Finding Berries: Segmentation and Counting of Cranberries using Point Supervision and Shape Priors

Peri Akiva[1] Kristin Dana[1] Peter Oudemans[2] Michael Mars[2]
[1]Department of Computer and Electrical Engineering, Rutgers University
[2]Department of Plant Biology, Rutgers University
{peri.akiva, kristin.dana}@rutgers.edu {oudemans, mm2784}@njaes.rutgers.edu

Abstract

Precision agriculture has become a key factor for increasing crop yields by providing essential information to decision makers. In this work, we present a deep learning method for simultaneous segmentation and counting of cranberries to aid in yield estimation and sun exposure predictions. Notably, supervision is done using low cost center point annotations. The approach, named Triple-S Network, incorporates a three-part loss with shape priors to promote better fitting to objects of known shape typical in agricultural scenes. Our results improve overall segmentation performance by more than 6.74% and counting results by 22.91% when compared to state-of-the-art. To train and evaluate the network, we have collected the CRanberry Aerial Imagery Dataset (CRAID), the largest dataset of aerial drone imagery from cranberry fields. This dataset will be made publicly available.

1. Introduction

The challenges of agriculture presents new opportunities for computer vision methods. Evaluation of crop health, sun exposure and anticipated yields using computational algorithms leads to new methods of farming and resource management. Automated segmentation and counting provides a method of determining value of produce and anticipated profits, as well as optimizing irrigation and water management. Current yield estimation methods rely on data from previous years or manual measurements of small regions. These processes limit the accuracy of predicted yield since weather can be vastly different in consequent years, and randomly sampled measurements may be skewed and costly. Recent studies [42, 23, 37] show that lack of informed decision-making is a significant cause of lost produce. For example, [37] investigates the effect of sudden changes in air temperature from heat waves causing regions with up to 100% yield loss due to a combination of heat stress and water stress. Managing water resources requires balancing the tradeoff of irrigation costs and yield risk. Our goal is to build a non-invasive vision-based crop analysis platform that segments and counts exposed berries, and can serve as a low-cost automated tool for estimating yield and sun exposure. Current precision agriculture state-of-the-art (SOTA) methods utilize ground vehicles, high resolution cameras, lidars, multispectral sensors, and thermal sensors to automate this process, creating more accurate and cost effective solutions to yield and sun exposure estimations. These instruments are utilized in recent precision agriculture work seeking to detect and segment fruits and weeds [44, 3, 2, 24, 33]. However, these sensors are expensive, require specialized knowledge to operate, and often need close proximity to the targeted objects. Systems such as in [3, 2] also require the use of invasive ground vehicle and trained drivers which further increase the cost of the system. Surveys of remote sensing with unmanned aerial vehicles (UAVs) [34] highlight the importance of non-invasive systems in precision agriculture. Detection of fruit stress

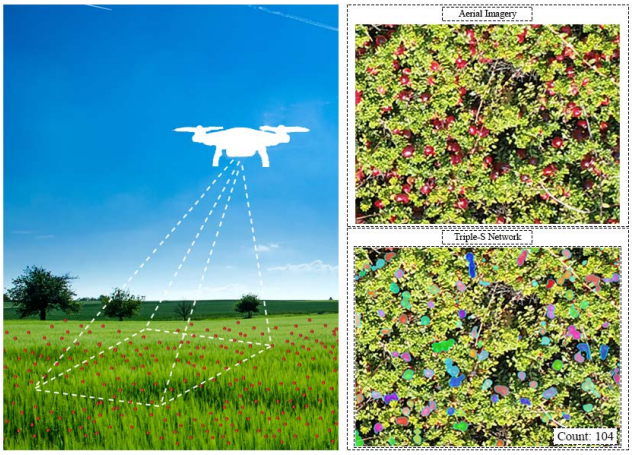


Figure 1: Overview of pipeline. Left: scene illustration of data collection stage. Top right: image captured by the drone. Bottom right: segmentation and count outputs of our Triple-S network. Colors in prediction mask are random and are used to represent instances (colors may repeat). Best viewed in color and zoomed.

and pathogens with UAVs [16, 1, 5, 31, 46, 22, 8] typically require expensive hyperspectral and thermal sensors. Additionally, most UAV methods focus on fruit crop images which are simpler compared to cranberry crops that have numerous occluding leaves in the canopy imagery (see Figure 2). Our approach seeks to count and segment cranberries in RGB images collected by non-invasive equipment. Recent segmentation methods require training algorithms using pixel-wise ground truth obtained manually [50, 29, 13], but such ground truth is expensive to obtain. We develop a novel method using only *point-wise annotations* which are an order magnitude cheaper than full pixel-wise supervision [4]. Our approach pairs the point-click annotations with additional shape and convexity cues to produce instance segmentation results.

The primary contributions of this work are as follows:

- We propose a method named Triple-S Network that encourages shape-specific instance segmentation predictions for small, many-object scenes driven by known shape priors and point supervision.
- We present a a selective watershed algorithm that uses both negative and positive seeds for selective segmentation masks generation.
- We outperform SOTA point supervision semantic segmentation and counting methods on our dataset.
- We provide the largest publicly available dataset of aerial images of cranberry crops with pixel-wise and center point annotations named Cranberry Aerial Imagery Dataset (CRAID).

2. Related Work

Computer Vision in Agriculture. Early precision agriculture using aerial imagery began in the 1980's and includes Soil Teq's field soil fertility mapping system for crops [49] using spectral features. Studies that correlate precision agriculture to higher yields of crops [6, 45] motivated researchers to use computer vision for new ways to measure, survey, and estimate yield of crops. Early work in this domain utilizes colors, shapes [7, 38, 12, 15], reflection levels [47, 22], and multi-spectral features [14, 10, 16] to detect and evaluate fruits, wheat, and weeds. Those methods apply image pre-processing techniques such as contrast and thresholding with machine learning algorithms such as k-nearest neighbors, decision trees, and support vector machines. While early models may perform well under controlled conditions and small datasets, they fail to generalize over diverse and noisy inputs common in real world applications. More recent models incorporate deep learning algorithms to generalize over different environments. Song et al. [44] propose patch-wise fruit classification, using a combination of color classifiers for key-point extraction and fixed patches around each key-point. Those patches are then classified as either fruit or non-fruit images. Bargoti and Underwood [3] propose a segmentation model using fully convolutional networks (FCN) [30] using fully supervised RGB images and meta-data pertaining to camera angle, camera location, type of tree captured, and weather conditions. The model's output is then processed by the watershed algorithm [35] to produce separable regions used for fruit counting. Combining elements from [3, 44], Kestur et al. [24] use 200×200 input patches to a modified fully convolutional network to generate masks patch-wise and stitch them together. A similar FCN model [33] is trained on near infra red (NIR) and RGB sequences. These models, however, offer limited performance with high operation costs, requiring ground vehicles and fully supervised ground truth data. In addition to significant prior work using convolutional networks in fruit imaging, automated weed detection methods use similar deep learning models [48] to distinguish between weed types in RGB images taken from a fixed altitude. The model is trained with image level labels that indicate weed type and is tested on input's 9 sub-images, providing patch-wise weed predictions.

Weakly Supervised Segmentation and Counting. Instance segmentation seeks to not only find the class of each pixel, but also the object instance it belongs to, which indirectly provides object counts in a given scene. Initial development in this task domain is derived from R-CNN [18, 17], utilizing proposal based segmentation. More recent work attempts to minimize the amount of supervision while producing similar performance. The work of [9, 25] first suggested semantic segmentation methods using bounding boxes, followed by [43] to showcase SOTA segmentation using predefined class-wise filling rates. While these methods perform well on everyday scenes, they require bounding box annotations which are more expensive than point annotations, and are computationally demanding, utilizing region proposal networks [18, 39] to generate proposal masks for sets of anchors originating at each pixel. Less supervised methods [51, 28] aim to segment scenes based on image level labels. PRM (Peak Response Map) [51] makes use of class peak response to obtain instance aware visual cues from given inputs. The network generates peak response maps by backpropagating local peaks found in intermediate attention maps. While [51] reports SOTA performance on common datasets, the method exhibits increasing errors when the size and number of objects in the scene increase, which is confirmed by our experiments. Additionally, the network requires pre-processed segment proposals generated by a separate region proposal network. Expanding on PRM [51], [28] refines its output to create pseudo masks used as ground truth to a fully supervised Mask R-CNN [20] model that is robust to noisy masks. Similar to PRM, [28] also requires a separate object proposal network to gener-

Figure 2: Examples of CRAID images with overlaid berry-wise and point-wise ground truth masks. Red and blue dots represent cranberry and background examples. Colors in ground truth masks are random and are used to represent instances. Colors may repeat. Best viewed in color and zoomed.

ate its pseudo masks while facing similar difficulties with small, many object scenes. While instance segmentation finds count indirectly, [40] chooses to directly find counts and locations using center point annotations. This method introduces the Weighted Hausdorff Distance loss to encourage better localization, while regressing over the joint latent features and network output to directly estimate counts. The work closest to our approach is LC-FCN (Localization-based Counting FCN) [27], in which the model aims to detect regions in objects using center point annotations driven by a loss function that encourages object boundaries, point localization, and overall image loss. The split loss used in LC-FCN utilizes the set of pixels representing the boundaries of objects obtained by the watershed algorithm and is calculated for individual blobs and overall image. In contrast, our split loss considers the set of pixels representing the possible regions objects can expand to without crossing to neighboring objects and is calculated against the prediction mask (see Figure 4 for visual comparison). This approach aims to penalize the model if the predicted area is too small, while LC-FCN only penalizes the model if an object crosses a boundary. Additionally, we better constitute object borders using our selective watershed algorithm which uses negative and positive ground truth annotations to define positive and negative regions, unlike LC-FCN which only uses positive ground truth annotations in it watershed split loss.

3. CRAID: CRanberry Aerial Imagery Dataset

3.1. Data Collection

We collect 21,436 cranberry images of size 456×608 to create the largest repository of aerial RGB imagery of cranberry fields which we name CRAID. Images were collected using a Phantom 4 drone from a small range of altitudes with manually fixed camera settings: 100 ISO, 1/240 shutter speed, and 5.0 F-Stop. Data was acquired at weekly intervals to capture albedo variations in cranberries,

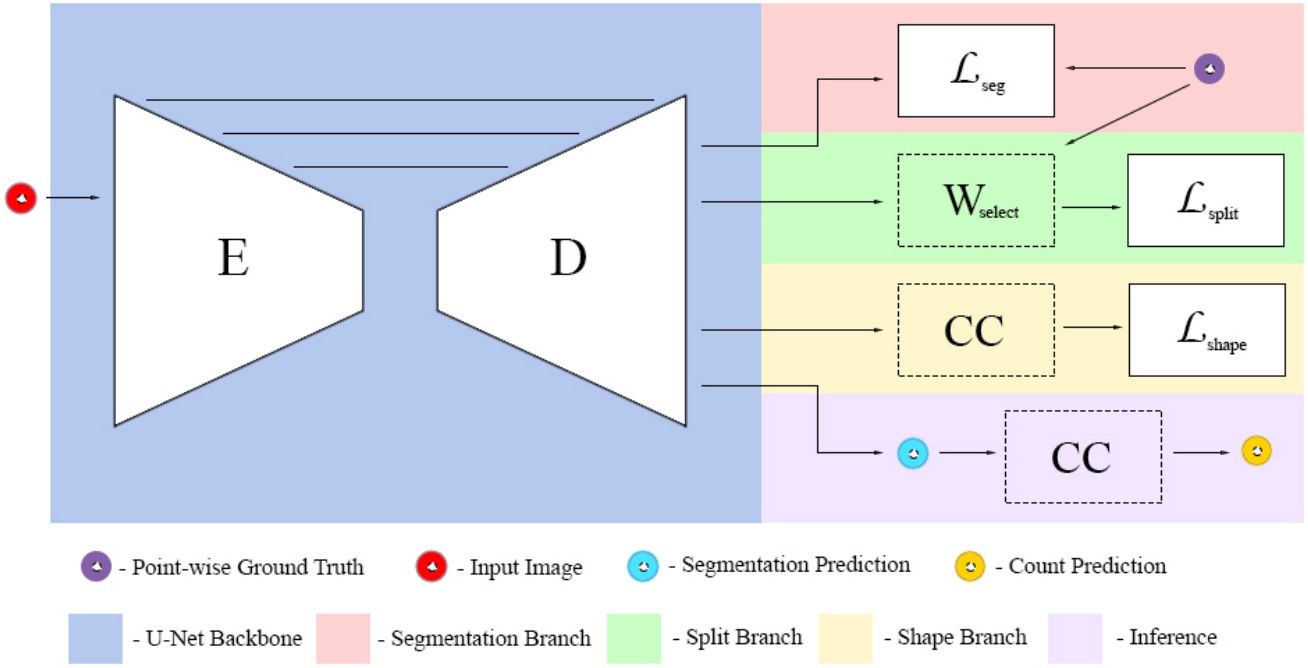


Figure 3: The network architecture of our proposed method. U-Net [41] with encoder E and decoder D is used as a backbone to generate masks guided by segmentation loss$\mathcal{L}_{Seg}$, split loss $\mathcal{L}_{Split}$ and shape loss $\mathcal{L}_{Shape}$. Our selective watershed algorithm, W_{select}, is used to better define expandable regions and boundaries in the predicted mask before computing the split loss. The shape loss branch first determines connected components, noted as CC, in the prediction mask before calculating individual shape loss. During inference, the predicted mask is obtained directly from the U-Net, and the count is calculated by the number of connected components present in the predicted segmentation.

starting at early bloom to post harvest. Drone trajectory is fixed throughout the collection season using initial randomly sampled path points at each cranberry bed. Before each recording session, a set of images of a checkerboard from different angels is captured for camera calibration purposes.

3.2. Annotation Procedure

We annotate 21,436 images with center points for training, and 702 images with pixel-wise annotations for testing and evaluation. All annotations are peer reviewed by other annotators through consensus, a process in which a given annotated image is passed to at least one more annotator for further labeling before it is submitted for a final review.

Center Point Annotations. Annotators are instructed to locate and tag cranberry center points, and equal number of background points. Background points are annotated at random locations, as far as possible from nearby cranberry annotations.

Berry-wise Annotations. Annotations follow two main guidelines: (1) only visible cranberries are annotated; (2) if a cranberry is occluded by leaves, the occluded parts are included resulting in a pixel-wise annotation that captures the shape of the occluded cranberry, hence termed *berry-wise* annotations. While order of visibility is not preserved during this annotation procedure, the annotations are instance-wise, which allows separability of objects if needed.

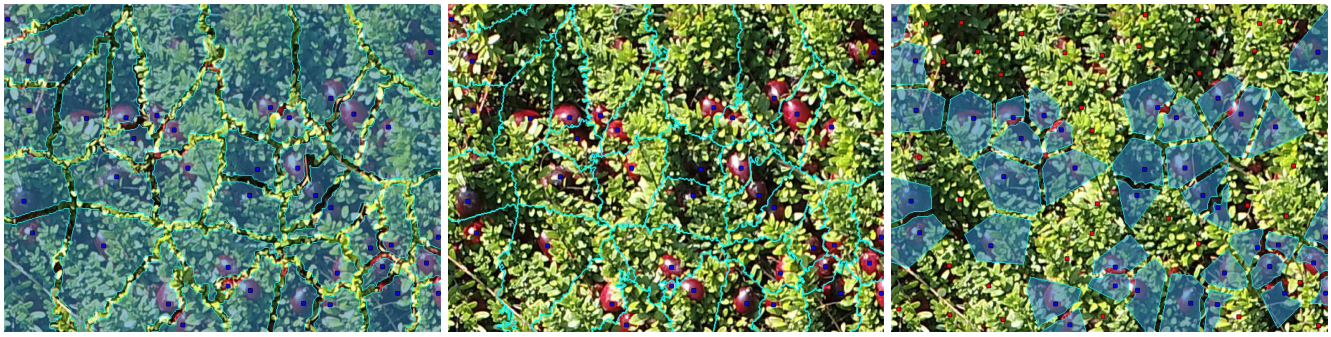
3.3. Dataset Details

CRAID has an average of 39.22 cranberries per image, with minimum count of 0 and maximum count of 167. Berry-wise annotated images have an average of 33.72% pixel cover. Average point-wise annotation time for a single image is 4.32 minutes, while berry-wise annotations take 22.13 minutes. Our relatively high annotation time compared to average estimated time reported in [4] is mainly caused by image complexity and high object counts.

4. Triple-S Network

Our approach is built upon U-Net [41] and consists of three branches: segmentation, split, and shape, constructing our proposed Triple-S Network illustrated in Figure 3. The segmentation branch aims to provide overall segmentation loss against point ground truth. The split and shape branches separate and refine individual blobs in segmentation outputs in accordance to boundaries and shape priors. The overall loss function is defined by

$$\begin{aligned}\mathcal{L}(X,Y) =& \lambda_0 \mathcal{L}_{\text{Seg}}(X,Y) + \lambda_1 \mathcal{L}_{\text{Split}}(X,Y) \\ &+ \lambda_2 \mathcal{L}_{\text{Shape}}(X,Y).\end{aligned} \tag{1}$$



(a) (b) (c)

Figure 4: Visual comparison between (a) watershed [35], (b) split watershed used in [27], and (c) our selective watershed. Highlighted pixels are considered in their respective method. Our method utilizes negative ground truth points (also called background) to generate more selective information better suited for learning. Note that (b) only considers the set of pixels representing the borders generated by the split watershed. Blue and red markers represent positive and negative ground truth points. Best viewed in color and zoomed.

Where X and Y represent the set of input images and point annotations respectively, and λ_* represents the weights of proposed losses. We define y as the set of ground truth points (from berry-wise annotations) and $\tilde{y}$ as the predicted mask of image x.

4.1. Segmentation Branch

$\mathcal{L}_{Seg}$ aims to encourage the model towards correct blob localization guided by positive point annotations. Let $\tilde{s}$ be the softmax probability prediction output for image x. Let p_n, $p_p \in y$ denote the positive and negative ground truth points, respectively. We then define the segmentation loss by

$$\mathcal{L}_{Seg}(\tilde{s}, y) = -\sum_{p_p} log(\tilde{s}) - \sum_{p_n}(1 - log(\tilde{s})). \quad (2)$$

4.2. Split Branch

The split loss function serves two purposes: discourage overlapping instances, and define expansion direction for predicted segments. We define the selective watershed algorithm $\mathrm{W}_{\text{select}}$, a modification of W [35], to utilize both positive and negative markers to produce background and object specific regions. Using $\mathrm{W}(y)$, we obtain a set R of distinct regions for all ground truth points. We can then find the set of positive regions as $r_p = \{r \in R : p_p \cap r \neq \varnothing\}$. The set of negative regions r_n is defined similarly and $r_p \cap r_n = \varnothing$. Figure 4 visualizes the set of pixels considered in our method compared to other variations of the watershed algorithm. We apply $\mathrm{W}_{\text{select}}$ on $\tilde{y}$ with y as markers to obtain r_p, the set of pixels representing the regions each instance can expand to without stepping onto other instances. r_p is then passed through an erosion algorithm [19] to better distinguish instances' boundaries. The complete split loss function is

$$\begin{aligned}\mathcal{L}_{Split}(\tilde{y}, y) = &-\sum_{r_p} \mathcal{E}(\mathrm{W}_{select}(\tilde{y}, y)) \\ &-\sum_{r_n} \mathcal{E}(\mathrm{W}_{select}(\tilde{y}, y)).\end{aligned} \quad (3)$$

$\mathrm{W}_{\text{select}}$ represents the selective watershed algorithm, and $\mathcal{E}$ represents the erosion algorithm [19].

4.3. Shape Branch

This work examines two shape priors appropriate for cranberries: convexity and circularity. These priors are used to guide the model towards meaningful structuring of the predicted blobs to fit berry-wise ground truth.

Convexity Loss. Let B represent the set of all blobs (distinct contiguous set of pixels) in y detected using the connected components algorithm [11] and let $b \in B$ denote an individual blob. Similarly, let $\tilde{B}$ be the set of blobs detected in $\tilde{y}$ with $\tilde{b} \in \tilde{B}$ denoting an individual blob. We can define a convexity measure of a predicted blob as the ratio between the blob area to its convex hull as follows

$$\mathcal{C}(\tilde{b}) = \frac{area(\tilde{b})}{area(ConvexHull(\tilde{b}))}. \quad (4)$$

Since objects in our dataset are always circular or elliptical when accounting for occlusion, the area enclosed by the predicted blob should always match or almost match the area enclosed by its convex hull, meaning our convexity measurement for each ground truth blob $b \in B$ is always close to one. The convexity loss $\mathcal{L}_{Convex}$ general form is given by

$$\mathcal{L}_{Convex}(\tilde{y}, y) = \frac{1}{|\tilde{B}|} \sum_{\tilde{b} \in \tilde{B} \in \tilde{y}, b \in B \in y} z(\tilde{b}, b), \quad (5)$$

where

$$z(\tilde{b}, b) = \begin{cases} \frac{1}{2}(\mathcal{C}(\tilde{b}) - \mathcal{C}(b))^2, \\ \quad \text{if } |\mathcal{C}(\tilde{b}) - \mathcal{C}(b)| < 1 \\ |\mathcal{C}(\tilde{b}) - \mathcal{C}(b)| - \frac{1}{2}, \text{ otherwise.} \end{cases} \tag{6}$$

This is a variation of Huber loss [21], in which the gradient is calculated with respect to the residual (supplementary for details). Note that $|\tilde{B}|$ represents the cardinality of that set.

Circularity Loss. Another approach is to directly find the circularity difference between the predicted blob $\tilde{b} \in \tilde{B}$ and ground truth $b \in B$. We formulate our loss function similar to the unconstrained nonlinear programming problem formulation proposed by [36], looking to find the least square reference circle (notated as LSC) of a given blob. Let r_b be the set of radii originating at the center of blob b. The circularity measurement of blob b is then defined by the difference between the maximum and minimum radii that exist in r_b. The reference circle LSC is given by

$$\begin{aligned} LSC(b) = &\max(\sqrt{(u_i - u_c)^2 + (v_i - v_c)^2}) \\ &- \min(\sqrt{(u_i - u_c)^2 + (v_i - v_c)^2}) \forall i \in b, \end{aligned} \tag{7}$$

where (u_i, v_i) are the coordinates of pixel i in blob b with center (u_c, v_c). Coordinates are with respect to the input image x. The final circularity loss formulation is defined as

$$\mathcal{L}_{Circ}(\tilde{y}, y) = \frac{1}{|\tilde{B}|} \sum_{\tilde{b} \in \tilde{B} \in \tilde{y}, b \in B \in y} z(\tilde{b}, b), \tag{8}$$

with

$$z(\tilde{b}, b) = \begin{cases} \frac{1}{2}(LSC(\tilde{b}) - LSC(b))^2, \\ \quad \text{if } |LSC(\tilde{b}) - LSC(b)| < 1 \\ |LSC(\tilde{b}) - LSC(b)| - \frac{1}{2}, \text{ otherwise.} \end{cases} \tag{9}$$

Since we want to predict circular or close to circular blobs, our ground truth $LSC(b)$ is zero, encouraging minimal difference between maximum and minimum radii.

4.4. Count Branch

This section explains the count loss used for ablation study, in which we explore the contribution and advantage of direct count learning with and without the usage of shape priors. $\mathcal{L}_{Count}$ aims to directly guide the model towards the correct number of instances, c, present in image x. This branch first separates blobs B from the segmentation prediction using connected components algorithm [11] noted as CC, and the resulting connected components count $\tilde{c}$ is used as count prediction. This means that small regions present in the segmentation prediction results high count prediction, which penalizes the model and discourages it from such predictions. More formally

$$\mathcal{L}_{Count}(\tilde{C}, C) = \frac{1}{|C|} \sum_{c \in C, \tilde{c} \in \tilde{C}} z_c \,, \tag{10}$$

where

$$z_c = \begin{cases} \frac{1}{2}(\tilde{c} - c)^2, & \text{if } |\tilde{c} - c| < 1 \\ |\tilde{c} - c| - \frac{1}{2}, & \text{otherwise,} \end{cases} \tag{11}$$

Here, $\tilde{c} = |CC(\tilde{y})|$. C and $\tilde{C}$ are the ground truth and predicted counts for all samples. $\mathcal{L}_{count}$ is small if the difference between $\tilde{c}$ and c is small.

5. Experiments

5.1. Implementation Details

Network Architecture. Since we want to highlight the contribution of shape priors and boundary setting, we choose to adopt a standard fully convolutional network (FCN) introduced at [41]. The network consists of an encoder with eight blocks, each consists of two convolution layers followed by batch normalization and rectified linear unit (ReLU) layers. After each block we apply a 2×2 max pooling layer with a stride of 2. The encoder captures 3 channel inputs, and yields 1024 channel output. The decoder is also formed with eight blocks, each consists of feature map upsampling, two up convolution layers which halve the number of channels followed by batch normalization and ReLU layers. The output at each decoder block is concatenated to the corresponding encoder block. At the final layer, we use a 1×1 convolution layer to map 64 channel output to the number of classes.

Training and Evaluation Setup. We train our network from scratch using 90/5/5 data split with Adam optimizer [26], starting learning rate of 0.001, and cosine annealing scheduler [32]. Random flips and normalization transforms are applied to the training input. We let the networks train on a single NVIDIA GTX 1080 for 25,000 iterations or until convergence, whichever comes first. For metrics, we report Mean Absolute Error (MAE) for counting and Mean Intersection over Union (mIoU) for segmentation. MAE score calculates the sum of absolute differences between count ground truth and predictions, divided by the number of examples. Count predictions are found using the number of connected components [11] computed on the segmentation prediction mask. mIoU computes the ratio between intersection and union between prediction and ground truth masks. We also consider the inverse MAE to mIoU ratio Q_{cs} to indicate how well a model does on both metrics. Q_{cs} measures the trade-off between counting and segmentation performances, and is used as a joint performance indicator. During training, models with best MAE, best mIoU, and best Q_{cs} are saved. For models that incorporate segmentation, validation and testing is preformed on fully supervised images, and models are chosen using the best Q_{cs}. For counting specific methods, models are selected based on the best MAE results. Formal formulation of reported

Training Ground Truth	Method \ Metric	mIoU (%)	MAE	Q_{cs}
Pixel-wise	U-Net [41]	78.56	9.60	8.19
Points	U-Net [41]	60.61	18.67	3.25
	[40]	-	39.22	-
	m[40]	-	21.76	-
	LC-FCN [27]	61.97	17.46	3.55
Points	Ours ($\mathcal{L}_{Seg}$ + $\mathcal{L}_{Split}$)	67.39	16.33	4.13
	Ours ($\mathcal{L}_{Seg}$ + $\mathcal{L}_{Split}$ + $\mathcal{L}_{Count}$)	62.54	**13.46**	4.65
	Ours ($\mathcal{L}_{Seg}$ + $\mathcal{L}_{Split}$ + $\mathcal{L}_{Circ}$)	67.85	14.25	**4.76**
	Ours ($\mathcal{L}_{Seg}$ + $\mathcal{L}_{Split}$ + $\mathcal{L}_{Circ}$ + $\mathcal{L}_{Count}$)	65.89	14.31	4.60
	Ours ($\mathcal{L}_{Seg}$ + $\mathcal{L}_{Split}$ + $\mathcal{L}_{Convex}$)	**68.71**	15.90	4.32
	Ours ($\mathcal{L}_{Seg}$ + $\mathcal{L}_{Split}$ + $\mathcal{L}_{Convex}$ + $\mathcal{L}_{Count}$)	65.35	15.93	4.10

Table 1: Mean Intersection over Union (%) accuracy (higher is better), Mean Average Error (MAE) (lower is better), and Inverse MAE to mIoU ratio Q_{cs} (higher is better) metrics on CRAID. Our proposed method outperforms the SOTA (trained with point annotations) in all metrics. All evaluation is done against pixel-wise ground truth.

metrics are defined by

$$\begin{aligned} \text{MAE} &= \frac{1}{n}\sum_{i=1}^{n} |\tilde{c}_i - c_i| \,, \\ \text{mIoU} &= \frac{1}{n}\sum_{i=1}^{n} \frac{y_i \cap \tilde{y}_i}{y_i \cup \tilde{y}_i} \,, \\ Q_{cs} &= \frac{1}{\text{MAE}} * \text{mIoU}. \end{aligned} \tag{12}$$

Where c, $\tilde{c}$ represent true and predicted counts in image i, and n indicates the number of examples in the dataset.

5.2. Baselines

We compare our method to SOTA in counting [40], joint counting and segmentation [27], and semantic segmentation algorithms [41]. All baselines were trained from scratch to ensure fair comparison. The original formulation in [40] was unable to learn meaningful counts in our data caused by a ReLU layer at the regressor branch that zeros estimated counts. Instead, we modify [40] (referred to as m[40]) by using a Parametric ReLU, which learns an additional parameter to better handle negative values. It is possible that further tuning of hyperparameters is necessary for better performance. U-Net with point supervision baseline was trained with adjusted class weights to allow better learning. Also, important to note that we recognize that LC-FCN [27] does not aim to segment images, but since its approach is similar enough and the lack of other comparable works, we slightly modified its code to output segmentation masks and included results in both metrics.

5.3. Results

Table 1 presents comparisons between baselines and our method for counting and segmentation metrics. As can be observed, our method outperforms [40], m[40], and LC-FCN [27] in those metrics. We see superior counting performance against [40], which proved unable to correctly count cranberries in CRAID images, although it performed significantly better on other datasets. The comparison to LC-FCN can also be seen in Figure 5, where better separability between objects results in better counting, and more accurate shape results in better segmentation performance. Notice that the segmented blobs maintain elliptical shapes, compared to irregular shaped blobs produced by [27].

5.4. Ablation

We explore the contribution of each added module in our proposed method and compare them to the SOTA methods in counting and segmentation. We find that using known shape priors as a blob structuring indicator dramatically improve segmentation performance. While using $\mathcal{L}_{Convex}$ shows better results on segmentation, $\mathcal{L}_{Circ}$ provides better outcome overall with the highest inverse MAE to mIoU ratio. Table 1 also shows that adding a count loss to segmentation and split losses boosts counting precision but degrades segmentation performance. The results also show that count loss always degrades overall results when paired with shape priors. We also examine how shape cues compare to color cues for our network. Typically, color cues are strong indicators in similar objects, which is a challenge in agriculture applications as color is a dynamic feature varying between seasons. The last row of Figure 5 shows how the network handles leaves around cranberries that redden

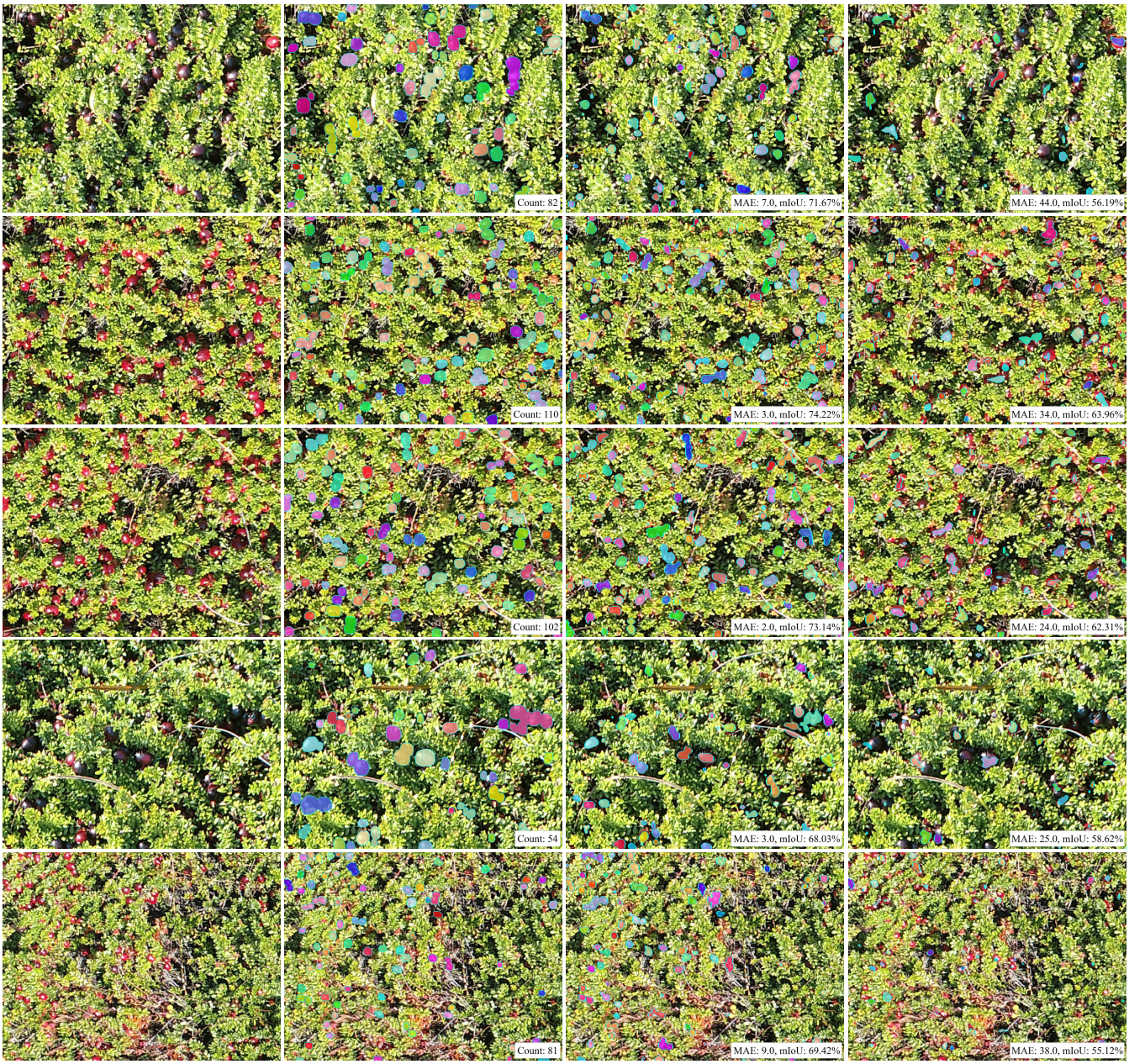


Figure 5: Qualitative comparison with SOTA methods on CRAID. Our method ($\mathcal{L}_{Seg}$ + $\mathcal{L}_{Split}$ + $\mathcal{L}_{Convex}$) shows that using shape priors and better boundary and region selection allows robust segmentation and counting performance. Colors in prediction masks are random and are used to represent instances. Colors may repeat. Best viewed in color and zoomed.

during late fruit ripening period. It can be seen that while there are many red leaves in the scene, the majority are predicted as background by the network.

5.5. Conclusion

In this paper, we present a novel approach to count and segment objects utilizing point supervision and shape priors. We propose the Triple-S network that employs our selective watershed algorithm, and shape loss functions to encourage convex and circular object masks. We present a first of its kind publicly available dataset and software toolkit for supporting precision agriculture in cranberry fields. The approach can be extended to other crops such as blueberries, grapes, and olives.

Acknowledgements This project was sponsored by the USDA NIFA AFRI Award Number: 2019-67022-29922. We thank David Nuhn who assisted in data collection. We acknowledge Aditi Roy at Siemens Corporate for conversations on segmentation baselines.

References

[1] Johanna Albetis, Sylvie Duthoit, Fabio Guttler, Anne Jacquin, Michel Goulard, Hervé Poilvé, Jean-Baptiste Féret, and Gérard Dedieu. Detection of flavescence dorée grapevine disease using unmanned aerial vehicle (uav) multispectral imagery. *Remote Sensing*, 9(4):308, 2017. 2

[2] Suchet Bargoti and James Underwood. Deep fruit detection in orchards. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017. 1

[3] Suchet Bargoti and James P. Underwood. Image segmentation for fruit detection and yield estimation in apple orchards. *Journal of Field Robotics*, 34(6):1039–1060, Sep 2017. 1, 2

[4] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the point: Semantic segmentation with point supervision. In *European conference on computer vision*, pages 549–565. Springer, 2016. 2, 4

[5] R Calderón, Juan Antonio Navas-Cortés, C Lucena, and Pablo J Zarco-Tejada. High-resolution airborne hyperspectral and thermal imagery for early detection of verticillium wilt of olive using fluorescence, temperature and narrowband spectral indices. *Remote Sensing of Environment*, 139:231–245, 2013. 2

[6] K. G. Cassman. Ecological intensification of cereal production systems: Yield potential, soil quality, and precision agriculture. *Proceedings of the National Academy of Sciences*, 96(11):5952–5959, 1999. 2

[7] N. Zhang C. Chaisattapagon. Effective criteria for weed identification in wheat fields using machine vision. *Transactions of the ASAE*, 38(3):965–974, 1995. 2

[8] Ovidiu Csillik, John Cherbini, Robert Johnson, Andy Lyons, and Maggi Kelly. Identification of citrus trees from unmanned aerial vehicle imagery using convolutional neural networks. *Drones*, 2(4):39, 2018. 2

[9] Jifeng Dai, Kaiming He, and Jian Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1635–1643, 2015. 2

[10] Salvatore F Di Gennaro, Enrico Battiston, Stefano Di Marco, Osvaldo Facini, Alessandro Matese, Marco Nocentini, Alberto Palliotti, and Laura Mugnai. Unmanned aerial vehicle (uav)-based remote sensing to monitor grapevine leaf stripe disease within a vineyard affected by esca complex. *Phytopathologia Mediterranea*, pages 262–275, 2016. 2

[11] Michael B Dillencourt, Hanan Samet, and Markku Tamminen. A general approach to connected-component labeling for arbitrary image representations. *Journal of the ACM (JACM)*, 39(2):253–280, 1992. 5, 6

[12] M. S. El-Faki, N. Zhang, and D. E. Peterson. Weed detection using color machine vision. *Transactions of the ASAE*, 43(6):1969–1978, 2000. 2

[13] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010. 2

[14] F. Feyaerts and L. Van Gool. Multi-spectral vision system for weed detection. *Pattern Recognition Letters*, 22(6-7):667–674, 2001. 2

[15] E. Franz, M. R. Gebhardt, and K. B. Unklesbay. The use of local spectral properties of leaves as an aid for identifying weed seedlings in digital images. *Transactions of the ASAE*, 34(2):0682–0687, 1991. 2

[16] Francisco Garcia-Ruiz, Sindhuja Sankaran, Joe Mari Maja, Won Suk Lee, Jesper Rasmussen, and Reza Ehsani. Comparison of two aerial imaging platforms for identification of huanglongbing-infected citrus trees. *Computers and Electronics in Agriculture*, 91:106–115, 2013. 2

[17] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 2

[18] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 2

[19] Robert M Haralick, Stanley R Sternberg, and Xinhua Zhuang. Image analysis using mathematical morphology. *IEEE transactions on pattern analysis and machine intelligence*, pages 532–550, 1987. 5

[20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2

[21] Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics*, pages 492–518. Springer, 1992. 6

[22] E Raymond Hunt and Silvia I Rondon. Detection of potato beetle damage using remote sensing from small unmanned aircraft systems. *Journal of Applied Remote Sensing*, 11(2):026013, 2017. 2

[23] R Kerry, P Goovaerts, Daniel Gimenez, and PV Oudemans. Investigating temporal and spatial patterns of cranberry yield in new jersey fields. *Precision Agriculture*, 18(4):507–524, 2017. 1

[24] Ramesh Kestur, Avadesh Meduri, and Omkar Narasipura. Mangonet: A deep semantic segmentation architecture for a method to detect and count mangoes in an open orchard. *Engineering Applications of Artificial Intelligence*, 77:59–69, 2019. 1, 2

[25] Anna Khoreva, Rodrigo Benenson, Jan Hosang, Matthias Hein, and Bernt Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 876–885, 2017. 2

[26] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[27] Issam H Laradji, Negar Rostamzadeh, Pedro O Pinheiro, David Vazquez, and Mark Schmidt. Where are the blobs: Counting by localization with point supervision. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 547–562, 2018. 3, 5, 7

[28] Issam H Laradji, David Vazquez, and Mark Schmidt. Where are the masks: Instance segmentation with image-level supervision. *arXiv preprint arXiv:1907.01430*, 2019. 2

[29] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 2

[30] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[31] Manuel López-López, Rocío Calderón, Victoria González-Dugo, Pablo J Zarco-Tejada, and Elías Fereres. Early detection and quantification of almond red leaf blotch using high-resolution hyperspectral and thermal imagery. *Remote Sensing*, 8(4):276, 2016. 2

[32] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 6

[33] Philipp Lottes, Jens Behley, Andres Milioto, and Cyrill Stachniss. Fully convolutional networks with sequential information for robust crop and weed detection in precision farming. *IEEE Robotics and Automation Letters*, 3(4):2870–2877, 2018. 1, 2

[34] Wouter H Maes and Kathy Steppe. Perspectives for remote sensing with unmanned aerial vehicles in precision agriculture. *Trends in plant science*, 24(2):152–164, 2019. 1

[35] Fernand Meyer. Topographic distance and watershed lines. *Signal Processing*, 38(1):113–125, 1994. 2, 5

[36] TSR Murthy and SZ Abdin. Minimum zone evaluation of surfaces. *International Journal of Machine Tool Design and Research*, 20(2):123–136, 1980. 6

[37] V Pelletier, S Pepin, J Gallichand, and J Caron. Reducing cranberry heat stress and midday depression with evaporative cooling. *Scientia horticulturae*, 198:445–453, 2016. 1

[38] A.j. Pérez, F. López, J.v. Benlloch, and S. Christensen. Colour and shape analysis techniques for weed detection in cereal fields. *Computers and Electronics in Agriculture*, 25(3):197–212, 2000. 2

[39] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 2

[40] Javier Ribera, David Guera, Yuhao Chen, and Edward J Delp. Locating objects without bounding boxes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6479–6489, 2019. 3, 7

[41] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 4, 6, 7

[42] Teryl Roper. The physiology of cranberry yield. *Wisconsin Cranberry Crop Management Newsletter Vol. XIX*, 2006. 1

[43] Chunfeng Song, Yan Huang, Wanli Ouyang, and Liang Wang. Box-driven class-wise region masking and filling rate guided loss for weakly supervised semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3136–3145, 2019. 2

[44] Y. Song, C.a. Glasbey, G.w. Horgan, G. Polder, J.a. Dieleman, and G.w.a.m. Van Der Heijden. Automatic fruit recognition and counting from multiple images. *Biosystems Engineering*, 118:203–215, 2014. 1, 2

[45] John V. Stafford. Implementing precision agriculture in the 21st century. *Journal of Agricultural Engineering Research*, 76(3):267–275, 2000. 2

[46] Everton Castelão Tetila, Bruno Brandoli Machado, Nícolas Alessandro de Souza Belete, David Augusto Guimarães, and Hemerson Pistori. Identification of soybean foliar diseases using unmanned aerial vehicle images. *IEEE Geoscience and Remote Sensing Letters*, 14(12):2190–2194, 2017. 2

[47] Els Vrindts and Josse De Baerdemaeker. Weed detection using canopy reflection. *Precision Agriculture and Biological Quality*, 1999. 2

[48] Jialin Yu, Shaun M Sharpe, Arnold W Schumann, and Nathan S Boyd. Deep learning for image-based weed detection in turfgrass. *European journal of agronomy*, 104:78–84, 2019. 2

[49] Feng Zheng and H. Schreier. Quantification of soil patterns and field soil fertility using spectral reflection and digital processing of aerial photographs. *Fertilizer Research*, 16(1):15–30, 1988. 2

[50] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[51] Yanzhao Zhou, Yi Zhu, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. Weakly supervised instance segmentation using class peak response. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3791–3800, 2018. 2