IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON NEW TECHNOLOGIES FOR SMART FARMING 4.0: RESEARCH CHALLENGES AND OPPORTUNITIES

Received July 13, 2019, accepted July 24, 2019, date of publication August 5, 2019, date of current version August 15, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2933062*

# Deepblueberry: Quantification of Blueberries in the Wild Using Instance Segmentation

**SEBASTIAN GONZALEZ[1], CLAUDIA ARELLANO[2], (Member, IEEE), AND JUAN E. TAPIA[1,2], (Member, IEEE)**

[1]Department of Engineering Sciences, Universidad Andres Bello, Santiago 7500971, Chile
[2]Research and Development Department, Universidad Tecnologica de Chile—INACAP, Santiago 8580704, Chile

Corresponding author: Juan E. Tapia (juan.tapia2043@uandresbello.edu; j_tapiaf@inacap.cl)

This work was supported in part by the Universidad Andres Bello, and in part by the Universidad Tecnologica de Chile—INACAP.

**ABSTRACT** An accurate and reliable image-based quantification system for blueberries may be useful for the automation of harvest management. It may also serve as the basis for controlling robotic harvesting systems. Quantification of blueberries from images is a challenging task due to occlusions, differences in size, illumination conditions and the irregular amount of blueberries that can be present in an image. This paper proposes the quantification per image and per batch of blueberries in the wild, using high definition images captured using a mobile device. In order to quantify the number of berries per image, a network based on Mask R-CNN for object detection and instance segmentation was proposed. Several backbones such as ResNet101, ResNet50 and MobileNetV1 were tested. The performance of the algorithm was evaluated using the Intersection over Union Error (IoU) and the competitive mean Average Precision (mAP) per images and per batch. The best detection result was obtained with the ResNet50 backbone achieving a mIoU score of 0.595 and mAP scores of 0.759 and 0.724 respectively (for IoU thresholds 0.5 and 0.7). For instance segmentation, the best results obtained were 0.726 for the mIoU metric and 0.909 and 0.774 for the mAP metric using thresholds of 0.5 and 0.7 respectively.

**INDEX TERMS** Blueberries, deep learning, quantification, segmentation.

## I. INTRODUCTION

Blueberries possess a blue to blue-black epidermis that is covered by a waxy bloom, giving the fruit a light blue appearance. High quality blueberries, free from injury, decay, and sun scald, are fully blue in colour with little to no red at the stem end, as well as feeling and appearing turgid. In order to be sold in the fresh market, the fruit should be fully blue and firm.

Blueberry clusters are not usually harvested entirely at the same time as the clusters usually contain fruits in different growth stages simultaneously. This make managing the harvesting process critical since blueberries must arrive at the market fully ripened and in good condition. This is particularly difficult for blueberry producers from the southern hemisphere such as Chile, the second largest producer of blueberries in the global market, since most of its production is exported to consumer markets as far away as the United States, China, Australia and Europe. This logistical challenge makes harvesting management very critical.

In order to develop automatic harvesting systems and to support accurate harvest management decisions, image-based blueberry quantification systems must be developed. Previous image based algorithms have focused primarily on the use of texture features to extract fruit information. Recent work has explored the use of state of the art techniques such as deep learning for fruit detection and classification. However, these systems have focused on a limited variety of fruits and blueberries have not been considered. This may be due, in part, to the lack of an annotated blueberry database to train such algorithms. Labelling databases is a time-consuming process, as there is significant effort involved manually labelling the ground truth database.

In this paper, we propose an instance segmentation algorithm to quantify blueberries from images captured in the wild. The contribution of this paper may be summarised as follows:

Two novel databases of blueberry images were captured and will be made available for researchers upon request.

The associate editor coordinating the review of this manuscript and approving it for publication was Kun Mean Hou.

1) Blueberry-V1: A database of 293 images were captured in the wild under challenging conditions such as variable illumination, partial and full occlusions, variations in brightness, contrast, shadows, and the presence of external objects such as people, hands and leaves.
2) Annotated database Blueberry-V2: A total of 10,161 blueberries were annotated for detection (as a bounding box) and 228 for segmentation. The Annotation (object coordinates) to be read by the Mask R-CNN [15] algorithm was modified in order to standardise its format and make it compatible with other manual markers.

Deep Blueberry algorithm: An instance segmentation algorithm based on Mask R-CNN was implemented [15]. This implementation differs from the original as it is shown as follows:

- Architecture: In order to achieve a lighter implementation than the original algorithm [15] several backbones such as ResNet50 [16] and MobileNetV1 [19] were tested. The latter allows the algorithm to run on a mobile device. In addition, the anchor scale and mini-mask shape were modified in order to detect small blueberries in the image. The detection of small fruits is particularly difficult since they can be confused with other structures in the image such as leaves or branches. The standard Mask-cnn is not able to detect such small structures.
- Database used for training: In this implementation the backbones were trained from scratch using the novel annotated database built in this work. The standard implementation of the algorithm was trained using the COCO database which contains images from different classes such as objects from cities, furniture and fruits among others. However, it is not able to detect small fruits such as blueberries. The only fruits included in this database are bananas, apples and oranges.
- Evaluation metrics: Additional IoU and @Ap metrics computed for each image (instead of the average IoU for the whole database) were introduced. These metrics allow the performance of the algorithm for each image to be understood. Facilitating the analysis of the errors per image.
- Blueberry quantification algorithm: The proposed method includes an additional step that allows the number of blueberries segmented in each image to be quantified. This is the final output of the algorithm which could be used for improving the management of the harvesting process.

The remainder of the paper is organised as follows: Section II provides some background concepts in image processing and deep learning. Section III describes the proposed method, including the developed databases and the evaluation metrics. Experiments and Results are reported in section IV. Conclusions are presented in section VI.

## II. STATE OF THE ART

Automated systems based on computer vision for the classification and detection of images of blueberries and other crops in orchards have been previously proposed in the literature. Consequently, the state of the art in instance segmentation algorithms and computer vision techniques applied to fruit image analysis are reviewed in subsections II-A and II-B, respectively.

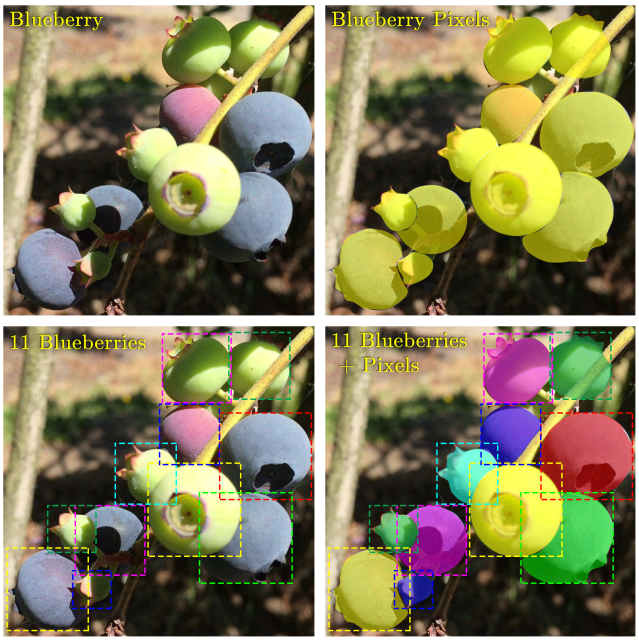


**FIGURE 1. Examples of computer vision tasks. Top-left shows image classification, top-right semantic segmentation, bottom-left object detection, and bottom right-instance segmentation.**

### A. INSTANCE SEGMENTATION ALGORITHMS

Computer vision techniques have greatly improved in recent years, mainly due to the development of neural network techniques such as deep learning [24]. Several Deep Learning algorithms have been applied to a wide range of fields, such as medical image analysis, biometrics, scene understanding, autonomous driving, amongst others [1], [2], [34]. The most common applications of deep learning for image analysis are image classification, object detection, semantic segmentation and instance segmentation (see Figure 1) [5], [22]. Image classification has the goal of categorising an image into a particular class, returning the corresponding label and the classification confidence rate. In object detection, the goal is to localise regions of interest and classify each one individually, resulting in the class label for each object in the image and also their coordinates denoted by a bounding box. Semantic segmentation has the purpose of classifying each pixel of the image and group them according to their class. The task of instance segmentation can be thought of as object detection with the addition of semantic segmentation, where the goal is to detect each object in the image and classify each pixel of every instance. In contrast to object detection, the output of an instance segmentation algorithm is a mask around the object of interest instead of a bounding box.

Instance segmentation, in particular, has drawn increasing interest from the computer vision community [10]. This task

in particular proves to be challenging since it requires the correct detection of all objects in the image while simultaneously segmenting each instance of the object in a precise manner. Algorithms such as Fast/Faster R-NN [11], and Fully Convolutional Networks (FCN) [28] have been used as baseline systems for detection and semantic segmentation, respectively. The R-CNN algorithm allows the detection of the bounding boxes for all candidate object regions, which are then used as Regions Of Interests (ROI) for segmenting each instance. Most methods relying on R-CNN use segment proposals to achieve instance segmentation [12]–[14], [17]. Pinheiro *et al.* [30], [31] proposed an algorithm (DeepMask) that learnt the segment proposals and then classified them using Fast R-CNN. Dai *et al.* [6] proposed a complex multiple-stage cascade to predict the segment proposals from the bounding boxes. An algorithm that combined the segment proposal approach and object detection algorithms was proposed by Li *et al.* [26]. They used a FCN [7] to achieve the instance segmentation. He *et al.* [15] have proposed the Mask R-CNN algorithm for instance segmentation which is an extension of Faster R-CNN. It includes a branch for segmentation mask prediction in each ROI. This mask branch is a small FCN that is applied to each ROI predicting a segmentation mask in a pixel to pixel manner. This branch works in parallel with the branch for classification and bounding box regression tasks. This algorithm has shown good results in challenges such as object detection, instance segmentation and human keypoint detection. It is considered the state-of-the-art algorithm for instance segmentation applications.

### B. APPLICATIONS TO FRUIT IMAGE ANALYSIS

Computer vision techniques have been used to analyse fruit crop images, addressing several problems such as detection, classification and maturity estimation [25], [35] amongst others [18], [21], [27], [32].

Maturity stage classification of blueberries using outdoor color images was proposed by Li *et al.* [25]. The K-nearest neighbour (KNN), naive Bayesian classification (NBC) and ''supervised K-means clustering classifier (SK-means)'' algorithms were evaluated for the identification of mature, near-mature, near-young and young berries. They achieved accuracy scores of 0.96, 0.95 and 0.89 for the KNN, NBC, and SK-means classifiers respectively. However, they only used 46 images (23 images for training and 23 for validation). Similar studies, using hyperspectral images (40 in total), were reported by Yang *et al.* [35]. The most useful spectral bands are selected based on the Kullback-Leibler Divergence metric (KLD). Three classifiers: KNN, Support Vector Machine (SVM) and AdaBoost were used to test the performance of the selected bands, achieving a classification accuracy of 0.88.

Fruit detection has been studied by Kuang *et al.* [23], where a multi-class detector strategy based on feature extraction on multiple colour channels was proposed. Histogram of Oriented Gradient (HOG), Local Binary Pattern (LBP), Gabor wavelets-based LBP (GaborLBP) and global colour histogram were used as feature extractors on multiple colour channels from RGB, HSV, HSI and Lab colour spaces. To produce optimal results, the features were fused and used to train a SVM classifier on previously divided blocks of the image. Subsequently, 5-fold cross validation was used to select the block with the highest accuracy for each colour channel. In order to reduce feature redundancy, an optimal synthesis of colour channels was also selected using cross validation accuracy by fusing channels one by one until an optimal condition was met. The training was performed using a dataset of five classes of fruit (red apple, orange, pear, kiwi and persimmon) plus background. For evaluation, a database of 1,778 images containing multiple fruit and multiple instances of fruit was used. The fusion of the four features extracted from the synthesis of the 8 colour channels yielded an accuracy of 0.972, with an Average Precision of 0.8135 and a detection time of 28 seconds per image.

Castro *et al.* [4] have recently explored the classification of Cape gooseberry fruit according to its maturity stage. They combined information from images using three colour spaces (RGB, HSV, and Lab) and four different classification algorithms including artificial neural networks, support vector machines (SVMs), decision trees, and K-nearest neighbour. Pacheco and López [29] have proposed the use of K-NN, MLP, and K-Means Clustering algorithm to classify tomatoes according to their organoleptic maturity (coloration).

The use of deep learning techniques for fruit classification was proposed by Sa *et al.* citeDeepFruit. They used a Fast Region based CNN detector pre-trained on a large database such as ImageNet [8]. Only a small number of images of each fruit category to be detected were used (Sweet peppers, Rock Melon, Apple, Avocado, Mango and Orange). They proposed a multimodal fusion approach that combined information from colour and Near Infra-red images. A similar process using a Faster R-CNN was proposed by Bargoti and Underwood [3] for detecting apples, mangoes and almonds. The apple and mango images (up to 100 samples of each fruit per image) were collected by a sensor located on a vehicle that toured the orchards, capturing daylight outdoor images. Almond images were captured using a hand-held camera (up to 1,000 samples of almonds per images).

A number of images with a pixel resolution of 1,616 × 1,232 were selected for training (729) and for validation and testing (224) for the apple class. For the mango class, images with a pixel resolution of 3,296 × 2,472 were used for training (1,154) and for validation and testing (540). For the almond class, images with a pixel resolution of 3,456 × 5,184 were used for training (385) and for validation and testing (200). Models were evaluated with and without data augmentation applied to the apple and mango datasets where the latter improved detection performance. In the case of apples, a transfer learning technique was used by initialising the weights with features learned from networks previously trained for mango and almond detection. ZFNet [38] and VGG-16 [33] were used as the backbone. The best $F_1$-scores obtained when using VGG-16 backbone for the apple, mango

**TABLE 1. Comparison with related works.**

| | Li, Lee, and Wang (2014) [25] | | | Yang, Lee, Gader, and Li (2013) [36] | | | Yang, Lee, and Gader (2014) [35] | | | DeepBlueBerry (This paper) | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **# Images** | 46 (Outdoor colour images) | | | 40 (Hyperspectral images) | | | 40 (Hyperspectral images) | | | 305 (10,161 objects) | 7 (228 objects) |
| **Method** | Classification | | | Classification | | | Classification | | | Mask R-CNN | |
| | NBC | KNN | SK-means | CART | KNN | RVM | KNN | SVM | AdaBoost | Object Detection | Instance Segmentation |
| **Score** | **0.56** | **0.83** | **0.52** | **0.915** | **0.994** | **0.939** | **0.987** | **0.958** | **0.984** | **0.759** | **0.909** |
| | **Average Accuracy** | | | **Accuracy** | | | **Overall Accuracy** | | | **mAP** | |

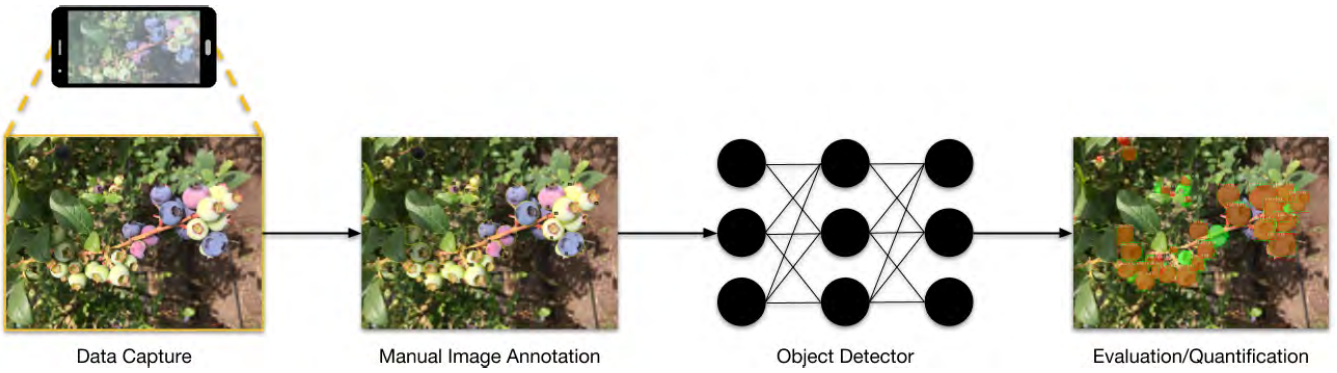


**FIGURE 2. General scheme of the work described in this paper. The data was captured in the wild using a mobile device and then manually annotated. The algorithm proposed includes detection and segmentation tasks. The final result is the quantification (instance segmentation) of the blueberries in the image.**

and almond detection were 0.904, 0.908 and 0.775 respectively. In the case when ZFNet was used the $F_1$-scores were 0.892, 0.876 and 0.726 respectively for each class of fruit. Detections were treated as true positives if the IoU value was greater than 0.2.

Recent work, have reported the use of vision techniques to detect and quantify fruits. Ponce *et al.* [20] have proposed an image analysis algorithm, based on mathematical morphology, that is able to individually segment olives and extract descriptive features to estimate their major and minor axes and their mass. This work aims at fruit grading which is a post-harvest task, where size-and-mass based fruit classification is especially important when processing high-quality table olives. Unfortunately, this algorithm only works in controlled environments and can not be replicated in the wild as is the scope of this paper.

Yu *et al.* [37], on the other hand, proposed a mask-cnn algorithm to detect and quantify strawberries in the wild. Fruit detection results of 100 test images showed an average detection precision rate of 95.78%, the recall rate was 95.41% and the mean intersection over union (MIoU) rate for instance segmentation was 89.85%. A similar algorithm is proposed in this paper but applied to blueberries. In this case additional challenges are faced such as the larger number of instances of the fruit in each image, the smaller size of the fruit and the similarity in colour of some berries with respect to the background (leaf and branches).

## III. PROPOSED METHOD

In this paper, a new instance segmentation algorithm for blueberry quantification based on Mask R-CNN is proposed. This algorithm has been trained from scratch using a novel database captured in the wild. The original Mask R-CNN implementation was designed for city object detection, such as, cars, buildings, and people. As such, it was not normally able to detect, segment and quantify blueberries in the wild. Figure 2 shows a flowchart of the proposed method. First, a novel blueberry image database was captured from the wild using a mobile device. A Mask R.-CNN algorithm, using multiple architectures was then implemented. For clarity, (i) the convolutional backbone architecture used for feature extraction over an entire image, and (ii) the network head for bounding-box recognition (classification) and mask prediction that is applied separately to each ROI of the image were differentiated. The last step of the proposed method outputs each instance of a blueberry allowing the total number of blueberries in the image to be quantified.

### A. IMAGE-CAPTURE AND DATABASE

According to the literature, there is no database of blueberries captured in wild conditions for object detection and/or instance segmentation that has been made available for research. Therefore, as a first step and contribution of this work, an entirely new database of high-resolution blueberry images (Blueberry-V1) was captured and annotated

(Blueberry-V2). These databases will be made available for researchers upon request.

**Blueberry-V1**: This database is composed of images and videos captured from a commercial blueberry farm in Los Angeles, Chile. An 8MP iPhone 6 camera with a focal length of 4.15mm and a maximum aperture of f/2.2 was used. The data was captured outdoors during daylight hours. A total of 84 images with a 3,264 × 2,448 pixel resolution and 4 videos with a 1,920 × 1,080 pixel resolution were captured. The videos were processed and turned into individual frames of the same resolution, out of which 293 frames were manually selected, taking data variability into consideration and discarding blurry images. All images present challenging conditions of illumination, partial and full occlusions, variations in brightness, contrast, shadows, and also external objects such as people, hands and leaves.

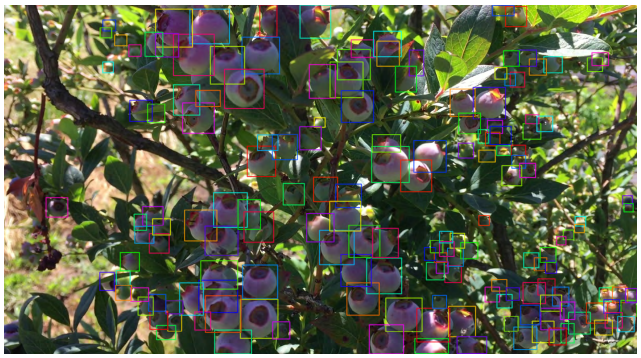

**FIGURE 3.** Image annotation example for object detection. The different bounding box colours are for better visualisation.

For the object detection database, ground truth annotation was done manually by drawing a bounding box on each blueberry in the image using self-developed software. A total of 10,161 blueberry annotations were made. Figure 3 illustrates the bounding boxes annotated for a single image. The annotated data was saved in a JSON file and contains the key-value pairs of each blueberry. Each key matches an image filename, and its paired value contains a list of elements denoting the bounding box for each blueberry. The key-value pairs (integer) are: "x", "y", "w", "h". Where (x, y) represent the coordinates of the upper-left vertex of the bounding box and (w, h) represent the coordinates of the lower-right vertex of the same bounding box.

**Blueberry-V2**: For the semantic segmentation database, the ground truth annotation consisted of carefully drawing the edges of each blueberry in the image. This was done using the VGG Image Annotator tool [9], where a total of 7 images with 228 blueberries in total were annotated. The smaller number of annotated segmented examples in comparison with the bounding box annotation is due to the manual nature of this task. Figure 4 illustrates the blueberry edges annotated for a single image.

### B. NETWORK ARCHITECTURE

The instance segmentation algorithm proposed in this paper is based on the Mask R-CNN algorithm the architecture of

**FIGURE 4.** Image annotation example for instance segmentation.

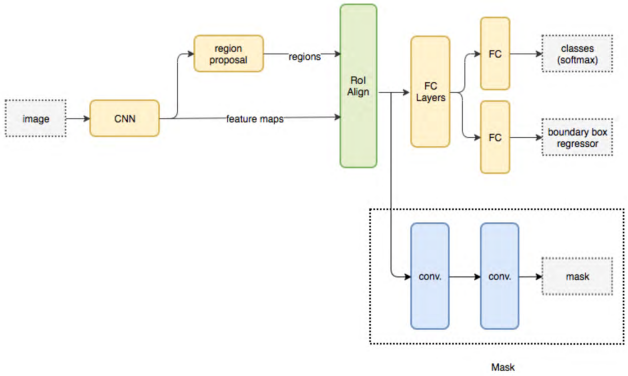


**FIGURE 5.** Architecture of the original Mask R-CNN framework. The CNN represents the backbone used for the feature extraction. Three models were tested: ResNet101, ResNet50 and MobileNetV1.

which is shown in Figure 5. The following implementation is proposed:

**Backbone:** This is a standard CNN that serves as a feature extractor. Three backbones were implemented: ResNet101, ResNet50 and MobilNetV1. The early layers of these CNN detect low-level features (edges and corners), and later layers successively detect higher level features (berries, leaves). Passing through the backbone network, the image is converted from $1,920 \times 1,080 \times 3$ (RGB) to a feature map of shape $32 \times 32 \times 2,048$. This feature map becomes the input for the following stages. Several experiments were performed for each backbone in order to select the best hyper-parameters that maximise detection performance (See Section IV).

**Feature Pyramid Network:** The FPN improves the standard feature extraction pyramid by adding a second pyramid that takes the high level features from the first pyramid and passes them down to lower layers. By doing so, it allows features at every level to have access to both lower and higher level features. This part of the architecture allows the detection of small blueberries in the image.

**Region Proposal Network:** The RPN is a lightweight neural network that scans the image in a sliding-window

fashion and finds areas that contain objects (berries). The regions that the RPN scans over are called anchors, which are boxes distributed over the image area. In practice, there are about 100,000 anchors of different sizes and aspect ratios that overlap to cover as much of the image as possible. The sliding window is handled by the convolutional nature of the RPN which allows all regions in a GPU-Card to be scanned. The RPN does not scan over the image directly. Instead, it scans over the backbone feature map. This allows the extracted features to be reused efficiently and avoids duplicate calculations.

**ROI Classifier & Bounding Box Regressor:** This stage runs on the ROIs proposed by the RPN. It generates two outputs for each ROI: **Class:** The class of the object in the ROI. Unlike the RPN, which has two classes, this network is deeper and has the capacity to classify regions to specific classes (i.e blueberry, leaf). It can also generate a background class, which causes the ROI to be discarded.

**Bounding Box Refinement:** Its purpose is to refine the location and size of the bounding box that encapsulates the object.

**ROI Align:** It crops a part of a feature map and resizes it (to a fixed size). It is similar in principle to cropping part of an image and then resizing it.

**Segmentation Masks:** The mask branch is a convolutional network that takes the positive regions selected by the ROI classifier and generates masks for them. A small mask size helps keep the mask branch light. During training, the ground-truth masks are scaled down to different sizes to compute the loss. During the inference step, the predicted masks are scaled up to the size of the ROI bounding box, which gives the final masks (one per object).

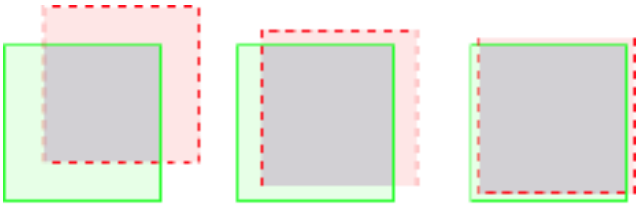

**FIGURE 6.** Visual example of Intersection Over Union quality. The dashed red-box is the predicted detection, the continuous green box is the ground truth, and the grey area is the overlap between the two. The example shows three different IOU scores from left to right, with the rightmost being the best.

### C. EVALUATION METRICS

To evaluate the proposed algorithm the Intersection over Union (IoU) measure is used [39]. The IoU measures the overlap between two boundaries. This is used to measure how much the boundary predicted by the algorithm overlaps with the ground truth (the real object boundary). See Figure 6. Traditionally, state-of-the-art datasets, an IoU threshold equal or greater than 0.5 is used in order to classify whether the prediction is a true positive or a false positive.

$$IoU = \frac{Area\ Overlap}{Area\ Union} \tag{1}$$

The mean Average Precision (mAP) [39] was used to evaluate detection performance. The mAP metric is the product of precision and recall of the detected bounding boxes. The mAP value ranges from 0 to 1. The higher the number, the better it is. The mAP can be computed by calculating average precision (AP) separately for each class (blueberry and background) and then averaging over the classes. A detection is considered a true positive only if the IoU is above a certain threshold. In this work, two threshold values were evaluated 0.5 and 0.7 in according to the state of the art [11].

All blueberry detections from test images may be combined in a precision/recall curve. The final area under the curve was used to compare the algorithms. $N$ represents the number of images. The mAP can be computed as follows:

$$mAP = 1/N \sum_{i=1}^{N} AP_i \tag{2}$$

## IV. EXPERIMENTS AND RESULTS

In order to design the best Mask R-CNN algorithm for the segmentation of different instances of blueberry, a three-stage set of experiments were performed. The first stage aims at detecting blueberries, the second stage deals with instance segmentation, and the third stage was designed to test several backbones for mobile implementation. All experiments were performed using images captured during the pre-harvest period in wild conditions, and trained using an NVIDIA Quadro P6000 GPU with 24GB of VRAM, an Intel Core i7-7740X CPU, and 64GB of RAM.

### A. BLUEBERRY DETECTION

The detection stage was performed using a ResNet101 backbone. Several test were performed using different training parameters such as:

- Number of training and validation images.
- Image size (resolution).
- Number of blueberries contained in images (Annotated blueberries).
- The use of data augmentation (horizontal-vertical flips).
- Number of epochs.
- Mini-mask shape.
- RPN anchor scales.
- Learning rate

A total of twelve tests (from Test 1 up to Test 12) for blueberry detection were carried out. The parameters used and the results obtained are reported in Table 2. The mIoU and mAP metrics are reported for each test in the last three rows of the table. Two values for the IoU threshold were used (0.5 and 0.7). A value greater or equal to the threshold was considered a true positive, whereas a score lower to the threshold was considered a false positive. The threshold equal or greater to 0.5 corresponds to the most used value in the state of the art. In order to be strict while evaluating performance a mAP threshold of 0.7 is also reported.

**TABLE 2.** Summary of the parameters used in the blueberry detection experiments (from Test 1 to Test 12). Results of each test are presented in the last three rows of each table. The best results are in italics.

| *Obj. detection* | **Test 1** | **Test 2** | **Test 3** | **Test 4** | **Test 5** | **Test 6** |
|---|---|---|---|---|---|---|
| **# Train Obj.** | 1,302 | 1,302 | 2,308 | 3,505 | 3,505 | 3,505 |
| **# Validation Obj.** | 326 | 326 | 577 | 876 | 876 | 876 |
| **# Train Img.** | 125 | 125 | 166 | 237 | 237 | 237 |
| **# Validation Img.** | 38 | 38 | 44 | 55 | 55 | 55 |
| **# Epochs** | 30 | 100 | 100 | 300 | 300 | 100 |
| **Img. Width** | 1,024 | 1,024 | 1,024 | 1,024 | 1,920 | 1,024 |
| **Img. Height** | 800 | 800 | 800 | 800 | 1,080 | 800 |
| **Mini-mask Shape** | 56×56 | 56×56 | 56×56 | 56×56 | 56×56 | 56×56 |
| **RPN Anchor Scales** | (32, 64, 128, 256, 512) | (32, 64, 128, 256, 512) | (32, 64, 128, 256, 512) | (32, 64, 128, 256, 512) | (32, 64, 128, 256, 512) | (32, 64, 128, 256, 512) |
| **Data Augmentation** | X | X | X | X | X | ✓ |
| **mIoU** | **0.587** | **0.559** | **0.495** | **0.468** | **0.394** | **0.542** |
| **mAP(IoU≥0.5)** | **0.647** | **0.745** | **0.490** | **0.584** | **0.185** | **0.499** |
| **mAP(IoU≥0.7)** | **0.561** | **0.696** | **0.466** | **0.553** | **0.179** | **0.498** |
| | **Test 7** | **Test 8** | **Test 9** | **Test 10** | **Test 11** | ***Test 12*** |
| **# Train Obj.** | 3505 | 3576 | 6117 | 6117 | 7009 | *7009* |
| **# Validation Obj.** | 876 | 874 | 1529 | 1529 | 3152 | *3152* |
| **# Train Img.** | 237 | 238 | 251 | 251 | 177 | *177* |
| **# Validation Img.** | 55 | 54 | 51 | 51 | 128 | *128* |
| **# Epochs** | 100 | 500 | 100 | 30 | 100 | *500* |
| **Img. Width** | 1,024 | 1,024 | 1,024 | 1,920 | 1,920 | *1,920* |
| **Img. Height** | 800 | 800 | 800 | 1,080 | 1,080 | *1,080* |
| **Mini-mask Shape** | 28×28 | 28×28 | 28×28 | 56×56 | 28×28 | *28×28* |
| **RPN Anchor Scales** | (32, 64, 128, 256, 512) | (32, 64, 128, 256, 512) | (32, 64, 128, 256, 512) | (16, 32, 64, 128, 256) | (8, 16, 32, 64, 128) | *(8, 16, 32, 64, 128)* |
| **Data Augmentation** | ✓ | X | X | X | X | ✓ |
| **mIoU** | **0.472** | **0.452** | **0.422** | **0.449** | **0.589** | ***0.575*** |
| **mAP(IoU≥0.5)** | **0.569** | **0.586** | **0.475** | **0.529** | **0.710** | ***0.739*** |
| **mAP(IoU≥0.7)** | **0.542** | **0.578** | **0.452** | **0.508** | **0.685** | ***0.706*** |

*Test 1* and *Test 2* used similar parameters but with a different number of epochs (30 and 100 respectively). The increase in epochs improved the detection results from mAP(IoU≥0.5) of 0.647 in Test 1 to 0.745 in Test 2. This shows the convergence of the algorithm. In *Test 3* and *Test 6* a bigger number of images and annotated objects for training and testing were used but considering a similar number of epochs (100) than previous tests. Results shown a decrease in the detection rate. This was later improved in *Test 4* by increasing the number of epochs reaching an mAP(IoU≥0.5) of 0.584.

The influence of the image width and height was evaluated in *Test 5* were images were re-sized to $1,920 \times 1,080$ instead of $1,024 \times 800$ as previous tests. The learning rate was increased from 0.001 (default value in previous experiments) to 0.02. In this case a poor performance of the algorithm was obtained (mAP(IoU≥0.5) = 0.185).

In *Test 7*, *Test 8* and *Test 9* the reduction of the mini-mask shape from $56 \times 56$ to $28 \times 28$ was explored. These tests were performed using different number of images, number of annotated objects and epochs. The best result was obtained in *Test 8*, where 500 epochs were used with 3,576 annotated objects for training and 874 for validation. The annotated objects included smaller sizes of blueberry which results in an improvement in detection (mAP(IoU≥0.5) = 0.586). *Test 10* uses a similar configuration to the previous one, but with an image resolution of $(1,920 \times 1,080)$, and a scale reduction of the RPN anchors. These two parameters shown to be the most effective for the detection of smaller fruit.

The distribution between training and testing images was 80/20 for all tests except for *Test 11* and *Test 12*. In these tests, a bigger number of annotated object were used with a proportion of 70/30 for training and validation. In *Test 11* RPN Anchor scales are further reduced. Similar parameters were used in *Test 12* but also data augmentation is used in this case. Figure 7 shows a visual example of the results where the number of detected blueberries is also reported. Additionally, Figure 8 shows the precision-recall curves for all tests using IoU thresholds at 0.5 and 0.7.

### B. INSTANCE SEGMENTATION

For the instance segmentation stage, two experiments were performed (Test 1 and Test 2). The parameters tested for both experiments and results obtained are reported in Table 3. The mIoU and mAP metrics for each experiment are shown at the last three rows of the table. Thresholds of 0.5 and 0.7 were used for computing the mAP value.

Both tests used a 80/20 distribution between training and validation images and a resolution of $1,024 \times 800$. In test 1 the mini-mask shape parameters were set to $56 \times 56$ while in

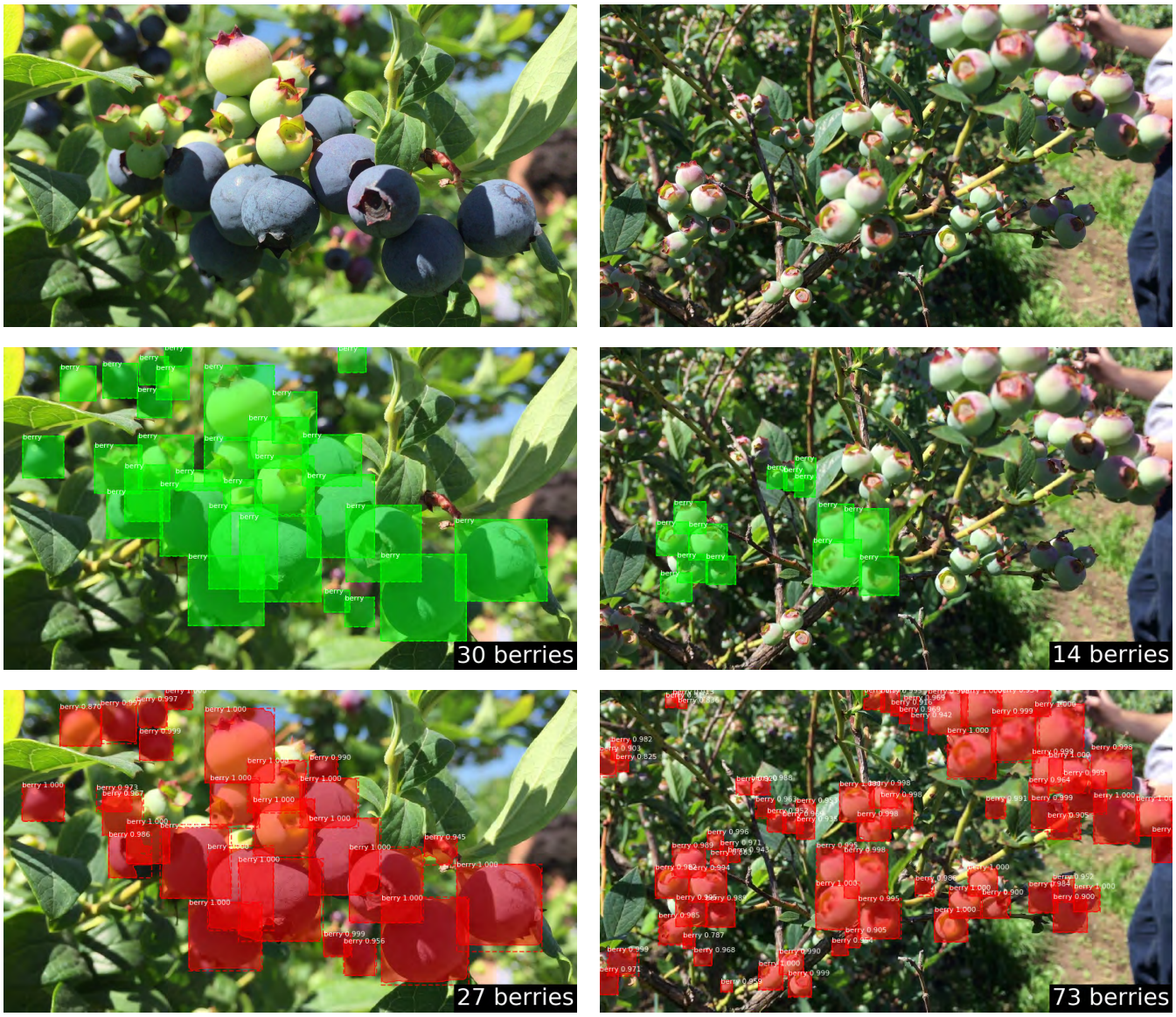


**FIGURE 7.** Examples of blueberry detection. The original image is shown at the top row, the ground truth at the middle row, and the resulting detected blueberries at the bottom. The number of automatically quantified blueberries is shown in the bottom-right corner of each image.

test 2 they were reduced to 28 × 28. The number of epochs was increased from 30 in test 1 to 100 in test 2. The best result was obtained in test 2 where an mAP(IoU≥0.5) of 0.909 was obtained.

An example of the instance blueberry segmentation is shown in Figure 9. Figure 10 shows the precision-recall curves for all tests using IoU thresholds at 0.5 and 0.7. Additional results from images taken in different angles are shown in Figure 11.

### C. ALGORITHM OPTIMISATION FOR MOBILE DEVICE IMPLEMENTATION

In this section, two backbones were tested: ResNet50 and MobileNetV1. These neural networks are designed for mobile device implementation. They are based in residual learning which simplifies their architecture making them less computationally expensive. Similar parameters used for training the ResNet101 are used for both ResNet50 and MobileNetV1 networks. The details of the images, training parameters used, and achieved results are shown in Table 4. A comparison of the computing performance and the Model Filesize of the three backbones tested in this paper (ResNet101, ResNet50 and MobileNetV1) is also reported.

**The traditional Mask R-CNN algorithm** fails when is it directly applied to blueberry detection and segmentation. This can be shown in Table 5. The traditional implementation of the algorithm only detected 77 blueberries (from a database of 128 images) while the proposed algorithm achieved a higher number of detected berries 6,766. Figure 12, shows the

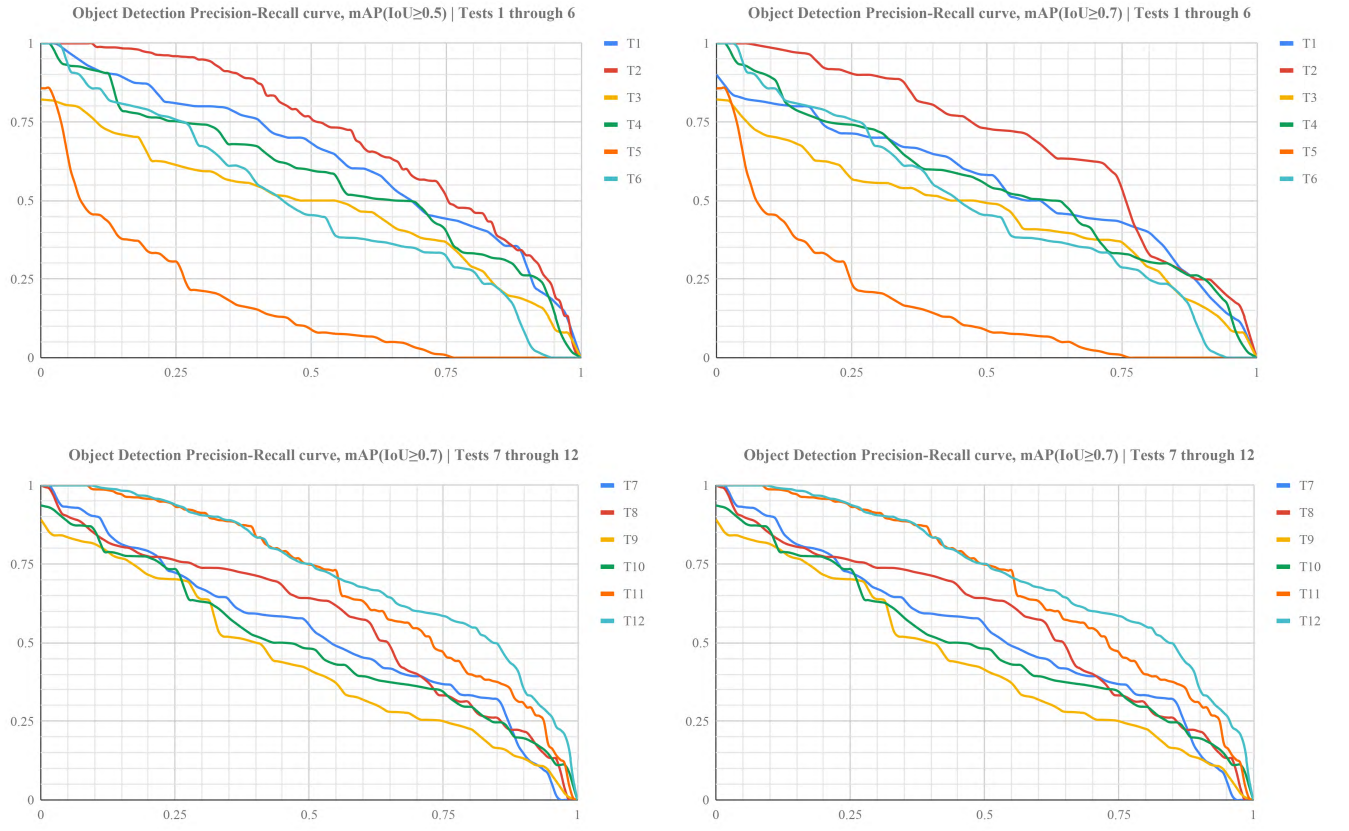


**FIGURE 8.** Precision-Recall curves for all tests. Top two images are Precision-Recall curves for tests 1 to 6 (left) and tests 7 to 12 (right) when using an IoU of 0.5. The bottom images are Precision-Recall curves for tests 1 to 6 (left) and tests 7 to 12 (right) when using an IoU of 0.7.


**FIGURE 9.** Examples of blueberry instance segmentation. The original image is shown at the left, the ground truth at the middle row, and the resulting segmented blueberries at the right. The number of automatically quantified blueberries is shown in the bottom-right corner of each image.

detection performance of both algorithms for the same input image.

## V. DISCUSSION

It is observed that the higher the amount of training data (number of images and annotated blueberry instances), the higher the image resolution, the lower the mini-mask shape and the lower the scale of the RPN anchors all positively influence performance in both tasks; the latter three specially for smaller blueberries. For object detection, analysis of the results show that 300 epochs are enough to achieve convergence. It is also observed that the use of data augmentation considerably improved the results.

The stand-alone training of the implemented algorithm demands high computing resources in order to get a trade off between the size of the model and accuracy of the results. However, once trained, the model could be easily used in mobile devices.

A limitation of this algorithm is the large number of hyper-parameters that need to be tuned. A large number of experiments are required to select the best hyper-parameters in order to achieve competitive results.

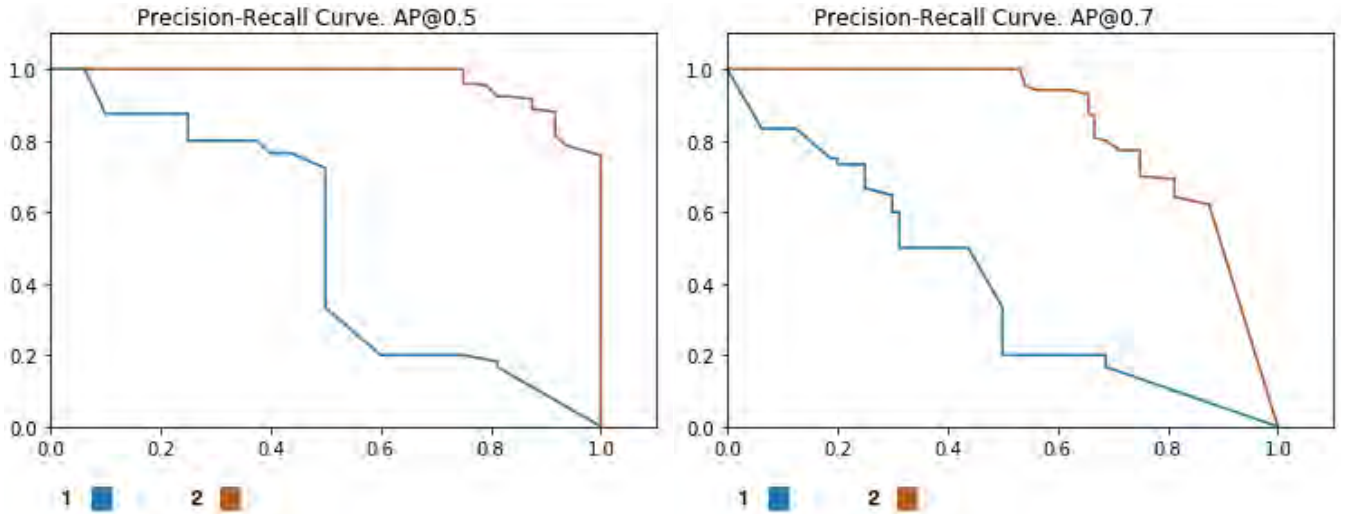


**FIGURE 10.** Precision-recall curves for test 1 and test 2 in the instance segmentation experiments. Left plot displays Precision-Recall curve computed using a IoU threshold of 0.5 and Right plot displays Precision-Recall curve computed using a IoU threshold of 0.5.


**FIGURE 11.** Additional Examples of blueberry Quantification using images acquired from different positions (Top and bottom). The ground truth images are on the left and the resulting segmented blueberries at the right. The number of automatically quantified blueberries is shown in the bottom-right corner of each image.

The algorithm is able to detect and count even small blueberries. This task is particularly difficult due to the high variability in size of the blueberries present in the images. In this work a minimum size of 10mm was detected and counted. Detection and segmentation of smaller blueberries are still an open problem.

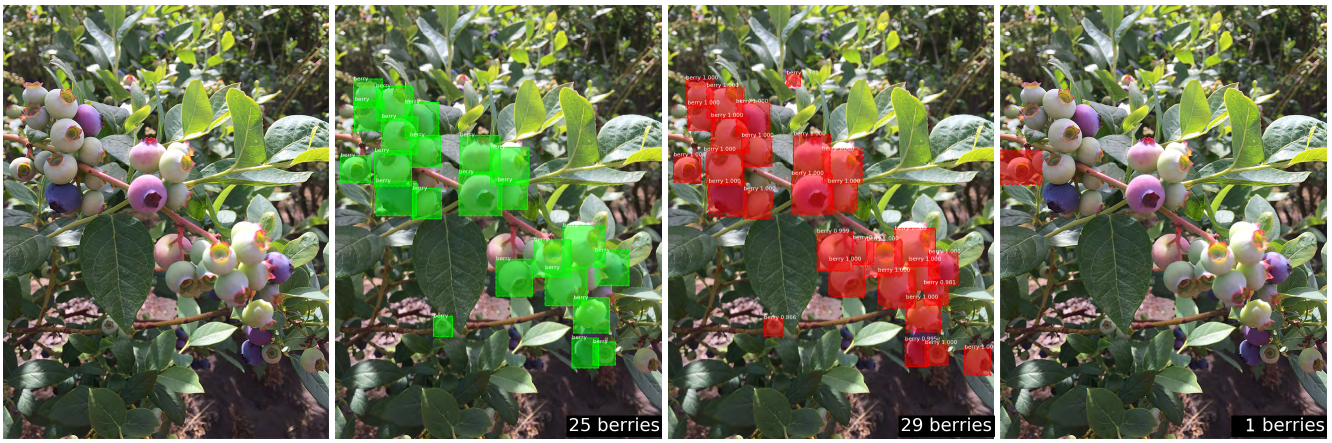


**FIGURE 12.** Comparison of our proposed method (DeepBlueBerry) and the standard Mask-cnn implementation. From left to right: Input image, ground truth, results obtained using DeepBlueBerry (29 blueberries) and results with traditional Mask R-CNN (1 blueberry).

**TABLE 3.** Summary of the parameters used during Test 1 and Test 2 of the instance segmentation algorithm. Evaluation results are reported in the final three rows of the table. The best results are in italics.

| *Ins. segmentation* | **Test 1** | ***Test 2*** |
|---|---|---|
| **# Train Obj.** | 30 | *156* |
| **# Validation Obj.** | 30 | *72* |
| **# Train Img.** | 3 | *4* |
| **# Validation Img.** | 3 | *3* |
| **# Epochs** | 30 | *100* |
| **Img. Width** | 1,024 | *1,024* |
| **Img. Height** | 800 | *800* |
| **Mini-mask Shape** | 56×56 | *28×28* |
| **RPN Anchor Scales** | (32, 64, 128, 256, 512) | *(32, 64, 128, 256, 512)* |
| **Data Augmentation** | X | X |
| **mIoU** | **0.484** | ***0.726*** |
| **mAP(IoU≥0.5)** | **0.498** | ***0.909*** |
| **mAP(IoU≥0.7)** | **0.498** | ***0.774*** |

**TABLE 4.** Results for object detection using different backbones models. Computing time is shown in seconds. Evaluation results are reported in the final three rows where the best results are in italics.

| *Obj. Det. backbones* | **ResNet101** | **ResNet50** | **MobileNet** |
|---|---|---|---|
| **# Validation Img.** | 128 | 128 | 128 |
| **# Detected Berries** | 6,561 | 6,766 | 5,282 |
| **# Detections per Image** | 25.629 | 26.429 | 20.633 |
| **Total Compute Time** | 368.635 s | 327.153 s | 211.227 s |
| **Compute Time per Image** | 1.439 s | 1.278 s | 0.825 s |
| **Compute Time per Berry** | 0.056 s | 0.048 s | 0.039 s |
| **Model Filesize** | 255.9 MB | 179.2 MB | 95.6 MB |
| **mIoU** | **0.575** | **0.595** | **0.567** |
| **mAP(IoU≥0.5)** | **0.739** | **0.759** | **0.693** |
| **mAP(IoU≥0.7)** | **0.706** | **0.724** | **0.591** |

Previous work using similar architectures has focused on a limited variety of fruits which sizes are usually similar to the images which facilitate the detection process (ie. apples, mangoes, etc). Blueberries have not been considered in this research mainly due to the lack of available annotated images. This work addressed the task of manually annotating images in order to provide a novel database for researchers.

**TABLE 5.** Comparison of blueberry detection when using the proposed method (DeepBlueBerry) and the traditional Mask R-CNN algorithm.

| *Comparison* | **DeepBlueBerry** | **Mask-RCNN** |
|---|---|---|
| **# Validation Img.** | 128 | 128 |
| **# Detected Berries** | *6,766* | 77 |
| **# Detections per Image** | *53* | 1 |
| **mIoU** | ***0.595*** | **0.155** |
| **mAP(IoU≥0.5)** | ***0.759*** | **0.008** |
| **mAP(IoU≥0.7)** | ***0.724*** | **0.008** |

An invitation to all the research community is made to continue labelling images of blueberries in order to have a larger publicly available database for research. A bigger database will improve the mAP/IoU measures making the system more accurate and reliable.

## VI. CONCLUSION

An instance segmentation method based on mask R-CNN was proposed for detecting, segmenting and quantifying blueberries in the wild. A novel annotated database for detection and instance segmentation of blueberries is also reported. The images were captured from a blueberry farm using a mobile device and then manually annotated to build the ground truth.

The mask R-CNN algorithm was implemented using three different backbones: ResNet101, ResNet50 and MobileNetV1. Several experiments were performed in order to select the best hyper-parameters of each implemented model.

The performance of the algorithm was evaluated using mIoU and mAP metrics. The best result was obtained when the ResNet50 backbone was used achieving a mIoU score of 0.595 and mAP scores (for IoU thresholds 0.5 and 0.7) of 0.759 and 0.724 respectively.

The ResNet 50 has a smaller number of layers which helps avoid overfitting given the small number of images in the database. A deeper implementation, requires in general, a bigger number of images on the training database to avoid

these problems. When a deeper network (ResNet101) starts converging the accuracy gets saturated and degrades. The model filesize achieved using Resnet50 is only 179.2MB compared with 255.9MB of the ResNet101. This filesize motivates us to explore the use of lighter architectures such as MobileNet.

The MobileNet backbone sacrifices performance while offering a sizeable decrease in compute time per image, going from 1.439 seconds to 0.825 seconds. For instance segmentation, the best scores was 0.726 for the mIoU metric. The mAP metric achieved were 0.909 and 0.774 for thresholds of 0.5 and 0.7 respectively.

## ACKNOWLEDGEMENT

Further thanks to Berries BIO-BIO, Luis Zapata and Pablo Sanchez from Los Angeles.

## REFERENCES

[1] M. Al-Qizwini, I. Barjasteh, H. Al-Qassab, and H. Radha "Deep learning algorithm for autonomous driving using GoogLeNet," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2017, pp. 89–96.

[2] M. Anthimopoulos, S. Christodoulidis, L. Ebner, T. Geiser, A. Christe, and S. Mougiakakou, "Semantic segmentation of pathological lung tissue with dilated fully convolutional networks," *IEEE J. Biomed. Health Inform.*, vol. 23, no. 2, pp. 714–722, Mar. 2019.

[3] S. Bargoti and J. Underwood, "Deep fruit detection in orchards," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Jun. 2017, pp. 3626–3633.

[4] W. Castro, J. Oblitas, M. De-La-Torre, C. Cotrina, K. Bazán, and H. Avila-George, "Classification of cape gooseberry fruit according to its level of ripeness using machine learning techniques and different color spaces," *IEEE Access*, vol. 7, pp. 27389–27400, 2019.

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[6] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3150–3158.

[7] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 379–387.

[8] J. Deng, W. Dong, R. Socher, K. Li, L. Fei-Fei, and L.-J. Li, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[9] A. Dutta, A. Gupta, and A. Zisserman. (2016). *VGG Image Annotator (VIA)*. [Online]. Available: http://www.robots.ox.ac.uk/ vgg/software/via/

[10] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez, "A review on deep learning techniques applied to semantic segmentation," 2017, *arXiv:1704.06857*. [Online]. Available: https://arxiv.org/abs/1704.06857

[11] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 580–587.

[13] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Simultaneous detection and segmentation," in *Computer Vision-ECCV*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham, Switzerland: Springer, 2014, pp. 297–312.

[14] B. Hariharan, P. Arbelaez, R. Girshick, and J. Malik, "Hypercolumns for object segmentation and fine-grained localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 447–456.

[15] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2961–2969.

[16] K. He, X. Zhang, S. Ren, and J. Sun "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[17] J. Hosang, R. Benenson, P. Dollár, and B. Schiele, "What makes for effective detection proposals?" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 4, pp. 814–830, Apr. 2016.

[18] M. S. Hossain, M. Al-Hammadi, and G. Muhammad, "Automatic fruit classification using deep learning for industrial applications," *IEEE Trans. Ind. Informat.*, vol. 15, no. 2, pp. 1027–1034, Feb. 2019.

[19] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*. [Online]. Available: https://arxiv.org/abs/1704.04861

[20] J. M. Ponce, A. Aquino, B. Millan, and J. M. Andújar, "Automatic counting and individual size and mass estimation of olive-fruits through computer vision techniques," *IEEE Access*, vol. 7, pp. 59451–59465, 2019.

[21] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Comput. Electron. Agricult.*, vol. 147, pp. 70–90, Apr. 2018.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2012, pp. 1097–1105.

[23] H.-L. Kuang, L. L. H. Chan, and H. Yan, "Multi-class fruit detection based on multiple color channels," in *Proc. Int. Conf. Wavelet Anal. Pattern Recognit. (ICWAPR)*, vol. 7, Jul. 2015, pp. 1–7.

[24] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[25] H. Li, W. S. Lee, and K. Wang, "Identifying blueberry fruit of different growth stages using natural outdoor color images," *Comput. Electron. Agricult.*, vol. 106, pp. 91–101, Aug. 2014.

[26] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, "Fully convolutional instance-aware semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2359–2367.

[27] X. Liu, S. W. Chen, S. Aditya, N. Sivakumar, S. Dcunha, C. Qu, C. J. Taylor, J. Das, and V. Kumar, "Robust fruit counting: Combining deep learning, tracking, and structure from motion," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1045–1052.

[28] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[29] W. D. N. Pacheco and F. R. J. López, "Tomato classification according to organoleptic maturity (coloration) using machine learning algorithms K-NN, MLP, and K-means clustering," in *Proc. 22nd Symp. Image, Signal Process. Artif. Vis. (STSIVA)*, Apr. 2019, pp. 1–5.

[30] P. O. Pinheiro, R. Collobert, and P. Dollár, "Learning to segment object candidates," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1990–1998.

[31] P. O. Pinheiro, R. Collobert, and P. A. R. Doll, "Learning to segment object candidates," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, vol. 2. Cambridge, MA, USA: MIT Press, 2015, pp. 1990–1998.

[32] K. P. Seng, L.-M. Ang, L. M. Schmidtke, and S. Y. Rogiers, "Computer vision and machine learning for viticulture technology," *IEEE Access*, vol. 6, pp. 67494–67510, 2018.

[33] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556

[34] J. E. Tapia and C. Arellano "Soft-biometrics encoding conditional GAN for synthesis of NIR periocular images," *Future Gener. Comput. Syst.*, vol. 97, pp. 503–511, Aug. 2019.

[35] C. Yang, W. S. Lee, and P. Gader "Hyperspectral band selection for detecting different blueberry fruit maturity stages," *Comput. Electron. Agricult.*, vol. 109, pp. 23–31, Nov. 2014.

[36] C. Yang, W. S. Lee, P. Gader, and H. Li, "Hyperspectral band selection using Kullback-Leibler divergence for blueberry fruit detection," in *Proc. 5th Workshop Hyperspectral Image Signal Process., Evol. Remote Sens. (WHISPERS)*, Jun. 2013, pp. 1–4.

[37] Y. Yu, K. Zhang, L. Yang, and D. Zhang, "Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN," *Comput. Electron. Agricult.*, vol. 163, Aug. 2019, Art. no. 104846.

[38] M. D. Zeiler and R. Fergus, *Visualizing and Understanding Convolutional Networks* (Lecture Notes in Computer Science), 2014, pp. 818–833.

[39] E. Zhang and Y. Zhang, *Average Precision*. Boston, MA, USA: Springer, 2009, pp. 192–193.

**SEBASTIAN GONZALEZ** received the B.S. degree in computer engineering from Universidad Andres Bello, in 2019. His current research interests include computer vision, pattern recognition, and machine learning applied to real agricultural problems, such as fruit detection, classification, and segmentation.

**CLAUDIA ARELLANO** received the degree in civil, electrical, and industrial and the M.Sc. degree from the Pontificia Universidad Catolica de Chile, in 2003 and 2005, respectively, and the Ph.D. degree from Trinity College Dublin, Ireland, in 2014. She is currently the Research and Development Director of informatic area with the Universidad Tecnologica de Chile- INACAP. Her research interests include patterns recognition, shape detection, and robust statistics.

**JUAN E. TAPIA** received the P.E. degree in electronics engineering from Universidad Mayor, in 2004, the M.S. degree in electrical engineering from the Universidad de Chile, in 2012, and the Ph.D. degree from the Department of Electrical Engineering, Universidad de Chile, in 2016. In addition, he spent one year of internship at the University of Notre Dame. From 2016 to 2017, he was an Assistant Professor with Universidad Andres Bello. He is currently the Research and Development Director of electricity and electronics with the Universidad Tecnologica de Chile—INACAP. His current research interests include pattern recognition and machine learning applied to soft biometrics, gender classification, feature fusion, and feature selection.