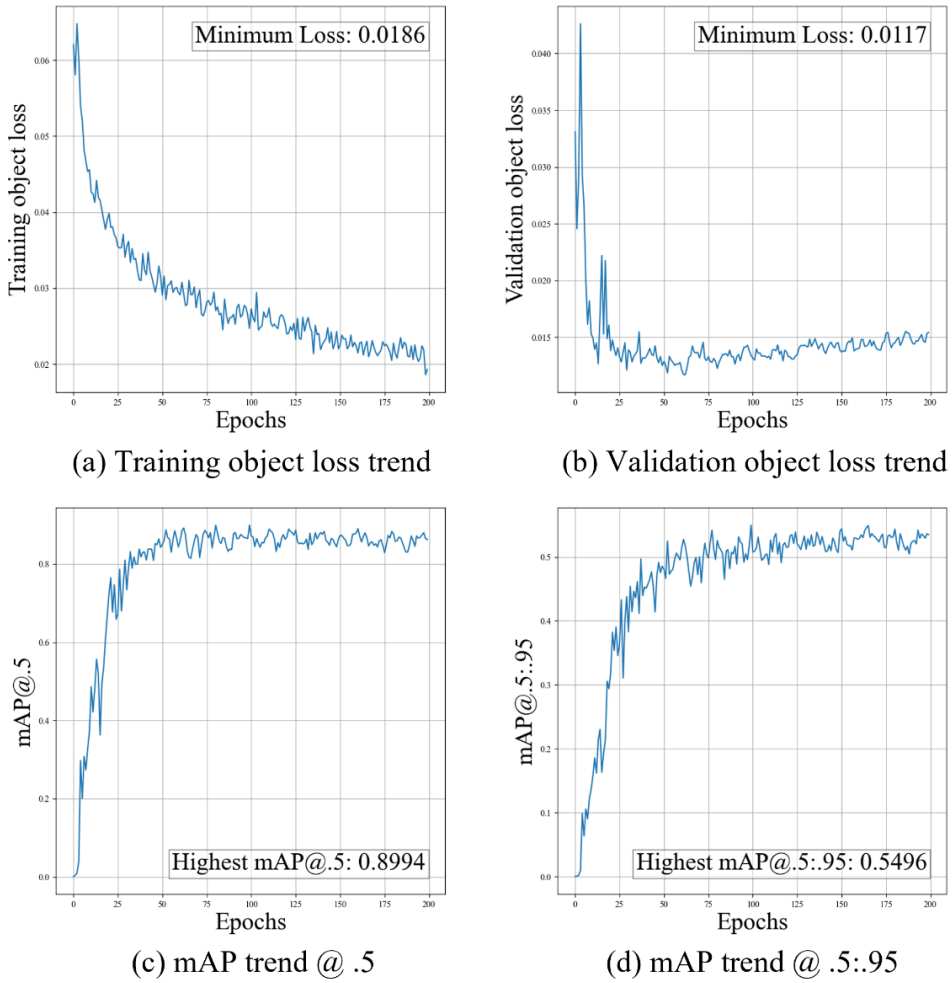


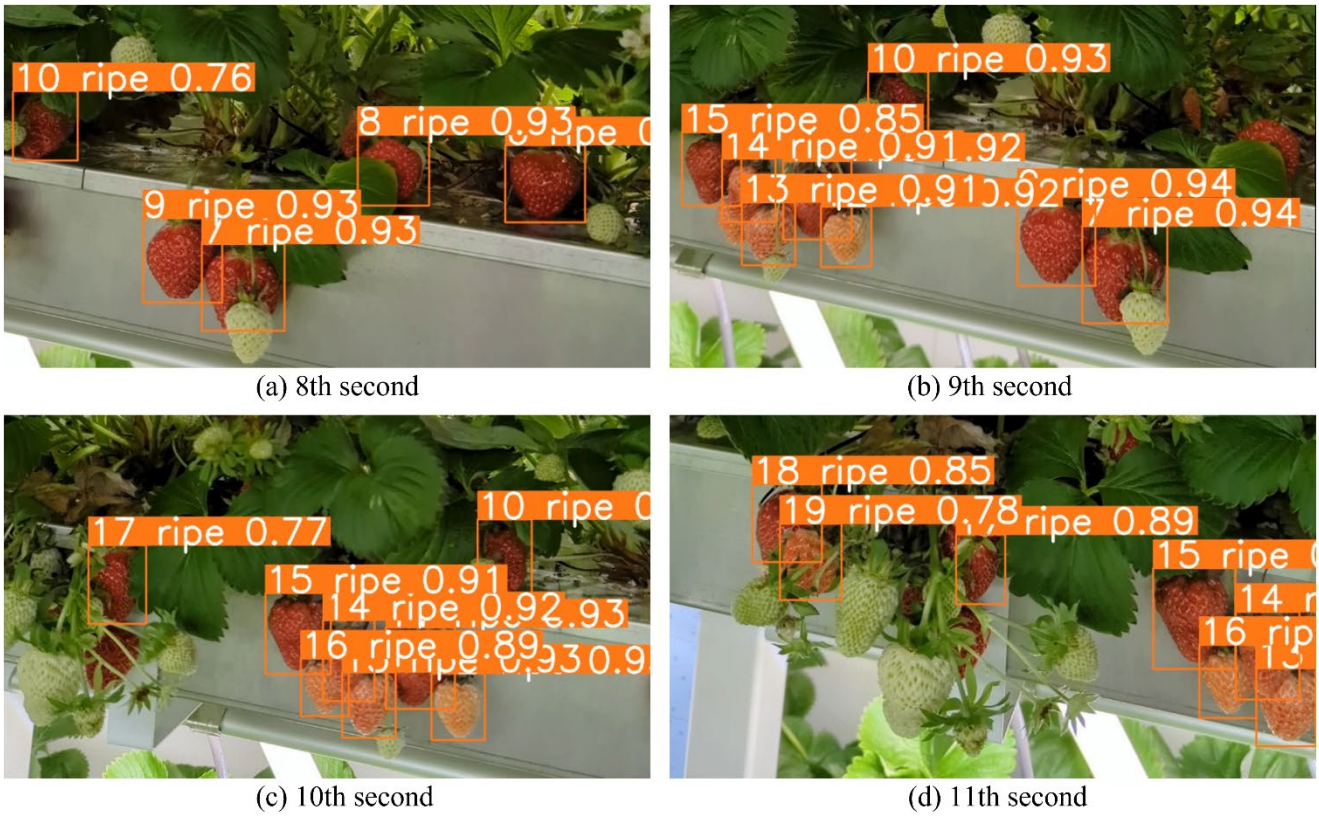
Figure 4. Over the course of 200 epochs of training, (a) and (b) illustrate the training and validation loss, while (c) and (d) depict the evolution of the mAP at 0.5 and 0.5:0.95, respectively.

We present the training loss and mAP trends for 200 epochs (Figure 4). The utilization of data augmentation techniques resulted in a substantial enhancement of the network's performance, validating the effectiveness. Figures 4(a) and 4(b) illustrate the training and validation loss, respectively, over 200 epochs. A noticeable overfitting pattern begins to emerge in the validation set around the 60th epoch, as the training loss continues to decrease. Therefore, we set the number of training epochs to 75 to achieve the best model performance in our dataset. By examining the validation loss, we find that the object loss reaches the lowest level of 0.0117 and oscillates around the level of 0.015. Thus, we can safely conclude that the improved object detection network, YOLOv5s with vision transformer, outperforms human performance in object detection tasks. Figures 4(c) and 4(d) show the mAP trend of the model under IoU thresholds of 0.5 and in the interval of 0.5 to 0.95 for 200 epochs. We recorded the highest mAP at 89.94% under the IoU of 0.5 and the highest mAP at 54.96% under the IoU interval of 0.5 to 0.95.

Output Demonstration and Interpretation

Representative detection and counting results for strawberry fruits are presented in Figure 5. To display the temporal information of the video, we selected one frame from each of the eighth, ninth, tenth, and eleventh seconds in which there was a large cluster of strawberry fruits to be more representative. To better illustrate our results, we presented the position information of the strawberries in bounding boxes, along with the counting numbers of fruit, the classification of the fruit, and the confidence output of the detection algorithm from left to right overlaid onto the original video. The camera moved from right to left during the video collection.

Overall, the proposed method achieved good performance for strawberry fruit detection and counting. However, it was observed that the heavily occluded object might not be detected. For example, the algorithm identified the eighth strawberry in the eighth second, but due to changes in the camera angle and the result in heavier leaf occlusion, the same strawberry in the ninth second was not identified. Fortunately, this did not affect the accuracy of the counting, since an eighth strawberry had been counted before. However, in some similar cases, when the camera angle changes again and the same strawberry reappears under the detection network, the tracking algorithm may mistakenly identify the reappearance as a new strawberry, which is one of error sources for the counting. In the tenth and eleventh seconds, there was another example of a heavily occluded strawberry underneath the seventeenth strawberry. Lowering the confidence threshold is a simple way to increase the probability of the algorithm detecting this severely occluded and shadowed strawberry. However, this comes with a trade-off that may result in more false positives. Therefore, one of future works is to address the balance between false positives and false negatives to further improve the counting performance.



(a) 8th second (b) 9th second

(c) 10th second (d) 11th second

Figure 5. Four representative frames selected from four consecutive seconds to illustrate the detection and counting results in the randomly chosen output video. The camera moved from right to left during the video collection.

Developed Method Analysis

Figure 6 illustrates the comparison between the developed counting method and the ground truth. Two people conducted manual counting of the fruits independently and averaged their results to establish the ground truth. The resulting ground truth served as the reference point for our algorithmic counting. We conducted our analysis by targeting unripe strawberries in the February 9th dataset and ripe strawberries in the March 2nd dataset, since February 9th dataset consisted primarily of unripe strawberries while the March 2nd dataset was comprised mostly of ripe strawberries. The results of this comparison provide a clear representation of the performance of our method and the degree of its alignment with the actual counts.

In Figure 6(a), the deep yellow data points represent the strawberry data collected on February 9th, while the deep green data points represent the data obtained on March 2nd. The comparison between the ground truth and algorithmic counting results demonstrates that our dynamic counting approach produced highly consistent results with the ground truth. To further evaluate the performance of our method, we computed the counting differences as a percentage between the ground truth values and the results obtained from our dynamic counting method (Figure 6(b)). The counting differences as a percentage were calculated by dividing the dynamic counting results by the ground truth values and rounding up to the nearest 5% interval. The actual counting differences achieved by the proposed method ranged from 87.50% to 112.77% (the percentage greater than 100% means that the algorithm overestimates the number of strawberry fruits). By calculating the absolute error of our counting results, we can obtain an arithmetic mean accuracy of 92.05% for ten videos. Our results demonstrate that our method achieves good counting performance even under our complex farm environment.

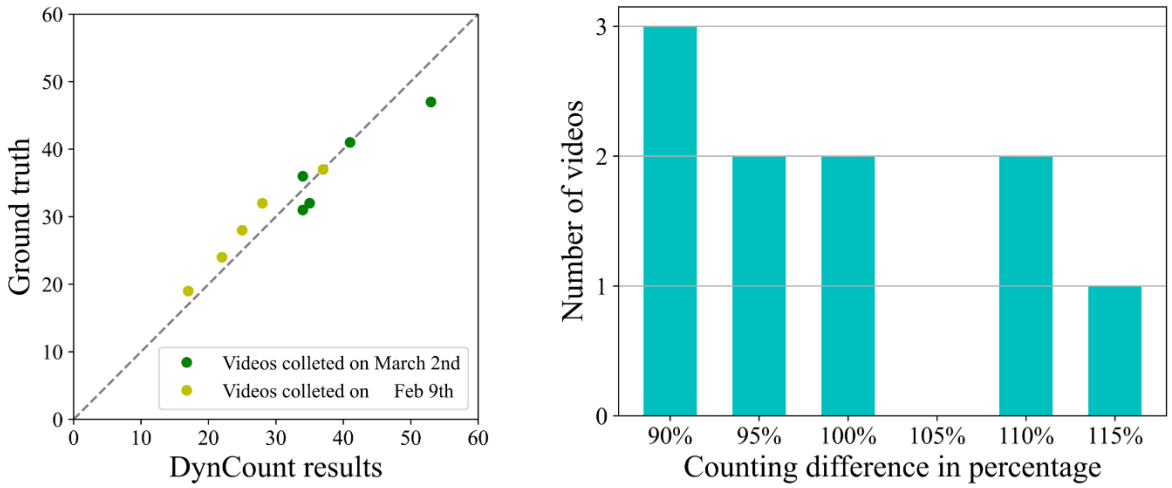


Figure 6. Performance evaluation for the proposed dynamic strawberry fruit counting method. (a) Regression analysis between the video-based counting results and the ground truth. (b) The counting difference between the counting results and the ground truth, rounded up to the nearest 5% interval. The ratio exceeding 100% indicates overestimation.

Discussion

High quality raw data plays a crucial role in our experiments, and there is still much room for improvement. Although a gimble was used to collect as stable as possible video data to the best of our effort, we still encountered inevitable shaking problems while walking. Moreover, we found that the video often experienced frequent focusing problems during filming. Additionally, ensuring a uniform speed during shooting is challenging but essential for the fine-tuned model to interpret the farm scenes accurately. Applying a UAV platform with an integrated camera is a potential solution to address the issues of focusing and inconstant motion during video collection. Additionally, it can significantly improve the efficiency of data collection, which is particularly useful for large-scale farm applications. Furthermore, combining continuously captured photos into a video could be another option to solve the errors caused by focusing problems.

Based on the testing results, misidentification, missed detection, and re-identification are three primary factors that cause a decline in accuracy. As the camera moves, the angle at which the strawberries appear in the lens changes, and if they also appeared occluded by leaves at that moment, the algorithm tends to re-identify them as new strawberries when they reappear in the lens. Additionally, compared to ripe fruits, unripe ones have a higher possibility to be missed due to the color and leaf occlusion. In future study, we are going to collect more data and attempt to develop a regression model to compensate for these factors and enhance the robustness of the algorithm.

Conclusions

Based on a YOLOv5 deep learning network and the self-attention mechanism of the vision transformer, the proposed object detection network achieved a maximum mAP accuracy of 89.94% at an IoU threshold of 0.5 for strawberry fruit detection. And the proposed StrongSORT-based dynamic counting algorithm achieved an average accuracy of 92.05% on the ten strawberry videos. These promising results demonstrate the potential of our approach for accurate and responsive yield prediction. In the future, we aim to optimize our algorithm further to reduce false cases and to gather more data to validate its robustness.

Acknowledgements

This work was supported by Homegrown Innovation Challenge-Spark Award project and FRQNT & MAPAQ Partnership Research Program-Sustainable Agriculture.

References

Bewley, A., Ge, Z., Ott, L., Ramos, F., & Upcroft, B. (2016). *Simple online and realtime tracking*.

Du, Y., Zhao, Z., Song, Y., Zhao, Y., Su, F., Gong, T., & Meng, H. (2023). StrongSORT: Make DeepSORT Great Again. *arXiv pre-print server*. doi:None

arxiv:2202.13514

Han, K., Wang, Y. H., Chen, H. T., Chen, X. H., Guo, J. Y., Liu, Z. H., . . . Tao, D. C. (2023). A Survey on Vision Transformer. *Ieee Transactions on Pattern Analysis and Machine Intelligence, 45*(1), 87-110. doi:10.1109/tpami.2022.3152247

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). *Mask r-cnn.* Paper presented at the Proceedings of the IEEE international conference on computer vision.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, Oct 08-16). *SSD: Single Shot MultiBox Detector.* Paper presented at the 14th European Conference on Computer Vision (ECCV), Amsterdam, NETHERLANDS.

Meimetis, D., Daramouskas, I., Perikos, I., & Hatzilygeroudis, I. (2023). Real-time multiple object tracking using deep learning methods. *Neural Computing & Applications, 35*(1), 89-118. doi:10.1007/s00521-021-06391-y

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., & Ieee. (2016, Jun 27-30). *You Only Look Once: Unified, Real-Time Object Detection.* Paper presented at the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA.

Ren, S. Q., He, K. M., Girshick, R., & Sun, J. (2015, Dec 07-12). *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.* Paper presented at the 29th Annual Conference on Neural Information Processing Systems (NIPS), Montreal, CANADA.

Segui, S., Pujol, O., Vitria, J., & Ieee. (2015, Jun 07-12). *Learning to count with deep object features.* Paper presented at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA.

Sun, S. P., Li, C. Y., Chee, P. W., Paterson, A. H., Jiang, Y., Xu, R., . . . Shehzad, T. (2020). Three-dimensional photogrammetric mapping of cotton bolls in situ based on point cloud segmentation and clustering. *Isprs Journal of Photogrammetry and Remote Sensing, 160*, 195-207. doi:10.1016/j.isprsjprs.2019.12.011

Tzutalin, D. (2015). LabelImg. *GitHub repository, 6*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., . . . Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems, 30*.

Wojke, N., Bewley, A., & Paulus, D. (2017). *Simple online and realtime tracking with a deep association metric*.

Zhou, W., & Bovik, A. C. (2009). Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures. *IEEE Signal Processing Magazine, 26*(1), 98-117. doi:10.1109/msp.2008.930649

Zhuang, F. Z., Qi, Z. Y., Duan, K. Y., Xi, D. B., Zhu, Y. C., Zhu, H. S., . . . He, Q. (2021). A Comprehensive Survey on Transfer Learning. *Proceedings of the Ieee, 109*(1), 43-76. doi:10.1109/jproc.2020.3004555