Apple Detection and Tracking Using Deep Learning Based Computer Vision Techniques

Rishabh Pahuja
Graduate Research Assistant
Department of Mechanical Engineering
Carnegie Mellon University

Dr. Abhisesh Silwal
Senior Project Scientist
Robotics Institute
Carnegie Mellon University

Dr. George A. Kantor
Research Professor
Robotics Institute
Carnegie Mellon University

Abstract- *We aim to develop a software pipeline that can detect and track apples in real-time in an orchard using deep learning approach, while effectively dealing with cases of apple occlusion. The center of each segmented apple is projected to the world frame using stereo reconstruction, and the apple centers are tracked using Kalman Filter. The combined use of segmentation and detection ensures that only the center of the apple is projected in the world coordinate system, while avoiding any undesired nearby objects in case of any occlusion. The Extended Kalman Filter enables consistent tracking of objects during occlusion, facilitating reassociation of apple tracking ID in case an apple is not detected for a few frames. We have successfully implemented all the steps in the pipeline and have shown that our pipeline can accurately detect and track all the apples in the orchard using real-time data collection, and also provide an estimated yield for farmers to make informed decisions about orchard management and optimize their harvest. To further improve accuracy, we aim to utilize targeted and controlled tree foliage agitation through airflow by directing compressed air streams to occluded apples.*

Fig. 1: The blue dotted line shows the trajectory followed by each apple w.r.t the rover

1 Introduction

Fruit growers all around the world are experiencing labour shortage due to diminishing interest of the workforce in agriculture. The problem was exacerbated by recent international travel restrictions in pandemic conditions which have limited availability of skilled migrant workers [1]. Therefore, there exists a requirement to evaluate alternate methods to manual picking.

Fruit picking robots have been developed since 1960s, however they have not been very effective even if the harvesting speed is low. A large share of fruits remain un-handled due to the poor quality of the machine vision systems [2]. A robot harvester has two components, 1. the physical robot with an end effector that shall pluck fruit, 2. perception pipeline that localizes the fruit so that they can be plucked effectively by the robot. In this work we are trying to build the vision system for an apple harvester robot that automatically plucks apples in an orchard and gives an estimate yield. Accurate pre-harvest estimations of fruit load per tree can support more informed decisions regarding harvesting decisions of labour, equipment, packaging as well as handling [3].

Pre-harvesting is usually done by manual counting of labour. However, given shortage of manual farm labour along with the large numbers of apple trees in an orchard, the estimate of fruit yield by human labour is impractical. Numerous studies have been carried out over the years to detect, count and localize fruits using an automated machine vision systems. When a fruit color is distinct from the background, a simple segmentation using color features can be

used. Zhou et al. [4] used red colour features in the detection of mature 'Gala' apples while Zaman et al. [5] used color features to estimate blueberry yield. There has been research done to give attention to the fruit shape and texture for detection. Work has also been done to extract feature descriptors for regions of interest using Histogram of Oriented Gradients (HOG) [6], Scale Invariant Feature Transform (SIFT) [7] or Speeded Up Robust Features(SURF) [8]. But these methods are not very robust since they are directly dependent on the light conditions.

Deep learning based methods have proven to be most robust for object detection. Chen et al. [9] used a Fully Convolutional Network [10] to estimate orange and apples, Bargoti and Underwood [11] used Faster-RCNN and transfer learning to estimate the yield of apple, mango, and almond orchards, and Sa, et al. [12] used Faster-RCNN (Faster Region-Convolutional Neural Networks) algorithm to detected multi-coloured (green, red,or yellow) capsicum fruit. However, detection alone cannot be used to localize and estimate the fruits as this could lead to: (1) double counting the same fruit which appeared in consecutive frames, (2) double counting of fruit which was tracked across several frames but was not detected for a few frames due to occlusion or detection inaccuracy and then detected again, (3) miscounting the newly detected fruit as a previously counted fruit since it was very close to the tracked fruit (4) inaccurate fruit localization as there can be a lot of unwanted information around it.

A lot of work has been done to localize fruits and count fruits using detection algorithms. In order to achieve the efficient and reliable picking of fruits, detections need to be localized. Different methods based on different cameras have been used for the localization of fruits and other agricultural crops. These include the use of stereo cameras, depth cameras or single camera with extra assumptions. Itakura et al. used Kalman filter with YOLO to count fruits [13]. Many works has been done to perform detection and localization of fruits using RGB-D cameras, however, Longsheng Fu et al. performed a literature survey for the application of RGB-D cameras for fruit detection and localization and came to the conclusion that the biggest advantage of RGB-D cameras is their low cost. However, this camera has comparatively low resolutions of depth. [14]. Xu Liu used a version of DeepSort [15] along with Structure from Motion (SFM) to get an estimated apple yield from an orchard and utilized hungarian algorithm to perform association of same apples in consecutive frames [16]. Tian-Hu Liu et al. proposed using modified version of YOLOv3 model for pineapple detection and binocular stereo vision for localization [17]. Yu et al. [18] utilized Mask R-CNN for strawberry detection and similarly, Gonzalez et al. [19] used the same network for blueberry detection. Mehta and Burks [20] localized citrus fruits using a fixed monocular camera. Xiong et al. [21] used a single RGB camera for weed localization, based on the assumption that the distance between the camera and the weed plane was fixed. Detection and segmentation networks have been widely used for the detection and counting of fruit, and their applications in fruit harvesting have recently been.

This work introduces a pipeline that uses YOLOv8 to detect apples in every frame. We use a modified version of Simple Online Realtime Tracking (SORT) [22] to prevent counting the same apple more than once. A deep learning based network architecture called U-Net [23] is used with YOLOv8 to segment out apples from the background and project the apple centers to world coordinate frame using a stereo camera setup. The entire pipeline can be seen in Figure 2. The implementation of our pipeline can be found on the link: *https://github.com/rishabhpahuja/Apple-Tracking*

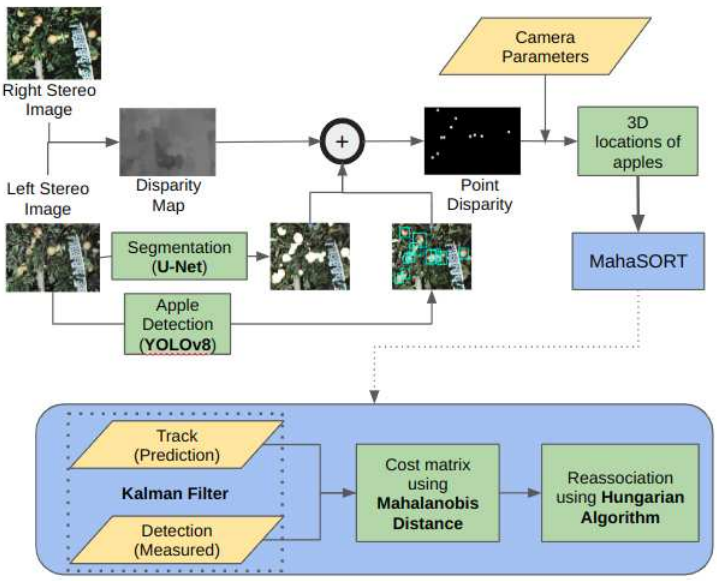


Fig. 2: Our fruit counting and localization pipeline consists of two major components. First component is to detect apples, remove unwanted information around the fruit, and project the fruits in world co-ordinate frame. U-Net segmentation and YOLOv8 detection are used in unisonto find the apple center. The left and right stereo images are used to find disparity map using RAFT stereo which along with apple centers are used to localize apple location w.r.t. the world coordinate frame. Second component is to track the apples in world coordinate frame, which is performed using MahaSORT, our implementation of the modified version of SORT [22]

2 Materials and methods

2.1 Data acquisition

The data was collected in an orchard in California using a robot with a stereo camera setup using an illumination invariant camera system [24]. The rover acts as a Real-Time Kinematic (RTK) rover with respect to a RTK base, recording the location of the rover. The data consisting of the video frame, camera intrinsics and rover coordinates with respect to the RTK base are stored in a ROS bag. The same color circles in two frames at different time instances around the fruit indicate the same fruit (shown in Figure 3) and which explains why tracking is important to localize and estimate the yield.

(a) (b)

Fig. 3: (a) Image at T=1 (b) Image at T=2; Frames at different instances having the same apples

2.2 Image Processing: Eliminating Unwanted Background Elements

One of the challenges in this entire pipeline was to eliminate unwanted elements around the apples. We solved this problem by using an intersection of object detection and semantic segmentation. We employed the YOLOv8 algorithm to detect apples and draw bounding box around them. YOLOv8 (You Only Look Once) detection algorithm is the current state of the art detection algorithm. It achieves an average precision of 53.9% with an image size of 640 pixels (compared to 50.7% of YOLOv5 on the same input size) with a speed of 280 FPS [25]. To make the pipeline run faster, U-Net segmentation is performed on the entire image. An intersection of bounding box and segmentation mask is taken to segregate all the apples and find their respective centers in the 2D image. This step is important since we need the exact location of the apple to be plucked by the robot, and the apple might be covered by neighbouring leaves, branches or even surrounding apples, we find the center pixel of each apple and project it in the world coordinate frame. Figure 4 shows the step-wise process to extract apple centers.

This entire process could have been performed by using an architecture like Mask-R CNN. However, it was imperative to use detection followed by segmentation because the labelled dataset for segmentation and detection were on different image sequences.

2.3 Projecting Apple Centers to World Coordinate Frame

In the previous section, the apple centers were identified, but in order to track them, they needed to be projected onto the world coordinate frame. To achieve this, it was necessary to obtain a disparity map between the left and right stereo images. We tried two ways to find to find the disparity map- 1. OpenCV method's of StereoSGBM, 2. RAFT-Stereo [26], which is a deep learning based approach to find disparity map. We used RAFTStereo to infer the stereo map as it does not require fine tuning and gives more accurate results. The figure 5 shows the disparity map for a left and right stereo image.

The disparity map is cropped to a point disparity map using the apple centers found in figure 4. The apple centers are projected to the camera coordinate frame using camera intrinsic matrix. We project the complete image onto the camera coordinate frame (as shown in figure 1), with the purpose of conducting a rigorous and systematic evaluation of the data. This step is undertaken as a means of verifying the accuracy and reliability of the calibrated camera.

(a) (b)

(c)

Fig. 4: (a) Bounding Box Detected by YOLOv8 (b) Segmented Apple Pixels by U-Net (c) Apple Centers (indicated in red)

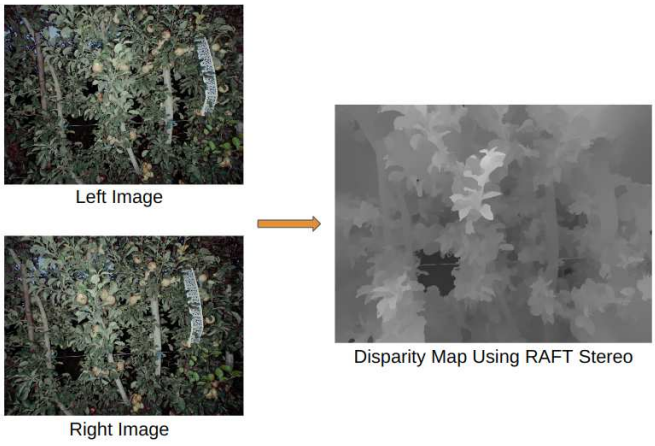


Fig. 5: Disparity map found using RAFT Stereo [26]

2.4 Apple Tracking and Counting

2.4.1 MahaSORT

Apple tracking and counting was performed by using a modified version of Simple Online Realtime Tracking (SORT) algorithm [15]. Figure 8 explains the overall

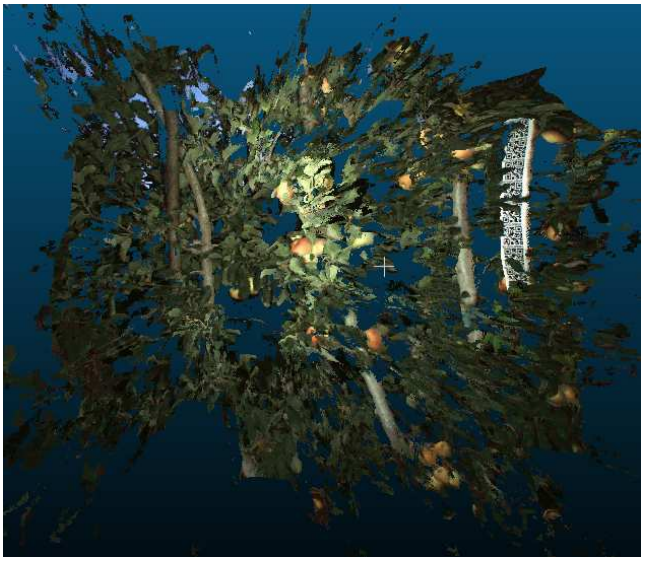


Fig. 6: Point Cloud of a Single Frame Showing Unwanted Information Around the Apples

pipeline of SORT algorithm. However, utilizing Intersection Over Union (IOU) as an association metric may not be very effective. This is because the IOU metric may fail when the rover moves with an irregular velocity, resulting in situations where the same apple will have a very low IOU score, leading to incorrect associations. To address this issue, we replace the IOU metric with a new metric called Mahalanobis Distance. The Mahalanobis distance is a measure of the distance between a point and a distribution, providing information about potential object locations based on motion. In our particular use case, the distribution will be the predicted location of the fruit for the next time step, while the point will be the fruits detected in the next time step.

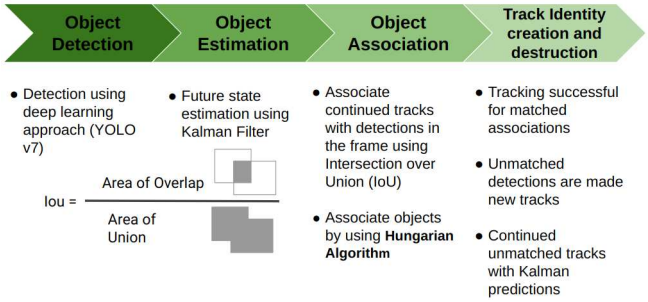


Fig. 7: Simple Online Realtime Tracking [15]

2.4.2 Kalman Filter for object tracking

Our proposed pipeline utilizes a single Kalman filter to track both the rover and the apples relative to the rover's starting point. This approach provides numerous benefits, as it ensures that the coordinates of the same apple in different frames remain consistent with respect to the world origin. Rover speed variation, minor image scaling and rotation variations were taken as process noise. The rover states were taken as $\mathbf{p} = [\mathbf{x}, \mathbf{y}, \mathbf{z}]^{\mathbf{T}}$ and apple centres were taken as $\mathbf{m} = [\mathbf{m_x}, \mathbf{m_y}, \mathbf{m_z}]^{\mathbf{T}}$. Since the rover and apples were tracked using one Kalman filter, the states become:

$$x = \begin{bmatrix} p \\ m_1 \\ m_2 \\ \vdots \\ m_n \end{bmatrix}_{(3n+3)X1} \tag{1}$$

where n denotes the number of apples detected in a frame

Our tracking algorithm employs a state vector that includes only the position of the objects being tracked, without any velocity information. As the coordinates of the apples remain constant across frames, while only the rover's coordinates change, we define the transition state matrix **F** and measurement matrix **H** as follows:

$$F = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \end{bmatrix}_{3X(3n+3)} \tag{2}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 1 & \cdots & 0 \end{bmatrix}_{(3n+3)X(3n+3)} \tag{3}$$

We assumed a constant motion model, where velocity variations were taken care by process noise and measurement noise. The Kalman filter has two major steps- prediction and update step. During the prediction step, the estimate states are predicted for the next time step.

$$x_{k+1|k} = F x_{k|k} \tag{4}$$

where $x_{k|k}$ is the previous location of the fruit and $x_{k+1|k}$ indicates the predicted state for the next time step using the previous state. In the same time step, error covariance matrix is also predicted.

$$P_{k+1|k} = F P_{k|k} + Q \tag{5}$$

where Q is the covariance of the process noise. The Kalman gain is a key parameter which determines the degree of influence assigned to the measurement and prediction steps. Specifically, it provides a measure of how much

weight should be given to each of these steps when updating the state estimate of the objects being tracked.

$$K_k = P_{k+1|k}H^T(HP_{k+1|k}H^T + R)^{-1} \tag{6}$$

where R is the covariance of the measurement noise. Given the new measured location z_k, the new state of the fruit is estimated using:

$$x_{k|k} = x_{k|k-1} + K_k(z_k - Hx_{k|k-1}) \tag{7}$$

And the error covariance matrix is also updated using:

$$P_{k|k} = (I - K_kH)P_{k|k-1} \tag{8}$$

where **I** denotes identity matrix

However, before updating the state estimates of the objects being tracked, accurate associations between tracks (i.e., previously detected apples that have been successfully associated with their tracks) and new detections must be established. To accomplish this, we first construct a cost matrix that assigns costs between each track and new detection pair, utilizing the Mahalanobis distance metric. By calculating the Mahalanobis distance between the predicted location of each track and the location of each new detection, we can determine the optimal association between the two sets of observations. This information is then used to update the state estimates of the objects being tracked in a robust and accurate manner.

$$C_{i,j} = (d_j - y_i)^T S_i^{-1}(d_j - y_i) \tag{9}$$

where, $(d_j - y_i)^T(d_j - y_i)$ indicate the euclidean distance between predicted tracks and detections while S_i is the error covariance matrix.

The aim of the Hungarian algorithm is to find the minimum total cost of assignment. This can be written as:

$$min\Sigma_{i=1}^{m}\Sigma_{j=1}^{n}C_{i,j}.x_{i,j} \tag{10}$$

where m and n are numbers of tracked and new fruit.

To determine the optimal assignment between fruit detections in consecutive frames, we utilized the Hungarian Algorithm to calculate the minimum total cost. If a fruit in the first frame was successfully associated with a fruit in the second frame, it was considered to be tracked and its location in the second frame was used to update the Kalman filter. In cases where a fruit detection in the second frame could not be assigned to a fruit in the first frame, we used the predicted location from the Kalman filter as the new location for that fruit and increased a corresponding counter by one. Any unassigned fruit in the second frame was identified as a new tracked fruit, and its states were added to the Kalman filter. By incorporating this process, we are able to maintain an accurate and consistent estimate of the objects being tracked across multiple frames.

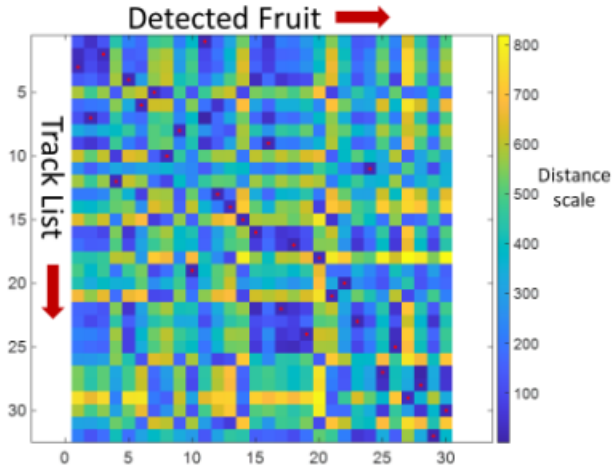


Fig. 8: Hungarian Assignment. X axis indicates new fruit while Y axis indicates tracked fruit (previously detected fruit)

3 Results and Discussion

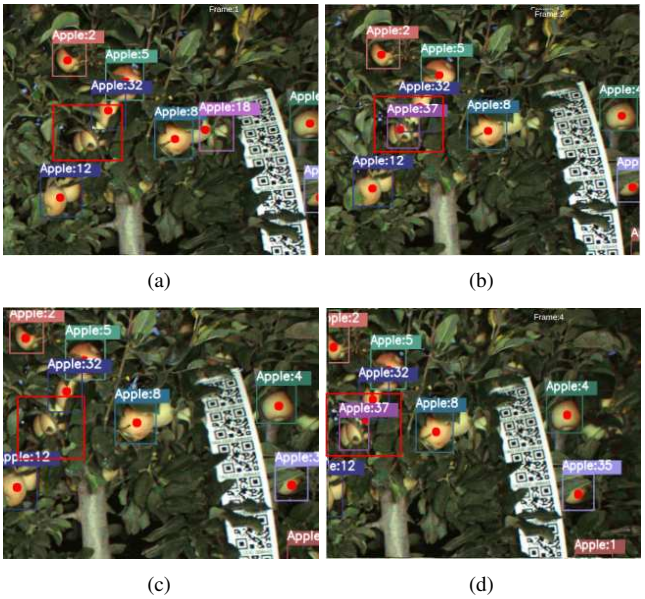
3.1 Fruit Detection

The apple detection model using YOLOv8 was trained on images collected from an orchard. When it was tested on images from the same orchard, the model was able to achieve an average recall of 0.87 and an average precision of 0.94 over all the frame sequences. Figure 9 shows the detection output by YOLOv8. Our evaluation of the detection results indicated that the detector successfully detected the majority of the apples with high accuracy, as evidenced by the minimal number of false positives and the absence of detection of occluded apples. The visual results were conformed with the reported average precision and recall values. Since all the apples are not detected it results in higher false negatives and thus, lower recall value. This can be seen from the formulae in equations 11 & 12.

$$Precision = \frac{TruePosition}{TruePositive + FalsePositives} \tag{11}$$

$$Recall = \frac{TruePosition}{TruePositive + FalseNegatives} \tag{12}$$

Fig. 9: Image showing the apple detections by YOLOv8



(a) (b)

(c) (d)

Fig. 10: (a) Apple 37 (shown in red box) was not detected (Frame 1) (b) Apple 37 was correctly associated (Frame 2) (c) Apple 37 was not detected (Frame 3) (d) Apple 37 was correctly associated (Frame 4)

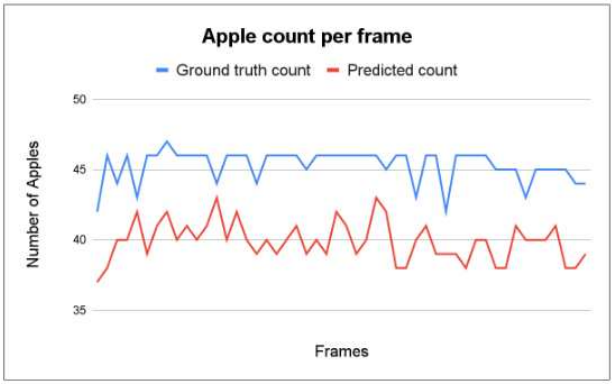


Fig. 11: Comparison between the true count and the estimate count of the pipeline

3.2 Fruit Tracking

Figure 10 shows the tracking output of our pipeline. In consecutive frames, the same apple is consistently assigned an apple ID, indicating that the Mahalanobis distance between corresponding detections was low and the Hungarian Algorithm successfully assigned the correct ID to the apples in the consecutive frames. One particularly interesting result is the case of apple ID 37, which was not detected in frame 1, detected in frame 2, not detected in frame 3, and detected again in frame 4. Despite being undetected and detected again, the apple was still given the correct ID. When the apple was first detected in frame 2, it became a track and its location was predicted for frame 3 using the Kalman filter. However, since there was no detection in frame 3, the location could not be updated. In frame 4, when the apple was detected again, it was associated with the earlier prediction and the location was updated using the update step of the Kalman filter.

The number of apples in the video segment was manually counted and compared with the output of the pipeline. Figure 11 illustrates the comparison between the manual count and the apples estimated by the pipeline. As shown in the figure, the estimated number of apples is consistently lower than the ground truth number of apples. This discrepancy is attributed to the fact that some apples were not detected by the YOLO detector, likely because they were occluded by surrounding leaves and branches. It is worth noting that the graph depicted in Figure 11 is consistent with the precision and recall values. This is because lower estimate yield means higher false negatives, which attribute to lower recall values.

4 Conclusions

This study introduced a very simple yet effective method to localize and track fruits in an orchard. Video based Kalman tracking was demonstrated to provide an improved estimate of total fruit load. The apple count results are very close to the actual count as indicated by 11. The difference in count is due to the fact that the occluded apples are difficult to detected. Our team aims to improve the results by building a agitated system that uses a compressed air system to improve the visibility of occluded fruits.

5 Future Work

It was observed that the apple centres found by unison of detection and segmentation, in certain cases, can be located on a leaf that is cutting through the fruit's center, resulting in inaccuracies. Furthermore, our current pipeline is ineffective when an apple is concealed by another apple (self-occlusion). To address these challenges, we suggest using Mask-R CNN [27] for instance segmentation, which will enable each apple to be distinctly isolated from the others. Following segmentation, we plan to project the segmented

pixels onto a world frame, eliminate outliers, fit the remaining points to a sphere, and then identify the apple center for tracking. This proposed pipeline has the potential to address issues such as self-occlusion and inaccurate apple localization.

References

[1] Zhou, H., Wang, X., Au, W., Kang, H., and Chen, C., 2022. "Intelligent robots for fruit harvesting: Recent developments and future challenges". *Precision Agriculture,* **23**(5), pp. 1856–1907.

[2] Edan, Y., Han, S., and Kondo, N., 2009. "Automation in agriculture". *Springer handbook of automation*, pp. 1095–1128.

[3] Anderson, N., Underwood, J., Rahman, M., Robson, A., and Walsh, K., 2019. "Estimation of fruit load in mango orchards: tree sampling considerations and use of machine vision and satellite imagery". *Precision Agriculture,* **20**, pp. 823–839.

[4] Zhou, R., Damerow, L., Sun, Y., and Blanke, M. M., 2012. "Using colour features of cv.'gala'apple fruits in an orchard in image processing to predict yield". *Precision Agriculture,* **13**, pp. 568–580.

[5] Zaman, Q., Schumann, A., Percival, D., and Gordon, R., 2008. "Estimation of wild blueberry fruit yield using digital color photography". *Transactions of the ASABE,* **51**(5), pp. 1539–1544.

[6] Dalal, N., and Triggs, B., 2005. "Histograms of oriented gradients for human detection". In 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), Vol. 1, Ieee, pp. 886–893.

[7] Lowe, D. G., 1999. "Object recognition from local scale-invariant features". In Proceedings of the seventh IEEE international conference on computer vision, Vol. 2, Ieee, pp. 1150–1157.

[8] Bay, H., Tuytelaars, T., and van Gool, L., 2006. "Surf: Speeded up robust features.-proceedings of the 9th european conference on computer vision, springer lncs vol 3951, part 1".

[9] Chen, S. W., Shivakumar, S. S., Dcunha, S., Das, J., Okon, E., Qu, C., Taylor, C. J., and Kumar, V., 2017. "Counting apples and oranges with deep learning: A data-driven approach". *IEEE Robotics and Automation Letters,* **2**(2), pp. 781–788.

[10] Long, J., Shelhamer, E., and Darrell, T., 2015. "Fully convolutional networks for semantic segmentation". In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3431–3440.

[11] Bargoti, S., and Underwood, J., 2017. "Deep fruit detection in orchards". In 2017 IEEE international conference on robotics and automation (ICRA), IEEE, pp. 3626–3633.

[12] Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., and McCool, C., 2016. "Deepfruits: A fruit detection system using deep neural networks". *sensors,* **16**(8), p. 1222.

[13] Itakura, K., Narita, Y., Noaki, S., and Hosoi, F., 2021. "Automatic pear and apple detection by videos using deep learning and a kalman filter". *OSA Continuum,* **4**(5), pp. 1688–1695.

[14] , 2020. "Application of consumer rgb-d cameras for fruit detection and localization in field: A critical review". *Computers and Electronics in Agriculture,* **177**, p. 105687.

[15] Wojke, N., Bewley, A., and Paulus, D., 2017. Simple online and realtime tracking with a deep association metric.

[16] Liu, X., Chen, S. W., Aditya, S., Sivakumar, N., Dcunha, S., Qu, C., Taylor, C. J., Das, J., and Kumar, V., 2018. Robust fruit counting: Combining deep learning, tracking, and structure from motion.

[17] Liu, T.-H., Nie, X.-N., Wu, J.-M., Zhang, D., Liu, W., Cheng, Y.-F., Zheng, Y., Qiu, J., and Qi, L., 2023. "Pineapple (ananas comosus) fruit detection and localization in natural environment based on binocular stereo vision and improved yolov3 model". *Precision Agriculture,* **24**(1), pp. 139–160.

[18] Yu, Y., Zhang, K., Yang, L., and Zhang, D., 2019. "Fruit detection for strawberry harvesting robot in non-structural environment based on mask-rcnn". *Computers and Electronics in Agriculture,* **163**, p. 104846.

[19] Gonzalez, S., Arellano, C., and Tapia, J. E., 2019. "Deepblueberry: Quantification of blueberries in the wild using instance segmentation". *Ieee Access,* **7**, pp. 105776–105788.

[20] Mehta, S., and Burks, T., 2014. "Vision-based control of robotic manipulator for citrus harvesting". *Computers and electronics in agriculture,* **102**, pp. 146–158.

[21] Xiong, Y., Ge, Y., Liang, Y., and Blackmore, S., 2017. "Development of a prototype robot and fast path-planning algorithm for static laser weeding". *Computers and Electronics in Agriculture,* **142**, pp. 494–503.

[22] Bewley, A., Ge, Z., Ott, L., Ramos, F., and Upcroft, B., 2016. "Simple online and realtime tracking". In 2016 IEEE International Conference on Image Processing (ICIP), IEEE.

[23] Ronneberger, O., Fischer, P., and Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation.

[24] Silwal, A., Parhar, T., Yandun, F., Baweja, H., and Kantor, G., 2021. "A robust illumination-invariant camera system for agricultural applications". In 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3292–3298.

[25] Terven, J., and Cordova-Esparza, D., 2023. A comprehensive review of yolo: From yolov1 to yolov8 and beyond.

[26] Lipson, L., Teed, Z., and Deng, J., 2021. Raft-stereo: Multilevel recurrent field transforms for stereo matching.

[27] He, K., Gkioxari, G., Dollár, P., and Girshick, R., 2018. Mask r-cnn.