Journal of Big Data and Artificial Intelligence

WWW.JBDAI.ORG

ISSN: 2692-7977

JBDAI Vol. 3, No. 1, 2025 DOI: 10.54116/jbdai.v3i1.44

ACCURATE CROP YIELD ESTIMATION OF BLUEBERRIES USING DEEP LEARNING AND SMART DRONES

Hieu D. Nguyen
Rowan University
nguyen@rowan.edu

Brandon McHenry
Rowan University
mchenr49@students.rowan.edu

Thanh Nguyen
Rowan University
nguyent@rowan.edu

Harper Zappone
Rowan University
zappon37@students.rowan.edu

Anthony Thompson
Rowan University
thomps79@students.rowan.edu

Chau Tran
Rowan University
tranch29@students.rowan.edu

Anthony Segrest
Rowan University
segres62@students.rowan.edu

Luke Tonon
Rowan University
tonon153@students.rowan.edu

ABSTRACT

We present an AI pipeline that involves the use of smart drones equipped with computer vision to obtain a more accurate fruit count and yield estimation of the number of blueberries in a field. The core components are two object-detection models based on the YOLO deep learning architecture: a Bush Model, which is able to detect blueberry bushes from images captured at low altitudes and at different angles, and a Berry Model, which can detect individual berries that are visible on a bush. Together, both models allow for more accurate crop yield estimation by allowing intelligent control of the drone's position and camera to safely capture side-view images of bushes up close. In addition to providing experimental results for our models, which show good accuracy in terms of precision and recall when captured images are cropped around the foreground center bush, we also describe how to deploy our models to map out blueberry fields by using different sampling strategies and discuss the challenges of annotating very small objects (blueberries) and difficulties in evaluating the effectiveness of our models.

Keywords *blueberry crop yield, precision agriculture, smart drones.*

Figure 1: Image of a blueberry bush.

1. Introduction

Precision agriculture by using AI and autonomous drones has been shown to be an effective approach in not only estimating yield for many different crops but also in detecting and managing weeds and diseases. One popular specialty crop for which an accurate estimation of crop yield is important is the blueberry; prediction early in the growing season is important in helping farmers make pricing decisions, hire a sufficient number of pickers, and inform their distributors of the available supply before harvest time.

One AI-based approach to estimating crop yield involves using deep learning models trained on images to detect fruits and vegetables; many such models have been developed for many of these, including blueberries, grapes, apples, and tomatoes. However, early models were trained on either simulated data (Wang et al. 2023) or close-up images of clusters of fruit (from hand-held cameras or mobile devices) and not the entire bush or vine (Hani et al. 2020; Osman et al. 2021; Egi et al. 2022; Hofinger et al. 2023; Melnychenko et al. 2024); thus, these models are inappropriate for industrial use in which crop count and disease detection must be performed over a large field many acres in size. For example, a one-acre blueberry field can contain up to 1,000 bushes, which would make walking through such a field to capture a good sample of close-up images quite time-consuming; thus, a "boots-on-the-ground" approach is impractical.

However, by using unmanned aerial vehicles (UAV), in our case, drones, to capture images of blueberry bushes from a farther distance presents a much more efficient approach to mapping large fields.[1] In particular, we consider smart drones programmed with computer vision, that is, drones that have an onboard mini computer to process images captured by an onboard camera, to identify a blueberry bush and position itself (drone) to capture an optimal view of the bush that maximizes the number of visible berries on the bush. In addition, we make a distinction between a smart drone versus an autonomous drone, in which the former is capable of making its own decisions during flight, for example, by altering its path to avoid collision, as opposed to an autonomous drone in which its actions, including its flight path, are pre-programmed before takeoff. For example, commercial drones that are equipped with collision-avoidance systems (based on technologies such as infrared, stereo vision, and LiDAR) and provide object-tracking capabilities would fall under our classification of a smart drone.

In this work, we present a pipeline of object-detection models based on the YOLO (You Only Look Once [Redmon et al. June 2016]) deep learning architecture to estimate the crop yield of a blueberry field by using *smart* drones programmed with these models to accurately capture images of blueberry bushes and detect the number of harvestable berries on each bush (an example of a blueberry bush is shown in Figure 1). Our work is novel in its approach of detecting not only individual berries that are visible on an entire blueberry bush but also in detecting the bush itself to guarantee an accurate drone position and image capture of the bush. We distinguish such a smart mission from an autonomous mission as follows: in the latter, a drone is programmed before the start of the mission to fly to predetermined points of a blueberry field, say, along a row of bushes but keeping itself some fixed distance away

[1]We note that ground autonomous vehicles (or robots) offer an alternative solution; however, we shall not discuss them and their trade-offs in this paper because this topic has been well addressed in the literature and doing so will set us too far adrift from the focus of our work.

from the row (and high enough to clear neighboring rows); moreover, its onboard camera can be programmed to capture images of these bushes but from a fixed angle of view. However, this does not always guarantee that the bushes will be fully captured by the drone's camera. For example, this situation may occur if the drone's position is blown off course due to wind or temporary loss of GPS. Thus, it is not possible for the drone to adjust its distance from the bushes or its camera's angle of view in real time to compensate for this.

In contrast, drones programmed with our Bush Model can fly a more intelligent mission in which it is able to adjust its position and fly as close as possible alongside a bush (while maintaining a safe distance) to capture an optimal view of the bush so that berries on it appear as large as possible (for more accurate detection). This also allows for the implementation of various sampling strategies, for example, random stratified sampling of bushes, without having to know the exact GPS location of each bush, something that would be required for autonomous missions. However, coupled with real-time kinematic positioning, our pipeline makes precision mapping of blueberry fields possible where the location of each bush can be geotagged by using our Bush Model. The berry count can then be calculated for each bush after the flight by using our Berry Model. Then, given appropriate field data, which we describe in Section 3, crop yield can be estimated more accurately than from current methods.

The rest of the paper is divided as follows. In Section 2, we discuss related works. In Section 3 we present our proposed pipeline for mapping a blueberry field to obtain a more precise estimation of crop yield. In Section 4, we present experimental results for our models, which we hope provides baseline results for future researchers to compare their work against, and discuss challenges we faced in annotating tiny objects (blueberries) and how this impacted the effectiveness of our models.

2. Related Works

Recent advances in computer vision, in particular object-detection models based on deep learning, such as the YOLO architecture, have led to a proliferation of works in precision agriculture. More specifically, many of these works present highly accurate models to perform fruit detection and yield estimation. Because many articles have recently been published in this field (a 2020 survey article [van Klompenburg et al. 2020] reviewed 30 articles that used deep learning models), we limit our discussion to works that involve either deep learning models for detecting blueberries or drone-based methods.

Works similar to ours include Yildirim and Ulu. 2023, in which a pipeline for estimating crop yield of apples by using drones and objection-detection models based on the Faster R-CNN and SSD-Mobilenet architectures is described; Wang et al. 2023 and Melnychenko et al. 2024, in which improved YOLOv5 models are used to detect apples from aerial drone images; Hani et al. 2020, in which a semantic segmentation approach is used to detect and count apples but trained on ground-based images; Egi et al. 2022, in which a YOLOv5 is trained to detect tomatoes from drone-based images; Osman et al. 2021, in which YOLO-based models are used to detect apples, oranges, and pumpkins; and Hofinger et al. 2023, in which a YOLOv5 model is trained to detect black pine tree tops from UAV images.

We note that the aforementioned fruits and trees in the articles cited previously are somewhat large in comparison with blueberries, which, due to their size, are much more difficult to accurately annotate, especially in aerial drone images in which the spatial resolution of the berries is poor due to their small size and distance from the drone (Figure 1). Works related to fruits that are comparable in size include Akiva et al. 2020, in which a deep learning model based on U-Net and trained on aerial images is described to segment and count cranberries for yield estimation and sun exposure; Shen et al. 2023, in which YOLO-based models are used to perform real-time tracking and counting of grape clusters; and Pinheiro et al. 2023 in which YOLO-based models are used to not only detect grape bunches but also assess their quality in terms of damage from lesions.

Works that involve blueberries include MacEachern et al. 2023 and Yang et al. 2023, in which YOLOv3-v4 models are used to detect and estimate different stages of ripeness in blueberries (the latter in wild blueberries) but trained on images captured by hand-held cameras; Ni et al. 2020, in which a Mask R-CNN model is used to segment individual blueberries to estimate fruit maturity; Stefanović et al. 2022, in which a row detection segmentation model based on the U-Net architecture and trained on UAV images is described; and Filipović et al. 2023, in which a bush-detection model is described but trained on annotations of only the trunks of blueberry bushes and not the entire bush; however, our Bush Model is trained to detect the entire bush, which is necessary for accurate crop yield estimation.

There are very few works that are similar to ours in which their computer vision models are validated on images and the results are compared against the actual fruit count per plant or tree. In particular, for our validation dataset, berries on 15 blueberry bushes were all hand picked to obtain actual fruit counts (what we call "picked ground

truth"). These works are few in number because actual or harvested crop yield is typically recorded by weight and not by the number of fruit. Among such works, most involve large fruit such as apples (Bargoti and Underwood, 2017) and mangos (Stein et al. 2016; Payne et al. 2013) but also for almonds (Underwood et al. 2016) and grapes (Palacios et al. 2023). However, none of these works explicitly discuss the ratio of visual fruit count to actual fruit count (per bush) as we do in Section 6.4 to better understand the amount of occlusion.

3. Pipeline

Let F be a two-dimensional rectangular blueberry field of size A (in acres) that contains C bushes. We assume that GPS coordinates of the corners of F and the direction D of the rows of bushes in F are known; moreover, we assume that all rows run along the same direction (Figure 2). Let Y denote the crop yield of F (berries/acre).

Our pipeline for estimating Y can be summarized as follows: we first use stratified sampling to fly a smart drone over random points of F and capture images of bushes at these points by using our Bush Detection model. We then calculate the number of berries on these bushes by using our Berry Detection model to estimate Y by using formula 1. Here are the main steps of our pipeline, depending on the sampling method:

1. Stratified Sampling: We partition F into a grid of $M \times N$ non-overlapping square cells, denoted by $\{C_{mn}\}$. We present two stratified sampling strategies to sample bushes within each cell: point sampling and row sampling. In point sampling, we select a single bush closest to a randomly chosen point inside each C_{mn}. In row sampling, we sample a row of bushes inside C_{mn} (Figure 3).

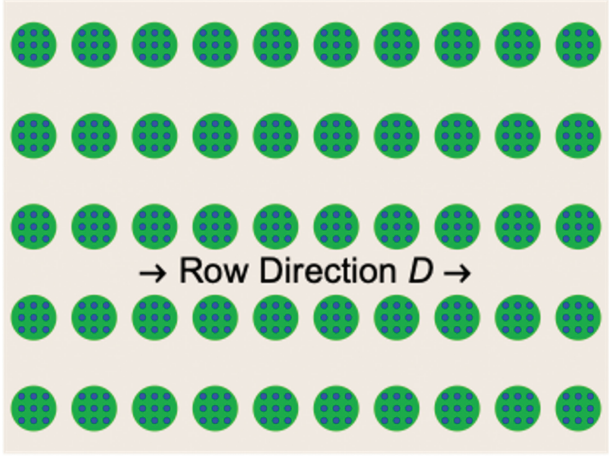

Figure 2: Row direction of a field.

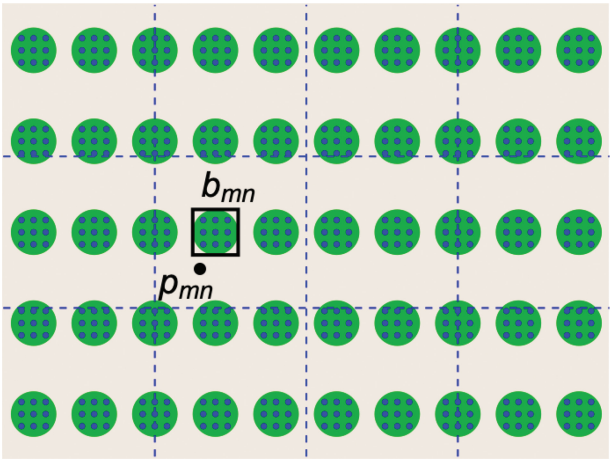

Figure 3: Stratified sampling of a field.

(a) Point Sampling: Select a random point, denoted by p_{mn}, inside each cell C_{mn}.

(b) Row Sampling: Select a random point p_{mn} along an edge of C_{mn} whose direction is perpendicular to D.

(c) Fly the drone to each position p_{mn} (at given altitude h).

2. Single Bush Detection

(a) At position p_{mn}, use a drone camera (pointed down) to capture bushes in its angle of view (see Figure 4 for a birds-eye view) and use our Bush Model to identify a bush closest to p_{mn}. Denote by b_{mn} the position of this closest bush, that is, the center of the bounding box enclosing the bush as illustrated in Figure 3. Then, apply object tracking by using DeepSort to begin tracking the bush.

(b) Program the drone to fly horizontally to position b_{mn} (while maintaining altitude h) so that the drone is directly over the bush.

3. Bush Image Capture (Angled-Side View)

(a) Fly the drone horizontally to one side of the bush (side chosen randomly to account for factors such as the sun's angle) in a direction perpendicular to D (Figure 2) so that it is distance d away from b_{mn} while simultaneously adjusting the camera angle to keep the bush in view by using object tracking. Further adjustments to either the drone position or camera angle (or combination of both) can be made so that the entire bush is in view of the camera (Figure 5).

Figure 4: Blueberry field from a birds-eye view.

Figure 5: Row of blueberry bushes from a angled-side view.

(b) Point Sampling: Use the drone camera to capture the image of the bush (at position b_{mn}) and record the coordinates of the bounding box predicted by the Bush Model, denoted by c_{mn}.

(c) Row Sampling: Fly the drone along D and use the camera to capture photos of bushes that appear in the center of its view and record the bounding box coordinates c_{mn} of each captured bush until the drone reaches the opposite edge of the cell C_{mn}. Apply object tracking to distinguish different bushes and adjust the drone's position to ensure a photo capture of the entire bush.

4. Berry Counting (Post-Mission)

(a) Use bounding box coordinates c_{mn} to determine the number of visible berries on the corresponding bush (from one side) captured by images in the previous step. We describe two approaches (we discuss their trade-offs in Section 4):

i. Image Cropping: Crop each image to contain only the bush (with bounding box coordinates c_{mn}) and apply the Berry Model on cropped image to obtain the number of visible berries on each bush.

ii. Bounding Box Filtering: Apply the Berry Model on the entire image and count only detections of berries that are inside the bush's bounding box to obtain the number of visible berries on each bush.

(b) Calculate the mean number of berries obtained in the previous step (averaged over all bushes) and double this answer to obtain the mean number of berries per bush, denoted by B.

5. Crop Yield Estimation: We calculate crop yield Y as follows:

$$Y = \alpha \cdot \frac{B \cdot C}{A} \tag{1}$$

where B is determined from the previous step, C is the bush count, A is the size of F, and α is a fixed proportionality constant, called the picked-visual ratio (PVR), which describes the ratio of what we refer to as the picked ground truth (GT) to the visual GT:

$$\alpha = \frac{\text{Picked GT}}{\text{Visual GT}} \tag{2}$$

The picked GT is defined to be the number of berries that are actually on a bush (or on a group of bushes) and verified by manually picking and counting all the berries on that bush. However, the visual GT is defined to be the number of berries on the same bush that is visible from a side view and ideally given by B if our Berry Model was perfect. We assume α to be given and that it would be determined from historical data because α would be highly dependent on factors such as climate, soil, variety of blueberry, and height and angle of the drone camera. In Section 6, we provide the first estimates of α that were obtained from two validation datasets in which all the berries on the foreground center bush of each image were picked by hand by our team to obtain the picked GT.

4. Datasets

4.1. Data Collection

Our data consist of images (still photos and video frames) of blueberry bushes (highbush blueberry, *Vaccinium corymbosum*) from two different varieties, Duke and Draper. These images were collected at outdoor blueberry farms in southern counties of New Jersey and captured by using a combination of hand-held (including cellphone) cameras and drone cameras to create a more diverse dataset. Although our total collection consists of more than a thousand such images, only a fraction of them were used to train our models due to limited resources and the time-consuming process of annotating these images, which we further discuss below.

The following datasets were created to train and validate our Berry and Bush Models.

4.2. Berry Datasets

The Berry datasets in total consist of 95 annotated images that were either captured by using drone cameras or hand-held cameras:

1. 35 aerial photos (drone)
2. 60 ground photos (hand-held)

These 95 images represent the total number that we have been able to completely annotate to date. Although relatively few in number, these images contain well over 1,00,000 annotations of berries, which we discuss later, in

Section 4.4. Thus, we believe that the size of our datasets at this point is sufficient to obtain reasonably accurate models, as we demonstrate later.

A train/validation split of 84/16 was used to divide our data into a Training Set (80 images) and three Validation Sets A, B, C, each consisting of 5 images (Table 1). We divided the Training Set into two parts (Table 2): a Drone subset consists of 20 images captured by DJI Phantom 3 and Autel Evo II Pro drones, and a hand-held subset that consists of 60 images captured by various hand-held cameras (Iphone, Android, Canon EOS Rebel). All in the Training Set were taken in the summer of 2021 and 2022 (Figures 6(a), 6(b) and 7(a), 7(b) for sample images). The validation datasets are different as follows: Validation Set A consists of five drone images taken in summer 2022 with a DJI Phantom 3 drone; Validation Set B consists of five drone images of the same foreground bushes as in Set A, but of their opposite side; Validation Set C consists of five drone images taken in Summer 2023 by using a DJI Mini 3 drone.

The reason for merging higher-quality hand-held images with drone images to create our (Merged) Training Set when we began this project 3 years ago is that we anticipated future improvements in drone cameras to match the spatial resolution of current hand-held cameras (different cameras can have the same pixel resolution but different spatial resolution, even after accounting for ground sampling distance). In particular, images captured with the DJI Mini 3 from 2023 (Validation Set C) seems to have higher spatial resolution in comparison with those captured with the DJI Phantom 3 in 2022 (Validation Set A). We present validation metrics in Section 6 that provide evidence to support this, although it should be noted that Validation Sets A and C are images of different blueberry

Table 1: Berry datasets.

Berry Datasets	Images	Green Annotations	Blue Annotations	Total Annotations
Training Set	80	61,680	17,471	79,151
Validation Set A	5	11,328	1,059	12,387
Validation Set B	5	8,810	906	9,716
Validation Set C	5	13,093	7,060	20,153
Total	95	94,911	26,496	1,21,407

Table 2: Berry training set.

Berry Training set	Images	Green Annotations	Blue Annotations	Total Annotations
Drone	20	31,372	4,694	36,066
Hand-held	60	30,308	12,777	43,085
Total (Merged)	80	61,680	17,471	79,151

(a) DJI PHantom 3 (b) Autel Evo II

Figure 6: Sample drone images from the Training Set. (a) DJI PHantom 3; (b) Autel Evo II.

(a) Canon EOS Rebel (b) iPhone

Figure 7: Sample hand-held images from the Training Set. (a) Canon EOS Rebel; (b) iPhone.

varieties, Duke and Draper, respectively, and that the ratio of green-to-blue berries differs significantly for these two datasets, which we address in the next section.

4.3. Bush Datasets

The Bush datasets consist of 256 drone images of blueberry bushes captured at various different altitudes and camera angles (e.g., birds-eye versus side views). We used a train/validation split of 90/10 to define our Training and Validation Sets (Table 3).

4.4. Annotation

Images in our dataset were manually annotated by using computer vision platforms Roboflow and CVAT, and involved a team of more than 10 people (mostly undergraduate research students in our research laboratory).

Berry Model: Images used for training our object detection Berry Model were annotated by drawing a rectangular bounding box tightly around each individual blueberry that is visible in the image and labeling it according to one of two color classes: Green or Blue (Figure 8). We did not have an objective criteria for separating the berries into these two classes, except by providing the annotators with examples of berry images that were labeled by consensus (Figures 9 and 10). We considered creating additional color classes to distinguish the berries but decided on two classes to allow for quicker annotation given the total number of berry annotations in our dataset (>120,000 annotations), which we believe to be the largest of its kind. The number of annotations (Green, Blue, total) are given in Table 1. Observe that the number of Green annotations is significantly higher than the number of Blue annotations, with the ratio of Green to Blue highest for Validation Set A (10.7) and smallest for Validation Set C (1.85). This is because the images in Set C were captured much closer to the first harvest date compared with the images in Set A.

Occluded berries were annotated if the annotator was convinced that the object was a berry and only its visible portion was annotated. Shadowy and/or blurry berries were also annotated according to the same criterion. We acknowledge that this criterion is dependent on the visual acuity of the annotator and exposes the difficulty of annotating tiny objects such as blueberries. One can use majority voting by using a group of annotators; however, given the large number of berries in a single image (>1,000 berries were annotated on average), we considered this approach to not be feasible given our limited resources. In addition, we annotated only those berries that were large enough to be *harvestable*. Berries that fruited late in the season are too small in size to be harvestable by commercial farms. Examples of berries that were not annotated are shown in Figure 11.

Table 3: Bush datasets.

Bush Dataset	Images	Bush Annotations
Training Set	473	2,684
Validation Set	26	314
Total	256	2,998

Figure 8: Example of blueberry annotations (Green and Blue classes).

Figure 9: Examples of berries labeled as Green.

Figure 10: Examples of berries labeled as Blue.

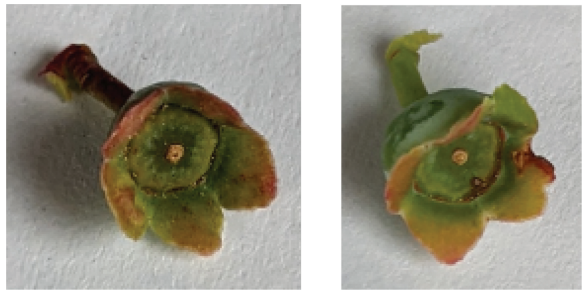

Figure 11: Examples of berries considered too small (not annotated).

Bush Model: For our Bush Detection Model, images were annotated by drawing a rectangular bounding box around each blueberry bush in the foreground of the image, including its trunk (if visible) and all branches (Figures 12 and 13). Neighboring bushes with long branches will unavoidably cause their bounding boxes to overlap, as seen in Figure 13.

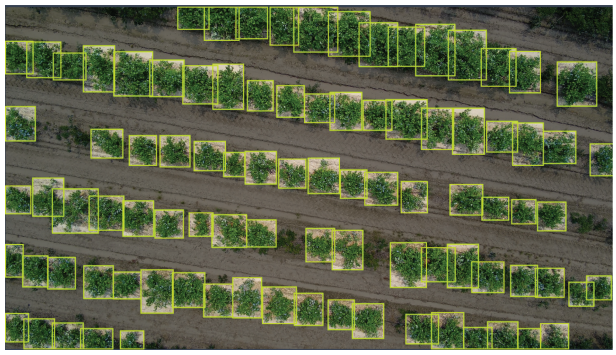

Figure 12: Example of bush annotations (birds-eye view).

Figure 13: Example of foreground bush annotations (side view).

4.5. Data Augmentation

We perform data augmentation as follows to create a diverse training dataset for more effective training. For our Bush Model, we applied variations in Hue (± 0.015), Saturation (± 0.7), Value (brightness) (± 0.4), Translation (± 0.1), Scale (± 0.5), and Flip Left-Right (0.5 probability).

However, for our berry counting model, data augmentation consisted of dividing each original full-size image into tiles. This is because the YOLO model works better with smaller, square images; thus, we divide each image into 640×640 tiles. If the image cannot be perfectly cut into 640×640 tiles (starting at top left), then the leftover pieces are discarded. See Figure 14 for a depiction of the tiling process. Any berry annotation with its bounding box that appears in two neighboring tiles is decided by choosing the tile that contains the center of the box.

Figure 14: Depiction of the tiling process.

5. YOLO Objection Detection

5.1. YOLOv5

YOLOv5 is the fifth version of the YOLO family of compound-scaled object detection models, first introduced by Redmon et al. (2016). We chose YOLOv5 instead of other object-detection models, for example, Faster R-CNN, because of its state-of-the-art performance (back when we began this project more than 3 years ago) in terms of inference speed relative to accuracy. There are many higher versions of YOLO now available; however, experiments show negligible improvement in training metrics for both the Berry and Bush Models when using say version YOLOv8 in comparison with YOLOv5. The focus of this paper is not to present a comparison of all the different versions of YOLO but to establish baseline results for YOLOv5 that others can compare their results against.

Both Berry and Bush Models were trained by using YOLOv5s (small) because there was little improvement in the training metrics when resorting to larger-sized models such as YOLOv5m (medium) and YOLOv5l (large). In addition, especially for the Bush Model, in which inference needs to be performed in real time by the drone on-board computer during flight, the small-size model makes this possible. We were able to achieve 6-9 frames per second performing detection the Bush model on a Jetson Nano with Deepstream running YOLOv5s.

5.2. Training

Both the Bush and Berry Models were trained on Bush and Berry Datasets, respectively, by using YOLOv5s' default hyperparameter settings found in its hyperparameter yaml file "hyp.scratch-low.yaml." We used a batch size of 32 and trained for up to 400 epochs for the Bush Model and 300 epochs for the Berry Model; the default early stopping criterion was used to stop training when cross-validation loss diverged from the training loss. Training metrics were calculated by using Ultralytics YOLOv5 utils Python library through the script "metric.py." We found little difference in accuracy by changing other default parameters and believe that optimizing these parameters would not change the conclusions of our paper; we believe that it is more important to increase the size of our dataset and add more higher-resolution images to significantly improve our model.

5.3. Validation

The best fold from fivefold cross-validation of each model (base on the highest mAP:0.5) was used to perform validation on the datasets in Tables 1 and 3. For the Berry Model, each image was first divided into overlapping 700 $\times$ 700 tiles, or as close as possible to these dimensions, so that they overlap by 60 pixels in each dimension; this avoids double-counting berries that may be split if non-overlapping tiles were used. Each tile was then passed through the Berry Model and post-processing was used to remove duplicate bounding boxes that appear in two overlapping tiles. To compute precision and recall for each class, a confidence threshold of 0.1 was applied to generate detections, and an Intersection over Union (IOU) threshold of 0.3 was applied to count true positives when comparing them against GTs; if more than one detection matched a GT in terms of both IOU threshold and class, then the detection with the highest confidence is selected. A low IOU threshold was used (in comparison with YOLOv5's default threshold of 0.6) to avoid eliminating correct predictions that did not overlap sufficiently with the GT. This is because bounding boxes of blueberries are small in dimension; thus, detection errors in the position of these boxes by just a few pixels can significantly impact their IOU.

5.4. DeepSORT Tracking

To test the accuracy of our Bush Model in tracking bushes as discussed in our pipeline in Section 3, we used DeepSORT to calculate multiple object tracking accuracy (MOTA). DeepSORT is a computer vision tracking algorithm for tracking objects from a video stream by assigning an ID to each detected object (Wojke et al. 2017). It is an extension of the Simple Online and Realtime Tracking (SORT) because it integrates appearance information based on a deep appearance descriptor. We applied DeepSORT to two short video clips: one of a drone performing Point Sampling and the other performing Row Sampling. Results are presented in Section 6.

6. Experimental Results

In this section, we present training and validation results for our Berry and Bush Models, both separately and also when combined to perform bush cropping to obtain a total berry count for only the foreground center bush. We also present tracking results (MOTA) for the Bush Model by using the DeepSORT algorithm.

6.1. Berry Model

Three different Berry Models were trained by using fivefold cross-validation: Drone, Hand-Held, and Merged. The Drone and Hand-Held Berry Models were trained on the 20 drone images and 60 hand-held images, respectively (Table 2). The Merged Berry Model (or simply Berry Model) was trained on the merged dataset of 80 images (drone and hand-held); we refer to this merged dataset as the Berry Training Set.

Training Results: Training metrics for the three Berry Models (Drone, Hand-Held, Merged) are given in Tables 4-6, respectively. The Hand-Held Model performed best across all metrics as expected (precision, recall, mAP, and F1) because the bushes in the hand-held images were shot at a closer distance, with berries appearing larger than those in drone images, thus making detection easier. The Drone Model performed almost as well as the Hand-Held Model in terms of precision, but recall was significantly worse. These two models are only as good as the data that they are trained on, and, so. when we present validation results below, we will see that their performance is reversed, thus supporting the need for a Merged Model that is robust to a variety of different types of images.

Validation Results: Tables 7, 8, and 9 show precision and recall for the three Berry Models (Drone, Hand-Held, Merged) validated on Validation Sets A, B, and C, respectively. For Sets A and B (Tables 7 and 8), the Drone Berry Model had the highest overall precision among all the models; however, the Merged Model had significantly higher overall recall and slightly better Blue precision. However, the Hand-Held Berry Model performed the worst in all categories. In particular, Green recall was extremely low, which indicated that the model failed to detect many

Table 4: Training metrics for the Drone Berry Model.

Training	Precision	Recall	mAP 0.5	mAP 0.5:0.95	F1
Fold 1	0.8271	0.6472	0.7281	0.3773	0.7262
Fold 2	0.8039	0.6378	0.7143	0.3724	0.7113
Fold 3	0.8184	0.6692	0.7350	0.3848	0.7363
Fold 4	0.8347	0.6583	0.7446	0.3945	0.7361
Fold 5	0.8032	0.6707	0.7295	0.3753	0.7310
Mean	0.8175	0.6566	0.7303	0.3809	0.7282
SD	0.0139	0.0142	0.0110	0.0089	0.0103

Table 5: Training metrics for the Hand-Held Berry Model.

Training	Precision	Recall	mAP 0.5	mAP 0.5:0.95	F1
Fold 1	0.8230	0.7518	0.8159	0.5119	0.7858
Fold 2	0.8527	0.7623	0.8339	0.5290	0.8050
Fold 3	0.8553	0.7450	0.8287	0.5176	0.7963
Fold 4	0.8606	0.8009	0.8639	0.5418	0.8297
Fold 5	0.8259	0.7640	0.8269	0.5320	0.7938
Mean	0.8435	0.7648	0.8338	0.5264	0.8021
SD	0.0177	0.0216	0.0180	0.0119	0.0169

Table 6: Training metrics for the Merged Berry Model.

Training	Precision	Recall	mAP 0.5	mAP 0.5:0.95	F1
Fold 1	0.8434	0.7267	0.7965	0.4822	0.7807
Fold 2	0.8303	0.7160	0.7852	0.4643	0.7690
Fold 3	0.8477	0.7253	0.7941	0.4805	0.7817
Fold 4	0.8323	0.7260	0.7962	0.4806	0.7755
Fold 5	0.8319	0.7085	0.7820	0.4636	0.7653
Mean	0.8371	0.7205	0.7908	0.4742	0.7744
SD	0.0079	0.0080	0.0067	0.0094	0.0072

Table 7: Drone, Hand-Held, Merged Berry Models: Validation metrics for Validation Set A.

Berry Model	Precision (Green Class)	Recall (Green Class)	Precision (Blue Class)	Recall (Blue Class)	Precision (Both Classes)	Recall (Both Classes)
Drone	0.775	0.708	0.807	0.245	0.776	0.67
Hand-Held	0.749	0.069	0.092	0.283	0.255	0.087
Merged	0.755	0.745	0.804	0.361	0.757	0.713

Table 8: Drone, Hand-Held, Merged Berry Models: Validation metrics for Validation Set B.

Berry Model	Precision (Green Class)	Recall (Green Class)	Precision (Blue Class)	Recall (Blue Class)	Precision (Both Classes)	Recall (Both Classes)
Drone	0.657	0.675	0.621	0.245	0.655	0.637
Hand-Held	0.608	0.205	0.111	0.461	0.337	0.228
Merged	0.601	0.734	0.69	0.317	0.605	0.697

Table 9: Drone, Hand-Held, Merged Berry Models: Validation metrics for Validation Set C.

Berry Model	Precision (Green Class)	Recall (Green Class)	Precision (Blue Class)	Recall (Blue Class)	Precision (Both Classes)	Recall (Both Classes)
Drone	0.609	0.478	0.866	0.433	0.68	0.461
Hand-Held	0.793	0.459	0.818	0.476	0.802	0.465
Merged	0.815	0.462	0.872	0.434	0.835	0.451

Table 10: Merged Berry Model validation metrics for Validation Sets A, B, C.

Validation Dataset	Precision (Green Class)	Recall (Green Class)	Precision (Blue Class)	Recall (Blue Class)	Precision (Both Classes)	Recall (Both Classes)
Set A	0.755	0.745	0.804	0.361	0.757	0.713
Set B	0.601	0.734	0.69	0.317	0.605	0.697
Set C	0.815	0.462	0.872	0.434	0.835	0.451

green berries, especially those in background bushes where they appear much smaller, which makes it more difficult for the model to detect them. Also, precision for class Blue was also quite low, which shows that the Hand-Held Berry Model did a poor job of correctly detecting blue berries.

As for Validation Set C (Table 9), performance reversed with the Merged Model now having the highest overall precision and only slightly worse overall recall compared with the Drone and Hand-Held Models. Observe that overall recall for the Hand-Held Model significantly improved compared with results in Tables 7 and 8. However, all three models performed poorly on overall recall. An inspection of the false negatives were of berries on background bushes that were annotated but either too small for any of the models to detect or too shaded for the model to distinguish as a berry. The results on overall precision provide evidence that training on combined drone and hand-held images helped to improve the (Merged) Berry Model, with only a slight decrease in overall recall (but best Green recall), in comparison with training on drone and hand-held images separately.

Precision and recall for only the Merged Berry Model to help better compare results for the three validation sets (A, B, and C) are isolated in Table 10. Results for Set C yielded the highest overall precision, but, unfortunately, also yielded the lowest overall recall, which we previously described as due to berries on background bushes that are too small in terms of pixel resolution for the model to detect. This is supported by results that we present later on when we consider detecting berries only on the foreground center bush.

Conversely, results for Set B had the lowest overall precision but highest overall recall. Figures 15, 16, and 17 show a sample image (B2) from Set B, but with different types of bounding boxes drawn: GTs (Figure 15),

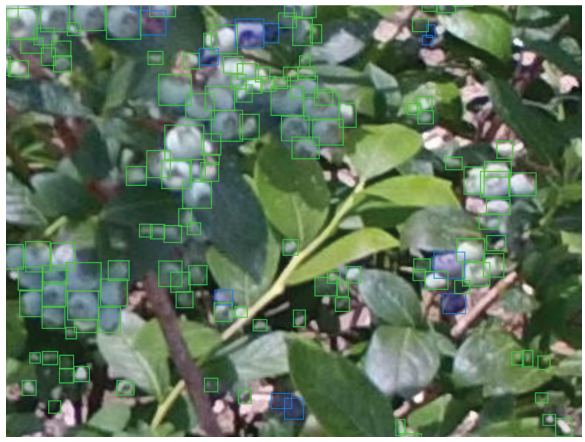

Figure 15: Close-up view of ground truth annotations in sample image from Validation Set B (green and blue boxes denote Green and Blue classes, respectively).

Figure 16: Close-up view of Berry Model predictions in sample image from Validation Set B (blue and red boxes denote Green and Blue classes, respectively).

Figure 17: Close-up view of Berry Model false positives (green) and false negatives (red) for sample image in Validation Set B.

predictions (Figure 16), and false positives and false negatives (Figure 17). A close inspection of the false positives shows that some of them could possibly be berries, but were difficult to discern clearly, which explains why they were not annotated.

6.2. Bush Model

Training/Validation Results: Training metrics for the Bush Model trained on the Bush Training Set are given in Table 11. Validation metrics for the Bush Model validated on the Bush Validation Set are given in Table 12. Both tables show high precision, at approximately 90%, and good recall, ranging from high 70% for training to low 80% for validation. A review of the true negatives (bushes that were not detected) indicates that the Bush Model struggled to detect those bushes at the edge of the image. Fortunately, this issue is not a concern because the goal of the Bush Model is to detect and track foreground bushes.

Sample predictions of the Bush Model from angled-side, birds-eye, and slanted views are shown in Figures 18, 19, and 20, respectively, including a false positive and false negative in the latter figure.

Bush Tracking: We calculated MOTA for two video clips: Bush Video 1 and Bush Video 2. Bush Video 1 (24 seconds) was captured by a DJI drone flying overhead to a bush and simultaneously adjusting its position and camera angle from birds-eye to angled-side view. Bush Video 2 (5 seconds) was captured by a DJI drone flying sideways along a row of blueberry bushes. MOTA results for both video clips are given in Table 13. Although MOTA is lower than what we hoped for, a review of the detections shows that the Bush Model does a very good job of tracking the foreground center bush, which is the primary goal of the model and performs worse for bushes at the edges of the image, something that we previously mentioned.

Table 11: Training metrics for Bush Model trained on Bush Validation Set.

Training Fold	Precision	Recall	mAP 0.5	mAP 0.5:0.95
Fold 1	0.899	0.76	0.867	0.508
Fold 2	0.842	0.723	0.807	0.433
Fold 3	0.881	0.774	0.869	0.493
Fold 4	0.939	0.893	0.946	0.592
Fold 5	0.877	0.733	0.829	0.492
Mean	0.888	0.777	0.864	0.504
SD	0.035	0.068	0.053	0.057

Table 12: Validation metrics for Bush Model validated on Bush Validation Set (using best fold).

Dataset	Precision	Recall	mAP 0.5	mAP 0.5:0.95
Bush Validation Set	0.916	0.834	0.916	0.538

Figure 18: Bush Model prediction on a sample validation image (angled-side view of bush).

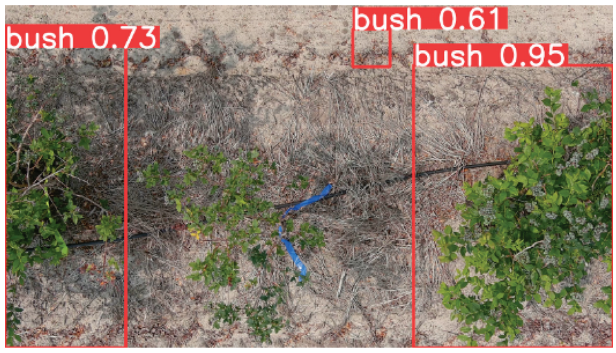

Figure 19: Bush Model prediction on a sample validation image (birds-eye view), showing a false positive (bush marked with confident 0.61) and false negative (bush tagged with blue ribbon).

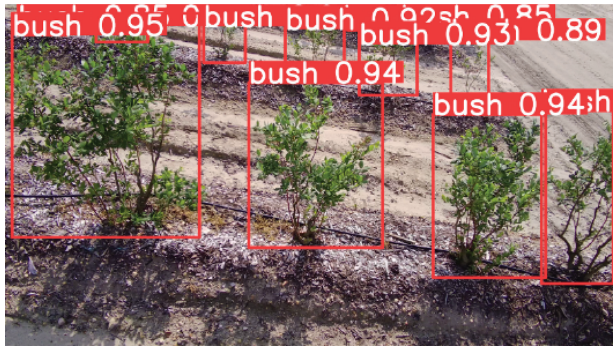

Figure 20: Bush Model prediction on an example validation image (slanted view).

Table 13: MOTA for Bush Model of Videos 1 and 2.

Parameter	Bush Video 1	Bush Video 2
Number of frames	365	75
Bush annotations	6464	324
Predictions	7104	352
Mismatch errors	149	7
False positives	2315	96
False negatives	1745	66
IOU threshold	0.5	0.5
MOTA	**0.3489**	**0.4786**

6.3. Bush-Cropped Berry Model

Together, the Berry and Bush Models can be combined into a pipeline to detect only those berries that appear in a single bush. We call this pipeline the Bush-Cropped Berry Model. In particular, we first pass a full-size image through the Bush Model to obtain an array of detected bushes and their corresponding bounding boxes. From these bounding boxes, we select one called the *central bounding box* (corresponding to the foreground center bush) (Figures 21(a) and 21(b)), whose center is closest (in terms of radial distance) to the center of the image and then crop the image (using OpenCV2) around the central bounding box. The cropped image is then passed through the Berry Model to detect berries. Detections are compared against those GTs contained within the central bounding box.

Shown in Tables 14, 15, and 16 are validation results by using the Bush-Cropped Berry Model (Drone, Hand-Held, and Merged) on Validation Sets A, B, and C, respectively. This time we see the Merged Model outperforming the Drone and Hand-Held Models in overall recall for all validation sets. The Hand-Held Berry Model performed the

(a) Full image (b) Cropped image

Figure 21: Example of image cropped around foreground center bush by Bush Model and then fed into Berry Model. (a) Full image; (b) Cropped image.

Table 14: Bush-Cropped Berry Model (Drone, Hand-Held, Merged): Validation metrics for Validation Set A.

Bush-Cropped Berry Model	Precision (Green Class)	Recall (Green Class)	Precision (Blue Class)	Recall (Blue Class)	Precision (Both Classes)	Recall (Both Classes)
Drone	0.778	0.675	0.733	0.168	0.776	0.631
Hand-Held	0.483	0.007	0.069	0.019	0.222	0.008
Merged	0.758	0.712	0.811	0.265	0.76	0.673

Table 15: Bush-Cropped Berry Model (Drone, Hand-Held, Merged): Validation metrics for Validation Set B.

Bush-Cropped Berry Model	Precision (Green Class)	Recall (Green Class)	Precision (Blue Class)	Recall (Blue Class)	Precision (Both Classes)	Recall (Both Classes)
Drone	0.683	0.688	0.727	0.251	0.684	0.655
Hand-Held	0.753	0.154	0.171	0.328	0.504	0.167
Merged	0.601	0.747	0.746	0.35	0.605	0.718

Table 16: Bush-Cropped Berry Model (Drone, Hand-Held, Merged): Validation metrics for Validation Set C.

Bush-Cropped Berry Model	Precision (Green Class)	Recall (Green Class)	Precision (Blue Class)	Recall (Blue Class)	Precision (Both Classes)	Recall (Both Classes)
Drone	0.708	0.583	0.848	0.618	0.752	0.595
Hand-Held	0.799	0.589	0.837	0.615	0.812	0.598
Merged	0.809	0.617	0.847	0.605	0.821	0.613

worst in both overall precision and recall for Sets A and B (Tables 14 and 15), as already observed when validated on entire images (Table 10) but performed surprising well on Green precision for Set B (Table 15).

For Set C, the Merged Model achieved the best overall precision and recall, with the Hand-Held Model having only slightly worse results. This demonstrates that the drone images in Set C have almost the same spatial resolution as the hand-held images. We believe going forward that the Merged Model will performed best (in terms of precision and recall) on images captured by future drones, whose camera resolution will only continue to improve.

For comparison, for Validation Sets A, B, and C, all validated by using the same Bush-Cropped Merged Berry Model, are isolated in Table 17. Here, the results are similar to those when validated on entire images (Table 10)

Table 17: Bush-Cropped Merged Berry Model: Validation metrics for Validation Sets A, B, and C.

Validation Dataset	Precision (Green Class)	Recall (Green Class)	Precision (Blue Class)	Recall (Blue Class)	Precision (Both Classes)	Recall (Both Classes)
Set A	0.758	0.712	0.811	0.265	0.76	0.673
Set B	0.601	0.747	0.746	0.35	0.605	0.718
Set C	0.809	0.617	0.847	0.605	0.821	0.613

Table 18: Calculation of α (predicted vs experimental) for the Berry Validation Set A by using the Bush-Cropped (Merged) Berry Model.

Image	Detections	Visual GT	Picked GT	α_p (Predicted)	α (Experimental)
A1	882	1,010	3,312	3.755	3.279
A2	1,451	1,230	3,996	2.754	3.249
A3	511	493	2,888	5.652	5.858
A4	711	847	2,920	4.107	3.447
A5	420	708	1,404	3.343	1.983
Mean	795	858	2,904	3.92	3.56
SD	408	282	950	1.09	1.41
Total	3,975	4,288	14,520	3.65	3.39

Table 19: Calculation of α (predicted vs experimental) for Berry Validation Set B by using the Bush-Cropped (Merged) Berry Model.

Image	Detections	Visual GT	Picked GT	α (Predicted)	α (Experimental)
B1	891	785	1,404	1.576	1.789
B2	885	806	2,920	3.299	3.623
B3	1,012	972	2,888	2.854	2.971
B4	1,856	1,842	3,996	2.153	2.169
B5	1,071	1,043	3,312	3.092	3.175
Mean	1,143	1,090	2,904	2.59	2.75
SD	406	435	950	0.71	0.75
Total	5,715	5,448	14,520	2.54	2.67

but observe that recall significantly improved for Set C because the model no longer needs to detect tiny berries on background bushes, which it had difficulty with when validating on entire images.

6.4. Estimation of PVR

The Bush-Cropped (Merged) Berry Model allows us to estimate the PVR α (see Part 5 of Section 3) by using its detections as an estimate for the picked GT. We denote by α_p (predicted α) to be any approximation of α calculated based on this estimate:

$$\alpha_p = \frac{\text{Picked GT}}{\text{Detections}} \approx \frac{\text{Picked GT}}{\text{Visual GT}} = \alpha \tag{3}$$

However, we distinguish α_p from experimental values of α calculated by using the annotated visual GT of the cropped image, that is, the number of berry annotations within the central bounding box), which we assume to be a very accurate estimate of the true value of α.

Tables 18, 19, and 20 give both predicted and experimental values for α for Validation Sets A, B, and C, respectively, including the total number of detections, visual GT, and picked GT are given for each image (cropped around the foreground center bush). Although the total number of detections seem to be good approximations of the visual GT for Validation Sets A and B (Tables 18 and 19, respectively), this is misleading because these detections

Table 20: Calculation of α (predicted vs experimental) for Berry Validation Set C by using the Bush-Cropped (Merged) Berry Model.

Image	Detections	Visual GT	Picked GT	α (Predicted)	α (Experimental)
C1	1,109	1,507	2,407	2.170	1.597
C2	831	924	3,215	3.869	3.479
C3	1,261	1,491	1,963	1.557	1.316
C4	713	618	2,307	3.236	3.733
C5	1,210	1,457	1,963	1.622	1.347
Mean	955	1,199	2,371	2.67	2.29
SD	225	406	513	1.10	1.21
Total	4,774	5,997	11,855	2.48	1.98

contain many false positives of berries (see overall precision and recall for the Bush-Cropped (Merge) Berry Model in Table 17), which cancel out the many false negatives of berries that were not detected. Thus, an accurate estimation of α_p will depend on an accurate Berry Model.

Experimental values of α differ widely for all three sets (Tables 18, 19, and 20), with mean experimental values highest for Set *A* and lowest for Set C. This shows that estimating α will be challenging because it seems to depend not only on the blueberry variety (recall that Sets A and C correspond to Duke and Draper varieties, respectively) but also which side of the bush is captured (recall that Sets A and B correspond to two sides of the same five bushes).

7. Discussion

Results of our Berry Model highlight challenges with annotating berries and training our models. Berries that are difficult to discern due to their small size, especially those on background bushes, can lead to subjective annotations and thus an ambiguous GT. Many false-positive detections could be argued as true detections of berries, depending on one's visual acuity, but difficult to confirm with certainty because of their low resolution. Moreover, occlusion of partially hidden berries, camouflage of green berries by leaves, and shaded berries make for training an accurate Berry Model quite challenging.

Results of the Bush Model clearly show that detecting bushes is not necessarily an easier task than detecting berries. Obviously, a bush is considerably larger than a berry; however, the complicated branch structure of a bush, in addition its branches possibly overlapping with a neighboring bush, creates challenges in training an accurate bush model.

Results of the Bush-Cropped Berry Model show the effectiveness of cropping around the foreground center bush to eliminate background berries and thus improved the model's precision and recall, which, in turn, provided a more accurate estimation of crop yield. Estimates of the PVR α based on the Bush-Cropped Berry Model show that it can vary significantly and depends on many factors such as the particular side of the bush that is captured and the blueberry variety, and other factors that we did not take into account: bush size, bush foliage density, environmental and soil conditions.

8. Conclusion

In this paper, we presented a pipeline of object detection models based on deep learning for detecting blueberry bushes and individual berries on them. These models allow a smart drone programmed with them to fly intelligent missions, namely to precisely locate bushes and capture their side views, thus obtaining a more accurate estimate of crop yield. We have already begun to test our pipeline by using a custom-build programmable drone to capture data and hope to report on our experimental results in the near future. We hope our work will spur interest in others to address the challenges raised in this paper and improve on our baseline results. All datasets, models, and source code will be made available on Github.

Acknowledgments: *The authors would like to acknowledge partial financial support from the New Jersey Council of County Colleges through their NJ Pathways to Career Opportunities Program, Department of Mathematics and College of Science and Mathematics at Rowan University. The following blueberry farms in South Jersey kindly provided us access to their fields to collect data: Macrie Brothers Farm, Moore's Meadow Farm, and Vacarrella Farm. We also thank other former and current team members who contributed to annotating our datasets: Robert Czarnota, Jacob Green, Lori Green, Felix Hakimi, Lance Ilagan, Nicholas Kaegi, Jamie Kahle, Brian Kim, Tuan Le, Ik Jae Lee, Duy Nguyen, Jonah Rodriguez, and Iosefa Sunia.*

References

Akiva, P., K. Dana, P. Oudemans, and M. Mars. 2020. "Finding Berries: Segmentation and Counting of Cranberries Using Point Supervision and Shape Priors." in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 219–228.

Bargoti, S., and J. Underwood. 2017. "Image Segmentation for Fruit Detection and Yield Estimation in Apple Orchards." *Journal of Field Robotics* **34**, no. 6: 1039–1060.

Egi, Y., M. Hajyzadeh, and E. Eyceyurt. 2022. "Drone-Computer Communication Based Tomato Generative Organ Counting Model Using Yolo v5 and Deep-Sort." *Agriculture* **12**, no. 9: 1290–17.

Filipović, V., D. Stefanović, N. Pajević, Z. Grbović, N. Djuric, and M. Panić. 2023. "Bush Detection for Vision-Based Ugv Guidance in Blueberry Orchards: Data Set and Methods." in 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 3646–3655.

Hani, N., P. Roy, and V. Isler. 2020. "A Comparative Study of Fruit Detection and Counting Methods for Yield Mapping in Apple Orchards." *Journal of Field Robotics* **37**, no. 2: 263–282.

Hofinger, P., H.-J. Klemmt, S. Ecke, S. Rogg, and J. Dempewolf. 2023. "Application of yolov5 for Point Label Based Object Detection of Black Pine Trees with Vitality Losses in Uav Data." *Remote Sens* **15**: 1–13.

MacEachern, C. B., T. J. Esau, A. W. Schumann, P. J. Hennessy, and Q. U. Zaman. 2023. "Detection of Fruit Maturity Stage and Yield Estimation in Wild Blueberry Using Deep Learning Convolutional Neural Networks." *Smart Agricultural Technology* **3**: 100099–11.

Melnychenko, O., L. Scislo, O. Savenko, A. Sachenko, and P. Radiuk. 2024. "Intelligent Integrated System for Fruit Detection Using Multi-Uav Imaging and Deep Learning." *Sensors* **24**, no. 6: 1913–1936.

Ni, X., C. Li, H. Jiang, and F. Takeda. 2020. "Deep Learning Image Segmentation and Extraction of Blueberry Fruit Traits Associated with Harvestability and Yield." *Horticulture Research* **7**, no. 1: 1–14.

Osman, Y., R. Dennis, and K. Elgazzar. 2021. "Yield Estimation and Visualization Solution for Precision Agriculture." *Sensors* **21**, no. 19: 6657.

Palacios, F., M. P. Diago, P. Melo-Pinto, and J. Tardaguila. 2023. "Early Yield Prediction in Different Grapevine Varieties Using Computer Vision and Machine Learning." *Precision Agriculture* **24**, no. 2: 407–435.

Payne, A. B., K. B. Walsh, P. Subedi, and D. Jarvis. 2013. "Estimation of Mango Crop Yield Using Image Analysis – Segmentation Methodn, Localisation and Yield Estimation Using Multiple View Geometry." *Computers and Electronics in Agriculture* **91**: 57–64.

Pinheiro, I., G. Moreira, D. Queirós da Silva, S. Magalhães, A. Valente, P. M. Oliveira, et al. 2023. "Deep Learning Yolo-Based Solution for Grape Bunch Detection and Assessment of Biophysical Lesions." *Computers and Electronics in Agriculture* **206**: 1–14.

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

Shen, L., J. Su, R. He, L. Song, R. Huang, Y. Fang, et al. 2023. "Real-Time Tracking and Counting of Grape Clusters in the Field Based on Channel Pruning with yolov5s." *Computers and Electronics in Agriculture* **206**: 107662.

Stefanović, D., A. Antić, M. Otlokan, B. Ivošević, O. Marko, V. Crnojević, et al. 2022. "Blueberry Row Detection Based on Uav Images for Inferring the Allowed Ugv Path in the Field." in ROBOT2022: Fifth Iberian Robotics Conference.

Stein, M., S. Bargoti, and J. Underwood. 2016. "Image Based Mango Fruit Detection, Localisation and Yield Estimation Using Multiple View Geometry." *Sensors* **16**, no. 11: 1915.

Underwood, J., C. Hung, B. Whelan, and S. Sukkarieh. 2016. "Mapping Almond Orchard Canopy Volume, Flowers, Fruit and Yield Using Lidar and Vision Sensors." *Computers and Electronics in Agriculture* **130**: 83–96.

van Klompenburg, T., A. Kassahun, and C. Catal. 2020. "Crop Yield Prediction Using Machine Learning: A Systematic Literature Review." *Computers and Electronics in Agriculture* **177**: 105709–105718.

Wang, H., J. Feng, and H. Yin. 2023. "Improved Method for Apple Fruit Target Detection Based on yolov5s." *Agriculture* **13**, no. 11: 2167.

Wojke, N., A. Bewley, and D. Paulus. 2017. "Simple Online and Realtime Tracking with a Deep Association Metric." in 2017 IEEE International Conference on Image Processing (ICIP), pp. 3645–3649.

Yang, W., X. Ma, and H. An. 2023. "Blueberry Ripeness Detection Model Based on Enhanced Detail Feature and Content-Aware Reassembly." *Agronomy* **13**, no. 6: 1613–1619.

Yildirim, S., and B. Ulu. 2023. "Deep Learning Based Apples Counting for Yield Forecast Using Proposed Flying Robotic System." *Sensors* **23**, no. 13: 6171.