agriculture

MDPI

Article

Cherry Tree Crown Extraction Using Machine Learning Based on Images from UAVs

Vasileios Moysiadis [1], Ilias Siniosoglou [1,2], Georgios Kokkonis [3], Vasileios Argyriou [4], Thomas Lagkas [5], Sotirios K. Goudos [6] and Panagiotis Sarigiannidis [1,2,*]

[1] Department of Electrical and Computer Engineering, University of Western Macedonia, 50100 Kozani, Greece; bmoisiadis@uowm.gr (V.M.); isiniosoglou@metamind.gr or isiniosoglou@uowm.gr (I.S.)
[2] R&D Department, MetaMind Innovations P.C., 50100 Kozani, Greece
[3] Department of Information and Electronic Systems Engineering, International Hellenic University, 57400 Thessaloniki, Greece; gkokkonis@ihu.gr
[4] Department of Networks and Digital Media, Kingston University, Kingston upon Thames KT1 2EE, UK; vasileios.argyriou@kingston.ac.uk
[5] Department of Computer Science, International Hellenic University, 65404 Kavala, Greece; tlagkas@cs.ihu.gr
[6] ELEDIA@AUTH, Physics Department, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece; sgoudo@physics.auth.gr
* Correspondence: psarigiannidis@metamind.gr or psarigiannidis@uowm.gr

Abstract: Remote sensing stands out as one of the most widely used operations in the field. In this research area, UAVs offer full coverage of large cultivation areas in a few minutes and provide orthomosaic images with valuable information based on multispectral cameras. Especially for orchards, it is helpful to isolate each tree and then calculate the preferred vegetation indices separately. Thus, tree detection and crown extraction is another important research area in the domain of Smart Farming. In this paper, we propose an innovative tree detection method based on machine learning, designed to isolate each individual tree in an orchard. First, we evaluate the effectiveness of Detectron2 and YOLOv8 object detection algorithms in identifying individual trees and generating corresponding masks. Both algorithms yield satisfactory results in cherry tree detection, with the best F1-Score up to 94.85%. In the second stage, we apply a method based on OTSU thresholding to improve the provided masks and precisely cover the crowns of the detected trees. The proposed method achieves 85.30% on IoU while Detectron2 gives 79.83% and YOLOv8 has 75.36%. Our work uses cherry trees, but it is easy to apply to any other tree species. We believe that our approach will be a key factor in enabling health monitoring for each individual tree.

Keywords: machine learning; tree detection; remote sensing; smart farming; Detectron2; YOLOv8

check for updates

Citation: Moysiadis, V.; Siniosoglou, I.; Kokkonis, G.; Argyriou, V.; Lagkas, T.; Goudos, S.K.; Sarigiannidis, P. Cherry Tree Crown Extraction Using Machine Learning Based on Images from UAVs. *Agriculture* **2024**, *14*, 322. https://doi.org/10.3390/agriculture14020322

Academic Editors: Luís Manuel Navas Gracia and Manuel Pérez-Ruiz

Received: 1 January 2024
Revised: 13 February 2024
Accepted: 14 February 2024
Published: 18 February 2024

1. Introduction

The agriculture domain explores innovative methods to leverage production and improve the quality of the products. In this context, recent technologies play a vital role and improve many aspects of the production procedure. We are living at the beginning of a new era called Smart Farming, where Unmanned Aerial Vehicles (UAVs), Unmanned Ground Vehicles (UGVs), and Machine Learning are only a few of the advanced technologies aiming to transform the traditional cultivation methods [1]. Consequently, many advanced operations in cultivation, such as remote monitoring, disease detection, weed detection and management, aim to offer additional information channels and provide valuable support for farmers and agronomists.

Tree detection is another research area with practical applications in agriculture and forestry [2]. Tree identification, tree counting, and crown delineation are three of them. Data sources suitable for this purpose include UAVs, satellites, and Light Detection And Ranging (LiDAR) [3]. While the most common choice for images from UAVs and satellites is in the visible band, multispectral [4] and hyperspectral images are also viable options.

There are various approaches in this research area, such as image processing, techniques based on point clouds, Machine Learning, and Deep Learning [5]. Image processing techniques [6] include simple methods with low complexity, offering fast results but coming with comparatively lower accuracy. They work well in simple scenarios and with clear images but may face challenges in complex environments. Additionally, they have relatively low costs as they rely on standard image processing techniques.

The second category includes methods based on point clouds. Most approaches in this category rely on LiDAR [7,8] for data acquisition, ensuring high accuracy. However, they come with a relatively high cost, as well as moderate to high complexity and slow preparation time, but they offer fast processing time. Alternatively, other methods utilise Structure from Motion (SfM) [9,10], providing medium accuracy and moderate complexity but at a lower cost.

Machine Learning algorithms, such as Support Vector Machines (SVM) or K-Means [11], can achieve medium to high accuracy, depending on the quality and representativeness of the training data. They require moderate training time, but inference can be relatively fast. Moreover, the cost is moderate, as it is associated with data labelling and computational resources for training. On the other hand, Deep Learning constitutes a subset of Machine Learning and includes state-of-the-art algorithms for object detection and image segmentation based on Convolutional Neural Networks (CNNs) [12]. They can achieve high accuracy, but the cost is relatively high due to the necessity for large labelled datasets and significant computational resources. In addition, they have high complexity with high preparation time, but the processing time is relatively fast. Finally, a combination of the above methods is also a commonly used practice.

Table 1 summarises the main characteristics of the main categories in tree detection. The values on the characteristics of the table are indicative as they are highly dependent on various parameters such as the data acquisition method, the complexity of the dataset, as well as the selected method within the category.

Table 1. Categories of research methods for tree detection and crown extraction.

Category	Examples	Accuracy	Complexity	Preperation Time	Processing Time	Cost
Image Processing	Template Matching Vegetation Index	Low	Low	Fast	Fast	Low
Machine Learning	SVM K-Means	Moderate/High	Moderate	Moderate	Relative Fast	Moderate
Point Cloud	LiDAR SfM	High Medium	Moderate/High Moderate	Slow Moderate/High	Fast Fast	High Low
Deep Learning	CNNs	High	High	Slow	Relative Fast	High

Regarding Deep Learning, numerous algorithms for object detection exist, with varying results in speed and accuracy. Fast-RCNN, Faster-RCNN, Mask-RCNN, SSD, and YOLO are only some of them [13]. All of them are based on Convolutional Neural Networks (CNN) and promise to detect the location of different objects with high accuracy in images or videos.

In addition, remote sensing based on images from UAVs is becoming a promising technique to observe large areas with cultivation and automatically identify possible diseases. Many vegetation indices exist that are used to predict different parameters of cultivation. For example, the Normalised Difference Vegetation Index (NDVI), Normalised Difference Red Edge (NDRE), and Soil-Adjusted Vegetation Index (SAVI) are three of the most known vegetation indices. In cultivations such as wheat or barley, it is sufficient to apply the preferred index and then use a classification algorithm to divide the field into different areas based on health status. However, in the case of orchards where the trees do not cover the entire field, it is appropriate first to detect each individual tree and then check their health status based on the preferred vegetation index. For example, the authors in [14] present a method that uses the canopy shape and vegetation indices to identify citrus greening disease.

In this paper, we present a tree detection method as a key enabler for remote health monitoring based on UAV images. We evaluated the proposed method on cherry trees, and the results show efficiency on tree detection with F1 Score up to 94.85% for Detectron2 (version 2.0, Facebook) and 87.51% for YOLOv8 (version 8.0, Ultralytics). In addition, the proposed improvement in effective coverage of the tree crown gives better results than Detectron2 and YOLOv8. In particular, it reaches effectiveness up to 85.30% on IoU when the cultivation has low coverage from weeds and grass. The effectiveness of Detectron2 is 79.53%, and of YOLOv8 is 75.36%. Moreover, our method can be easily used in other types of trees when they are not densely planted in the field.

Detectron2 is an object detection framework based on Mask-RCNN. It uses Convolutional Neural Networks (CNNs) and, as a result, returns the position and the mask of the detected objects in an image. YOLOv8 is the newest version of the YOLO (You Only Look Once) family and is considered state-of-the-art in object detection.

Some research efforts are already available for tree detection, but most of them are limited only to detection while not providing the mask of the tree. In [15,16], the authors compare Mask-RCNN with other methods of tree detection, but they lack the precision of the provided mask. In contrast, our approach not only adequately provides the mask for detected trees but also enhances these masks, ensuring they accurately represent the crown of the trees. To the best of our knowledge, there are no similar works for images captured from UAVs that automatically detect trees in an orchard and try to improve the provided masks.

The outline of our work is as follows:

- Acquiring and annotating orthomosaic images of orchards with cherry trees of various species.
- Training machine learning models with Detectron2 and YOLOv8 and evaluating their effectiveness in cherry tree detection.
- Generating a mask for each cherry tree using Detectron2 and YOLOv8.
- Improving the provided masks with an additional algorithm based on the OTSU [17] thresholding method to ensure comprehensive coverage of the tree crown.
- Evaluating the effectiveness of the provided masks based on Detectron2, YOLOv8 and the proposed algorithm.

Using the generated masks, we calculate various vegetation indices such as NDVI and NDRE, enabling the evaluation of potential stress for each tree. This assessment involves comparisons between different trees or the same tree across previous seasonal periods. Furthermore, it is crucial to conduct additional investigations using ground cameras to identify any potential diseases. We aim for this research to become an easy tool for remote health monitoring of various species of trees, contributing to future endeavours in Smart Farming research.

The rest of this paper is as follows: In Section 2, we present the most relevant research works for tree detection using Machine Learning. Section 3 briefly discusses the basic concepts of image segmentation and provides the essential features of Detectron2 and YOLOv8. Section 4 analyses the methodology we follow in this research. Next, Section 5 presents the evaluation results between Detectron2, YOLOv8 and the mask improvement method with OTSU thresholding. Section 6 discusses the results of this research and explores potential applications for remote sensing. Finally, Section 7 concludes this paper.

2. Related Work

Machine Learning has strong potential in Smart Farming. Disease, weed and pest detection and classification are some of the applications aiming to improve the quality and quantity of products by providing timely notifications to agronomists and farmers about potential threats.

Tree detection is one of the research areas with various applications. For example, species recognition or simple tree counting are two of them. Furthermore, Machine Learning and Deep Learning are the most promising technologies for extracting information from multiple sources. This section provides an overview of research efforts in tree detection based on Machine Learning or similar research areas.

In [18], the authors present a computer vision-based citrus tree detection method based on Connected Components Labelling (CCL) algorithm. The proposed method was evaluated on high-resolution orthomosaic multi-spectral images from UAVs and showed an accuracy of 97% and precision of 95% in heterogeneous agriculture patches. The authors in [19] used Mask-RCNN to extract trees from images captured from UAVs. The evaluation was based on three large-scale images with various tree species, and tests at different scales showed no impact on the detection, even with 40% lower resolution. Nevertheless, this was one of the first research efforts with Mask-RCNN, and the results were not used to create masks but only for tree detection.

In [20], the authors propose a tree detection method based on Deep Learning. The proposed method uses two CNNs to detect oil palm trees and achieves an accuracy of up to 94.99% for the F1-score. The first CNN is responsible for land cover classification, while the second is responsible for palm tree classification.

In [21], the authors used Convolutional Neural Networks for automatic citrus tree detection on UAV images. Their approach consists of three steps. In the first step, a CNN was used on a sliding window to detect the centres of tree rows. In the second step, the proposed method selects the possible centres of trees in the detected rows with high probability. The final step uses a CNN to specify whether the candidate centres as citrus trees or not.

The authors in [15] provide a comparison between U-Net and Mask-RCNN on tree detection of pomegranate trees. Their experiment uses a dataset captured with a UAV and shows that Mask-RCNN achieves better performance than U-Net.

A coconut tree detection method is presented in [22], where authors use Mask-RCNN to detect and provide a mask for each individual tree. The Mask-RCNN was evaluated with both ResNet-50 and ResNet-101, and several experiments were conducted with various configuration parameters. The best configuration achieves an accuracy of over 90%.

The authors in [16] provide a comparison of tree detection between Mask-RCNN, Local Maxima (LM) algorithm and Marker-Controlled Watershed Segmentation (MCWS). The results show that Mask-RCNN can achieve better accuracy than the other methods. Moreover, Mask-RCNN was tested in RGB band, single-band and multi-band images, showing that RGB images result in better accuracy.

A tree and building detection method based on LiDAR measurements is proposed in [23]. The proposed method is separated into different stages. Firstly, trees and buildings are extracted from LiDAR measurements. In the second stage, a Support Vector Machine (SVM) is used to distinguish trees from buildings. The next stage includes mathematical operations responsible for rejecting small objects and correcting some artefacts of the shape of trees and buildings. In the final stage, a K-means algorithm separates buildings based on height.

Another research project based on LiDAR measurements is presented in [24], comparing eight tree detection algorithms. The dataset includes different types of forest trees. The experimental results show comparable detection rates but differ in extraction and omission/commission rates.

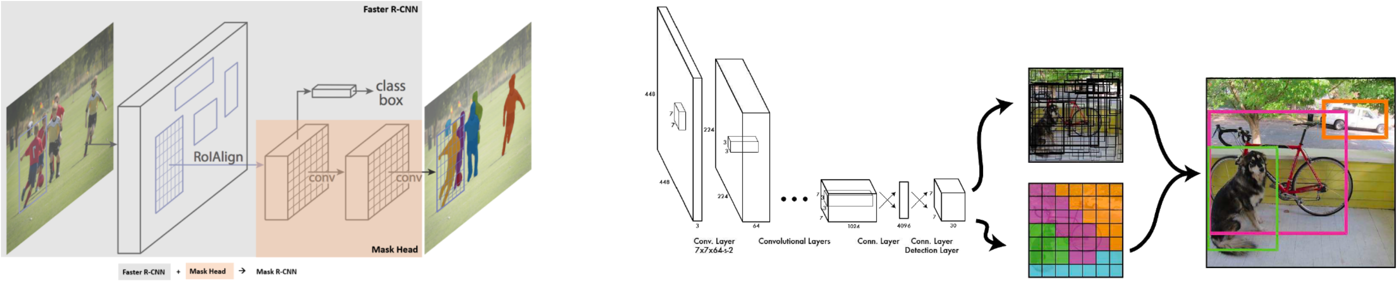
3. Background

Image classification and object detection are two tasks that are highly used in Smart Farming for various purposes, such as detecting or classifying diseases [25–27], weeds [28,29], fruits [30,31], pests [32,33] or monitoring crops [34] in the fields or greenhouses. Various machine learning and deep learning algorithms based on Convolutional Neural Networks (CNN) have been developed in recent years. Fast-RCNN, Faster-RCNN, and Mask-RCNN belong to the family of Region-based RCNN and provide the specific region of the detected object. More specifically, Fast-RCNN and Faster-RCNN provide the boundaries of a rectangle that surrounds the detected object, while Mask-RCNN [35] also provides a mask that specifies the exact pixels of the detected objects. These characteristics classify them as algorithms for instance segmentation. Detectron2 [36] is the latest and most used framework for object detection and image segmentation and uses Mask-RCNN as the basic model architecture.

Another promising object detection algorithm that achieves high accuracy in real-time is YOLO (You Only Look Once) [37] and its successors, with YOLOv8 being the newest version. To achieve high performance in real-time, it uses only a one-stage Convolutional Neural Network. A similar approach has SSD MobileNet, as it is also designed as a single-shot detector able to detect multiple objects in an image in real time.

In the following paragraphs, we will describe in brief Detectron2 [36] and YOLOv8 [38], which we are evaluating in this manuscript for cherry tree detection. Both belong in the category of instance segmentation, providing a mask of the detected object along with a boundary box.

Detectron2 is an object detection framework, based on Mask-RCNN; developed on top of Faster-RCNN and considered as state-of-the-art for instance segmentation. A simplified architecture diagram of Mask-RCNN is presented in Figure 1a. In the first stage, a Region Proposal Network (RPN) is responsible for extracting proposals as possible Regions of Interest (RoI) that contain an object of the image. As the backbone network, ResNet-50 and ResNet-101 are two of the default options for this stage. Mask-RCNN also utilises Feature Pyramid Network (FPN) along with ResNet-50 and ResNet-101, a more effective backbone network that results in better performance by reducing training time and improving accuracy.



(**a**) (**b**)

Figure 1. Simplified diagrams of architecture for (**a**) Mask-RCNN architecture [35], and (**b**) YOLO architecture.

The second stage is responsible for classifying the region proposals derived from the previous stage into specific classes. As outputs, it provides predictions of bounding boxes surrounding the detected objects. Furthermore, a parallel stage delivers the corresponding masks for the detected objects.

YOLOv8 [38] is the newest version of YOLO (You Only Look Once) [37] object detection family and is considered the state-of-the-art in object detection. YOLOv8 also has image segmentation capabilities, providing a corresponding mask of the detected object. It is a single-shot detector that performs object detection in a single pass through the neural network. In addition, it is designed to achieve real-time performance, making it suitable for various applications and adopted by many research works. Figure 1b illustrates simplified the network architecture of the initial version of YOLO. It has 24 convolutional layers followed by 2 fully connected layers. The network extracts features from the image with its initial convolutional layers and predicts output probabilities and coordinates using its fully connected layers.

YOLO divides the input image into a grid of cells, each responsible for predicting one or more bounding boxes. In the following step, YOLO calculates a score for each bounding box where higher values indicate the presence of an object. Finally, YOLO predicts class probabilities for each bounding box and decides the class of each detected object. To improve class predictions, it uses anchor boxes which have predefined shapes and sizes.

4. Materials and Methods

This section describes the methodology we follow to provide better results in tree detection and improve the provided mask. Firstly, we describe the data acquisition procedure for collecting images from UAVs from the experimental area. Secondly, we present the annotation process and the tools used to annotate the cherry trees. In the next subsection, we describe the configuration used for both Detectron2 and YOLOv8 to obtain optimal results. The final step involves presenting details for the additional algorithm designed to improve the masks created in the previous step. An overview of the methodology employed in our work is summarised in Figure 2.

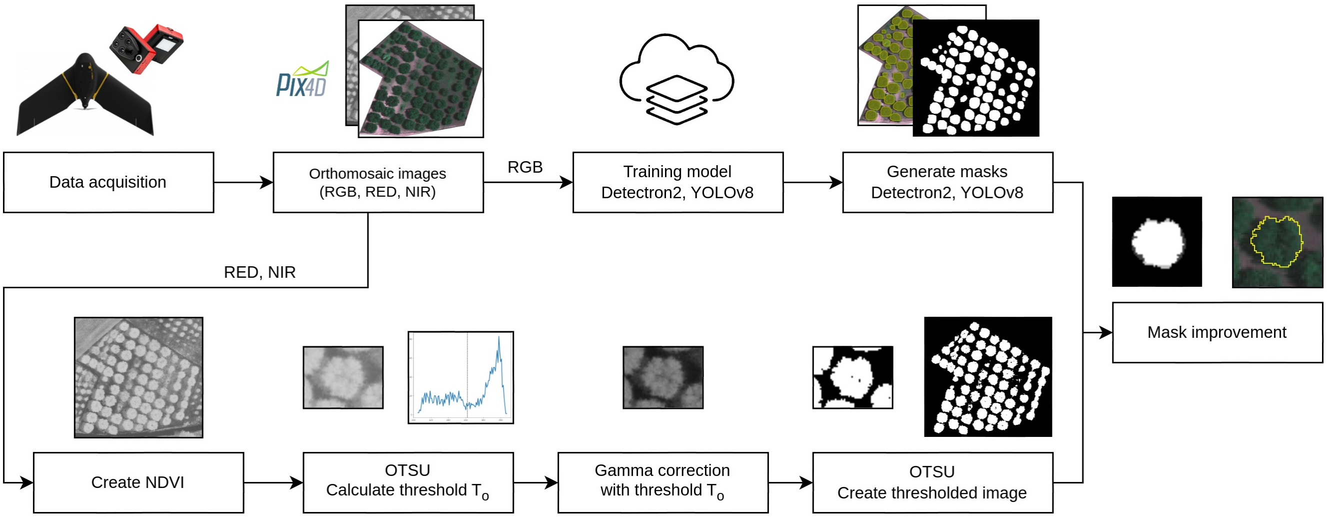

Data acquisition
PIX4D
Orthomosaic images (RGB, RED, NIR)
RGB
Training model Detectron2, YOLOv8
Generate masks Detectron2, YOLOv8
RED, NIR
Mask improvement
Create NDVI
OTSU Calculate threshold T$_o$
Gamma correction with threshold T$_o$
OTSU Create thresholded image

Figure 2. Methodology Diagram.

Table 2, presents a comparison of our work with similar research in orchards. It appears that most studies focus on tree detection rather than crown extraction. For instance, none of the works attempts to improve the provided masks. Additionally, the F1-Score in other comparable research is similar to our work, as they are based on similar algorithms like Mask-RCNN or simpler CNN-based approaches. It is important to note that a direct quantitative comparison is challenging since the metrics are highly dependent on the specific dataset used. For example, the proposed method in [18] achieves slightly better performance in F1 Score, but it uses additional data with point clouds along with high-resolution multispectral images.

Table 2. Comparison with similar research.

Research	Tree Detection	F1-Score	Tree Masks	Improve Masks
[18]	Yes	95.99%	Yes	-
[19]	Yes	-	-	-
[20]	Yes	94.99%	-	-
[21]	Yes	94.00%	-	-
[15]	Yes	-	Yes	-
[22]	Yes	-	-	-
[16]	Yes	94.68%	Yes	-
this work	Yes	94.85%	Yes	Yes

4.1. Data Acquisition

Data acquisition was acquired in two consecutive seasons in 2019 and 2020, in the Grevena Prefecture of Western Macedonia, Greece (lat: 40.1487, long: 21.3910) (Figure 3). Two flights took place during each season with a fixed-wing UAV. More specifically, the eBee X from senseFly SA (Lausanne, Switzerland) (Figure 4a) was used to acquire images from a flight altitude of approximately 120 m. The flight area was 1300 acres, where most of the cultivation was cherry trees. The eBee X was equipped with the parrot sequoia+ camera (Parrot, Paris, France) (Figure 4b) capable of capturing images in RGB bands as well as bands of RED, GREEN, REDEDGE and NIR.

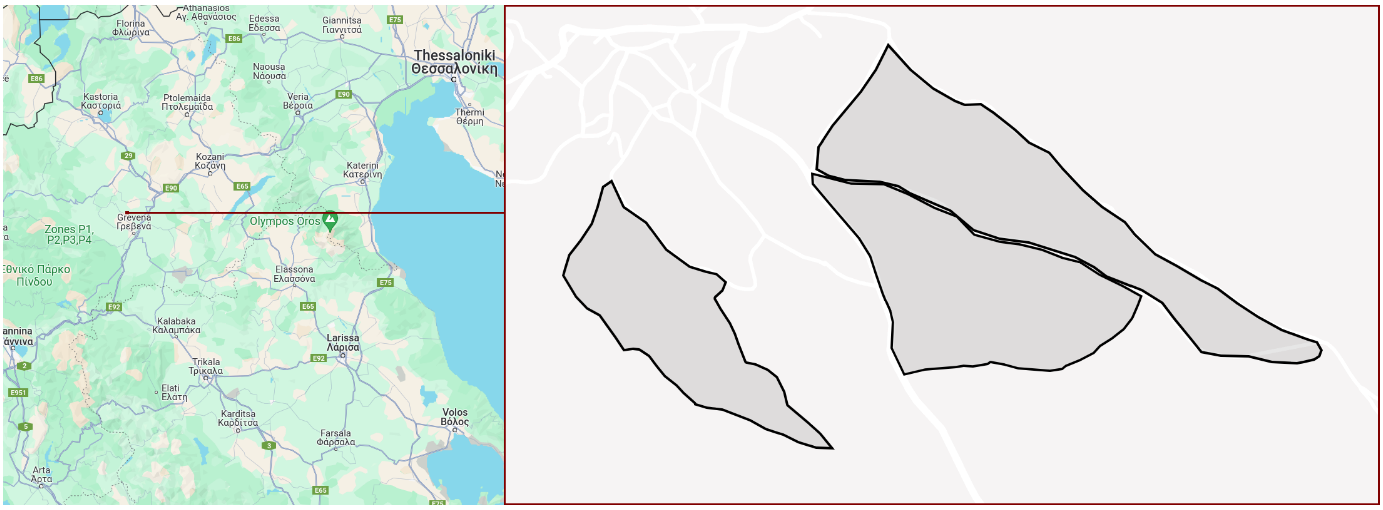


Figure 3. Location of the experimental area (Grevena Prefecture, Greece).

(**a**) eBee X (**b**) Parrot Sequoia+

Figure 4. UAV and camera used to capture photos in the experimental area.

We use Pix4Dmapper software (version 4.5.6) to create orthomosaic images for each flight and for each individual band. We use only the orchards with cherry trees for data annotation. Thus, we cropped the orthomosaic images in separate image files, where only cherry trees were present. We followed this procedure, respectively, for all image bands, but only those in RGB format were used for annotation. Figure A1 in Appendix A displays a small fraction of the dataset.

4.2. Annotation

We used the VGG Image Annotator (VIA) [39] for the annotation process, which is one of the most common image annotation tools. However, since the annotation process is time-consuming, we decided to reduce the manual annotation by following the next steps.

First, we annotated 3751 trees, using 2303 of them for training the model and 1448 for evaluating the machine learning procedure. Each tree was annotated four times for each cultivation year unless it did not appear in the orthomosaic image since it was too small or had no leaves on the flight day. Thus, the annotation dataset consists of two different classes, cherry trees and the background.

The trained model at this step was used to detect cherry trees of other orchards. We used the detected masks from this step and converted them to new annotated cherry trees. To achieve this, we used the following procedure. First, we detected all pixels on the edge of the mask and created a polygon. The polygon was simplified to reduce complexity by removing vertices and then converted to JSON format suitable for VGG Image Annotator.

In the next step, a second stage of manual annotation took place with the VIA tool. We manually corrected any faulty tree detections in this step and added those trees that the algorithm did not detect. Finally, we obtained 11,254 cherry trees annotated, 6440 for the training dataset and 4814 for the evaluation dataset. Thus, 57.22% of cherry trees were used for training and 42.78% were used for evaluation. In addition, we converted the final annotations to the corresponding format for YOLOv8. All information about annotation is summarised in Table 3.

Table 3. Dataset information.

	Orchards	Orthomosaic Images	Cherry Tree Annotations	Cherry Tree Annotations (%)
Train Dataset	32	128	6440	57.22%
Evaluation Dataset	20	80	4814	42.78%
Total	52	208	11,254	

4.3. Configuration

For the training process, we used a Debian Linux (version 11) virtual machineon the cloud with 2 vCPU cores, 8 GB of RAM and a Tesla T4 NVIDIA GPU. We evaluated both Detectron2 and YOLOv8 for instance segmentation in order to detect cherry trees in orchards. As the backbone network for Detectron2, we tested both ResNet-50 and ResNet-101. For YOLOv8 we tested YOLOv8m-seg and YOLOv8x-seg as pre-trained models. For YOLOv8, we trained both configurations for 500 epochs, and for Detectron2, we used 16,000 iterations, which are also equal to 500 epochs compared to the number of images included in the training dataset. In addition, we used a batch size of 4 for all configurations.

Furthermore, since all cherry trees have almost round crowns or are slightly elliptical, we selected a small difference in the length of the edges of the orthogonal for the anchor boxes. Thus, we have selected the values (0.8, 1.0, 1.2) as aspect ratios for Detectron2. Figure 5 illustrates the possible anchor boxes with these values for RoI extraction.

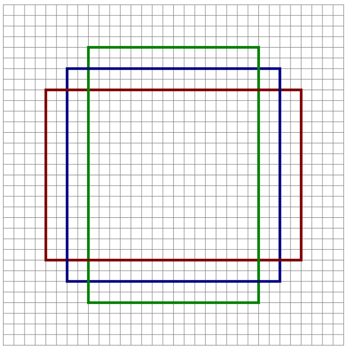


Figure 5. Aspect ratios of anchor boxes.

YOLOv8 has an anchor-free mechanism, meaning it predicts directly the centre of an object instead of the offset from a known anchor box.

The configuration mentioned above and some of the main hyper-parameters used for Detectron2 and YOLOv8 are summarised in Tables 4 and 5, respectively.

Table 4. Configuration for Detectron2.

Name	Value
pre-trained model	ResNet-50 & ResNet-101
batch size	4
iterations	16,000
lr	0.005
lrf	0.01
momentum	0.9
aspect ratios of anchor boxes	0.8, 1.0, 1.2
warm up factor	0.001
weight decay	0.0001

Table 5. Configuration for YOLOv8.

Name	Value
pre-trained model	YOLOv8m-seg & YOLOv8x-seg
batch size	4
epochs	500
lr	0.01
lrf	0.01
momentum	0.937
warm up epochs	3
weight decay	0.0005

4.4. Mask Improvement

While most of the masks generated by Detectron2 and YOLOv8 closely approximate the actual crowns of the trees, minor corrections are required to achieve optimal results. Furthermore, in some cases, two of the provided masks overlap for both Detectron2 and YOLOv8. When we want to calculate and analyse vegetation indices based on the tree crowns, it is preferable to ensure the independence of each tree.

For the above reasons, we employed an additional method to improve the provided masks. More specifically, we utilised the NDVI index of the orchard and applied a threshold based on the OTSU [17] method, along with gamma correction, to precisely outline the crown of the trees.

OTSU efficiently divides a given image into two areas of interest by determining the optimal threshold based on the image's histogram. It achieves this by iteratively searching for a threshold that minimises the intra-class variance, defined as the sum of the two variances of the two weighted classes, as shown in Equation (1).

$$\sigma_w^2(t) = \omega_0(t)\sigma_0^2(t) + \omega_1(t)\sigma_1^2(t) \tag{1}$$

The weights w_0 and w_1 represent the probabilities of the two classes separated by a threshold t, while σ_0^2 and σ_1^2 denote the variances of the two classes based on the image's histogram.

Assuming the image's histogram has L bins, the class probabilities w_0 and w_1 are calculated using Equations (2) and (3), respectively.

$$\omega_0(t) = \sum_{i=0}^{t-1} p(i) \tag{2}$$

$$\omega_1(t) = \sum_{i=t}^{L-1} p(i) \tag{3}$$

The OTSU method for threshold estimation is implemented in various graphics programming libraries, such as OpenCV.

Furthermore, before applying OTSU thresholding, we use gamma correction to improve the effectiveness of the algorithm, particularly in areas where the presence of weeds and grass is obvious.

We selected the NDVI since it is defined as the index that identifies the presence of photosynthesis. Thus, values over a threshold indicate the presence of the cherry trees in the orchard, while lower values indicate the soil surface or the surface with low coverage by grass and weeds. The image of the NDVI index should be perfectly aligned with the corresponding orthomosaic image of the orchard. Moreover, for optimal results, it is preferable to have low coverage of weeds in the orchard and to have all trees detected, especially those in close proximity to others.

The proposed method is divided into two stages. In the first stage, the algorithm is as follows: Firstly, we calculate the corresponding NDVI index for the cultivation area. Subsequently, based on this index, we create a grayscale pseudocolour image and use it as input for the next steps.

Secondly, for each detected cherry tree, we use the OTSU method to calculate a threshold on the surrounding area, which is enlarged by 100% of the mask provided by Detectron2 or YOLOv8. Before applying the threshold to the area, we apply a gamma correction based on Equations (4) and (5), where T_o is the threshold provided by the OTSU method in this step, V_{in} are the pixels of the input image, V_{out} are the pixels of the output image and 255 is the maximum value of a pixel. Gamma correction helps to make cherry trees more distinct from grass and weeds.

$$V_{out} = \left(\frac{V_{in}}{255}\right)^{gamma} \times 255 \tag{4}$$

$$gamma = \frac{255}{255 - T_o} \tag{5}$$

In the next step, after applying gamma correction, we recalculate the threshold using the OTSU method and apply it to obtain a thresholded image of the subsection around a specific tree.

Finally, we concatenate all the derived results from the subsections of the image to create a final thresholded image.

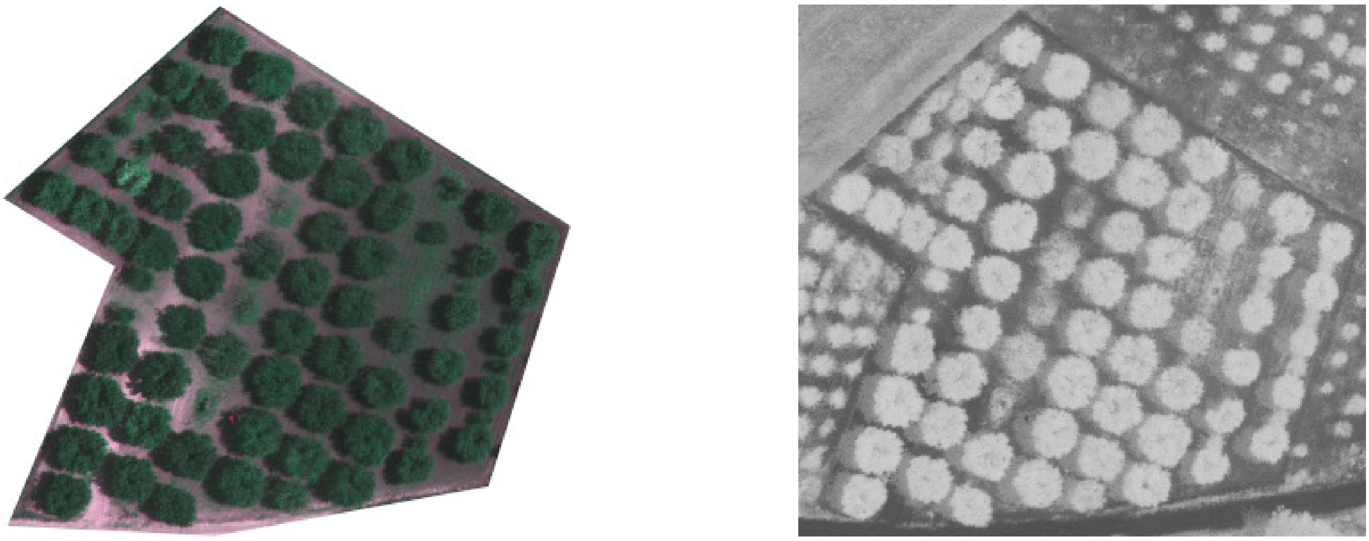
For example, Figure 6b displays the pseudocoloured grayscale image of the NDVI index for the orchard in Figure 6a. Figure 7a depicts the surrounding area of the image for a specific tree. In addition, Figure 8 displays the corresponding histogram and the detected threshold from the OTSU method for the specific tree. Furthermore, Figure 7b displays the same surrounding area for the specific tree after applying gamma correction.

Figure 7c displays the resulting black and white image of the two different classes defined by the OTSU method of the tree after the gamma correction. Finally, Figure 9 presents the overall image of the orchard, obtained by concatenating all thresholded images for each tree.

In the second stage of our method, we use this image as a reference to improve the mask of each tree detected from Detectron2 or YOLOv8. In our examples, we used the derived masks from Detectron2 since they provide better accuracy in cherry tree detection. Furthermore, the final masks from our method remain almost the same even when we choose as initial masks those from YOLOv8.

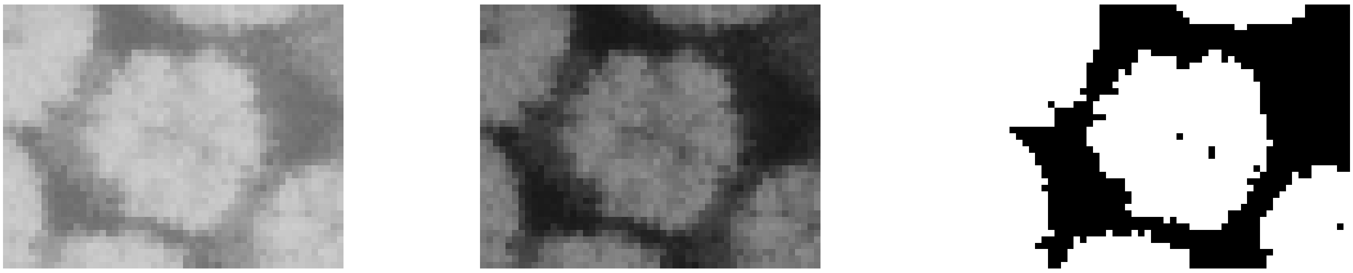
First, we remove all pixels from the perimeter of the mask where the corresponding pixels in the OTSU black and white image are equal to black. In addition, we remove pixels from the perimeter that belong to other masks. This part of the method resolves any overlaps with nearby trees.

Secondly, we search for nearby pixels of the mask where the corresponding pixel in the OTSU black-and-white image is equal to white. We conduct this process step by step for one additional pixel at a time along the perimeter of the existing mask. Throughout this procedure, we ensure that the additional pixels do not belong to other masks, preventing any overlap. At each step of our approach, we invert the order of the masks to ensure equal expansion when two cherry trees share the same area. We repeat this procedure until no additional pixels are left.



(**a**) RGB (**b**) NDVI

Figure 6. Orthomosaic image of an orchard with cherry trees: (**a**) in visible band (RGB), (**b**) in greyscale based on NDVI index.



(**a**) (**b**) (**c**)

Figure 7. Example of OTSU thresholding for a cherry tree: (**a**) Grayscale image of the surrounding area of a specific tree on the NDVI, (**b**) Grayscale image of the tree on NDVI after gamma correction, (**c**) Black and white image after applying OTSU thresholding method.

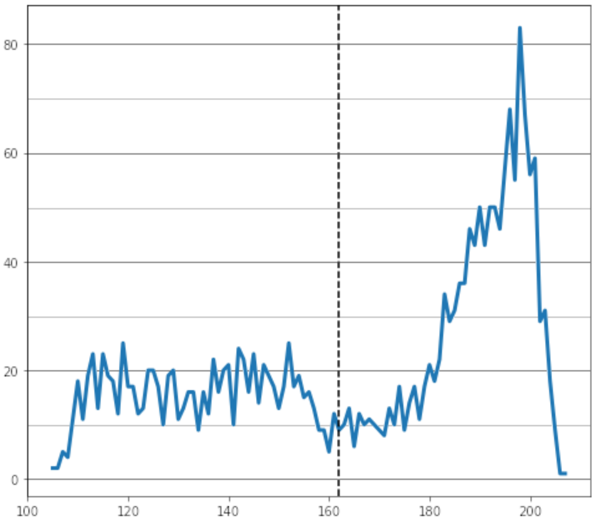


Figure 8. Histogram of the grayscaled image based on NDVI index for the surrounding area of a cherry tree.

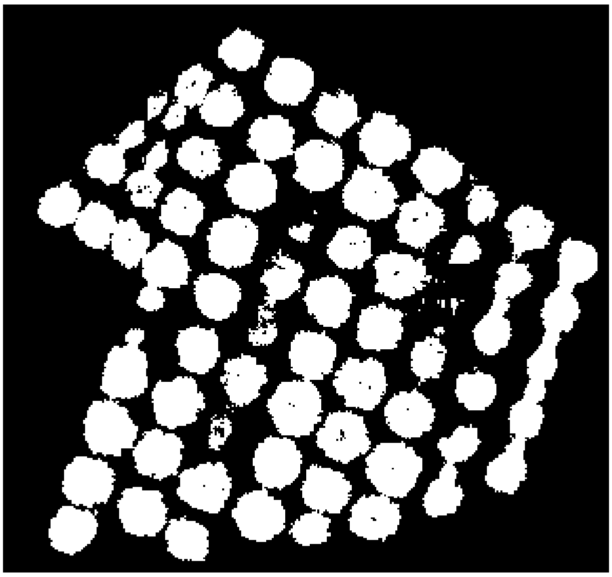
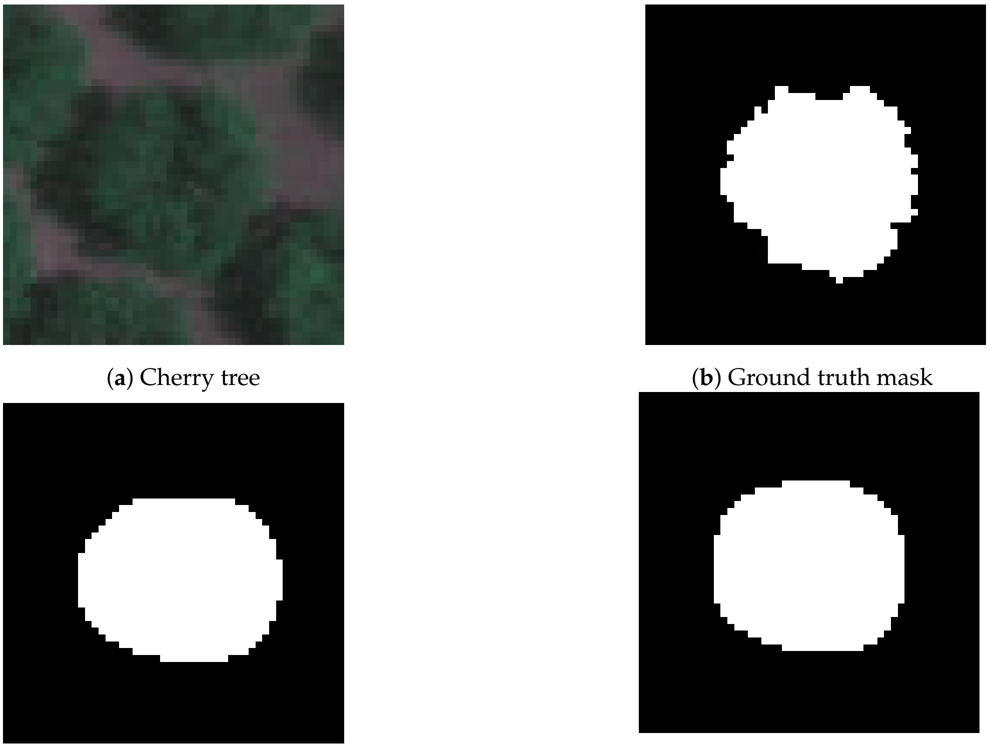


Figure 9. Mask of the orchard based on OTSU thresholding after gamma correction.

As an example, Figure 10c,d display masks detected by Detectron2 and YOLOv8, respectively, for the same cherry tree. The predicted masks are precise to ground truth (Figure 10b) but are not exactly the same. Figure 10a shows part of the orthomosaic photo from the specific cherry tree.



(**a**) Cherry tree

(**b**) Ground truth mask

(**c**) Mask from Detectron2

(**d**) Mask from YOLOv8

Figure 10. Comparison of the masks provided for a specific tree. (**a**) Cherry tree from the orchard. (**b**) Ground truth mask. (**c**) Mask from Detectron2. (**d**) Mask from YOLOv8.

Furthermore, Figure 11a shows the area removed or added to the initial mask using the OTSU method. Dark grey pixels indicate the removed area, while light grey pixels indicate the added area. Finally, Figure 11b shows the final mask of the detected tree after the step based on OTSU. Definitely, it aligns more precisely with the ground truth mask in

Figure 10b. Figure 11c illustrates the corresponding image in RGB format and the perimeter of the final mask, highlighting the improvement achieved through the suggested method. It is important to note that the proposed method effectively addresses even the shadows from the trees as seen in Figure 11c. This is due to the characteristics of the NDVI used, which has low values in areas with no vegetation. The shadow is separated from the cherry trees as the NDVI values belong to different classes during the OTSU thresholding.

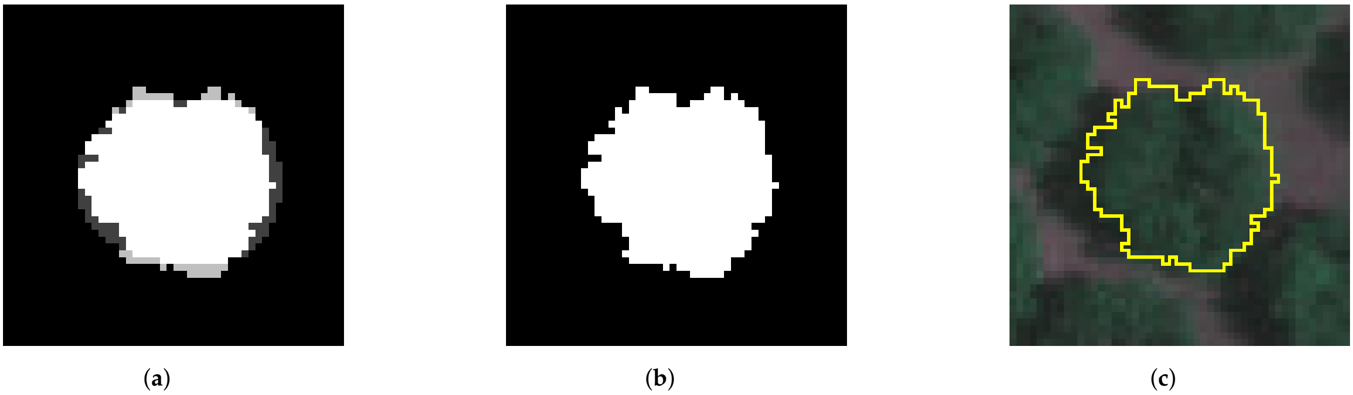


Figure 11. Example of changes in the mask of a specific tree. (**a**) Modifications on the provided mask from YOLOv8. (**b**) Final mask after OTSU thresholding. (**c**) The perimeter of the mask on the orthomosaic image for the specific cherry tree.

5. Evaluation

In this section, first, we present the evaluation of cherry tree detection accuracy using Detectron2 and YOLOv8. Secondly, we provide an evaluation of masks generated by Detectron2, YOLOv8 and the improved masks based on OTSU thresholding. The evaluation was conducted on orchards with cherry trees characterised by low weed and grass coverage.

5.1. Results of Cherry Tree Detection

After the training process, both Detectron2 and YOLOv8 are capable of detecting cherry trees and provide a mask for each one. To evaluate the accuracy of detection, we use the metric F1-Score. First, we have to calculate two other metrics, Precision (*P*) and Recall (*R*). The Precision metric corresponds to correctly identified trees divided by the total number of detected trees and is given in Equation (6). The Recall metric corresponds to correctly identified trees divided by the total number of actual trees and is given in Equation (6). Finally, we calculate the F1 score according to the Equation (8).

$$Precision = \frac{TP}{FP + TP} \tag{6}$$

$$Recall = \frac{TP}{FN + TP} \tag{7}$$

$$F1\ Score = \frac{2 \times P \times R}{P + R} \tag{8}$$

where *TP* (True Positive) measures the correct detected cherry trees, *FP* (False Positive) corresponds to falsely detected cherry trees, and *FN* (False Negative) corresponds to the number of cherry trees not detected by the algorithm.

In Table 6, we present a comparison of Detectron2 and YOLOv8. The first one utilises ResNet50 and ResNet101 as the backbone networks, while the second one employs YOLOv8m-seg, and YOLOv8x-seg as pre-trained models. The comparison reveals that Detectron2 with ResNet101 achieves slightly better performance (F1-Score: 94.85%) in cherry tree detection compared to ResNet50 (F1-Score: 93.88%). Additionally, YOLOv8's performance lags behind, with the highest achieved by YOLOv8x-seg (F1-Score: 87.51%) and the lowest by YOLOv8m-seg (F1-Score: 86.94%).

Table 6. Performance of Detectron2 and YOLOv8 on cherry tree detection.

Algorithm	Precision (%)	Recall (%)	F1 Score (%)
Detectron2 (ResNet50)	91.42	96.47	93.88
Detectron2 (ResNet101)	92.67	97.13	94.85
YOLOv8 (YOLOv8m-seg)	92.58	81.94	86.94
YOLOv8 (YOLOv8x-seg)	91.75	83.65	87.51

Figure 12 displays Precision and Recall values during the training process of Detectron2 for both pre-trained models based on ResNet-101 and ResNet-50. The peak F1-Score is achieved at iteration 15,680 and 15,200 for ResNet-101 and ResNet-50, respectively. Moreover, Figure 13 shows the Precision and Recall values during the training process of YOLOv8 for both pre-trained models YOLOv8x-seg and YOLOv8m-seg. The optimal F1-Score is attained at epochs 126 and 103 for YOLOv8x-seg and YOLOv8m-seg, respectively.

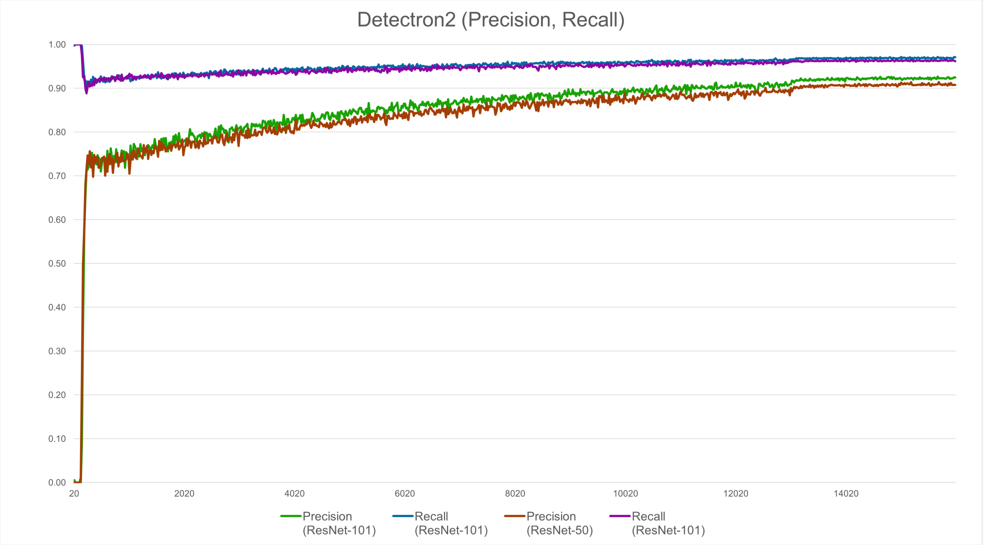


Figure 12. Convergence of Precision and Recall in Detectron2 with ResNet-101 and ResNet-50.

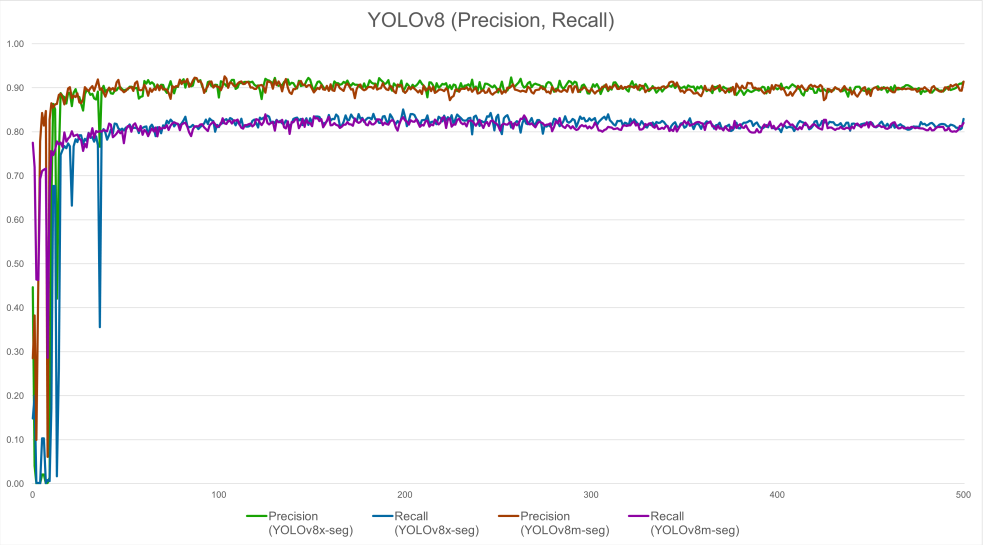
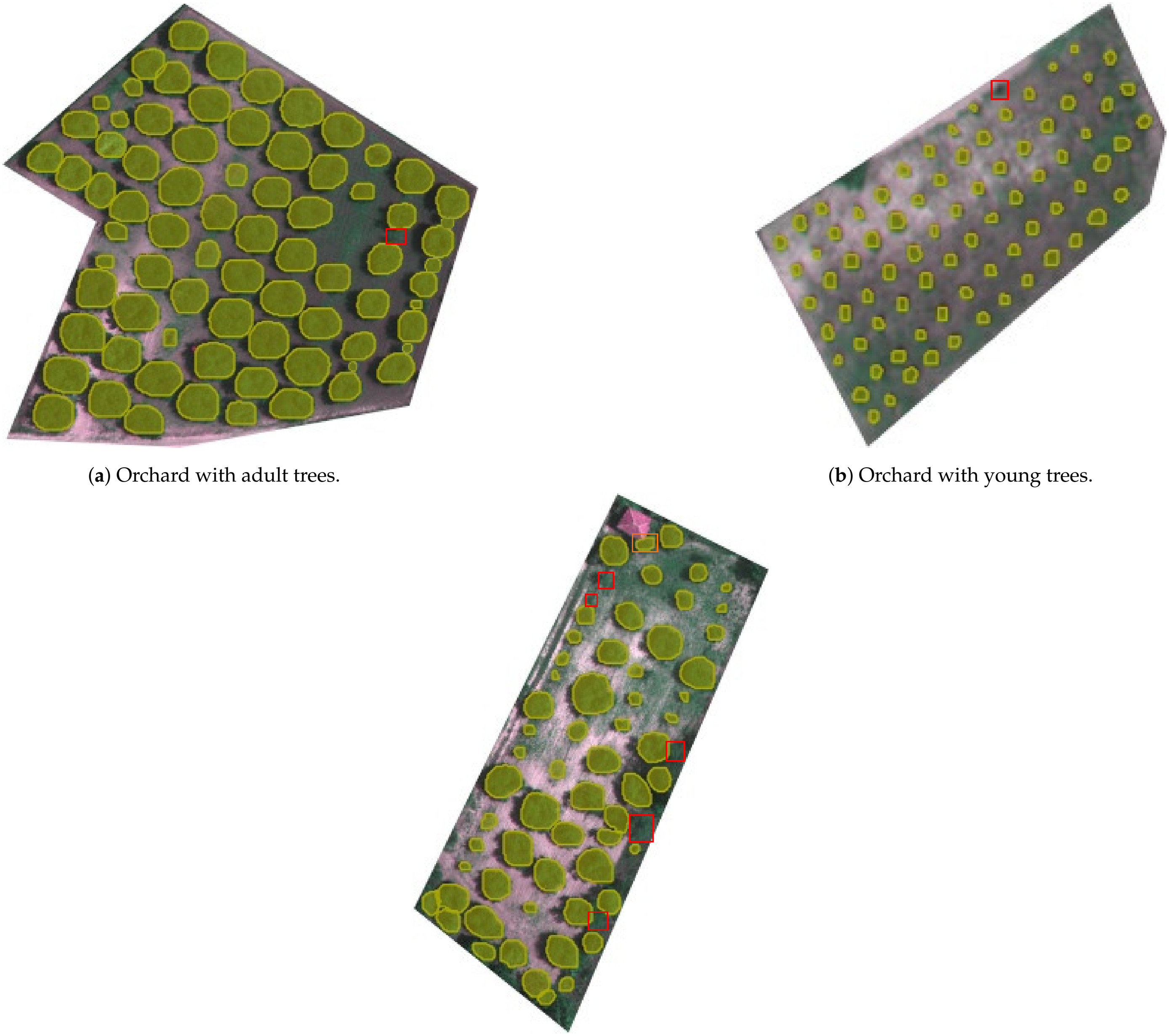


Figure 13. Convergence of Precision and Recall in YOLOv8 with YOLOv8x-seg and YOLOv8m-seg.

Although both algorithms detect some false trees or do not detect others, most problems occur with young trees whose crowns are too small or with trees that have no leaves making them difficult to discern in the captured UAV image.

Figure 14a shows the trees detected by Detectron2 in an orchard that has mostly adult trees and a moderate presence of grasses and weeds.. In this example, both algorithms successfully identified all cherry trees, except those with no leaves or some young trees close to others (red rectangles). Figure 14b shows another example of detected trees in an orchard where many trees are young with small crowns. However, both Detectron2 and YOLOv8 detect almost all of them, with the exception of one tree (red rectangle). Finally, both algorithms give satisfactory results even in fields with unclear terrain with high weed coverage. Figure 14c shows such an example, where some cherry trees are not detected (red rectangle) and one tree is falsely detected (orange rectangle).



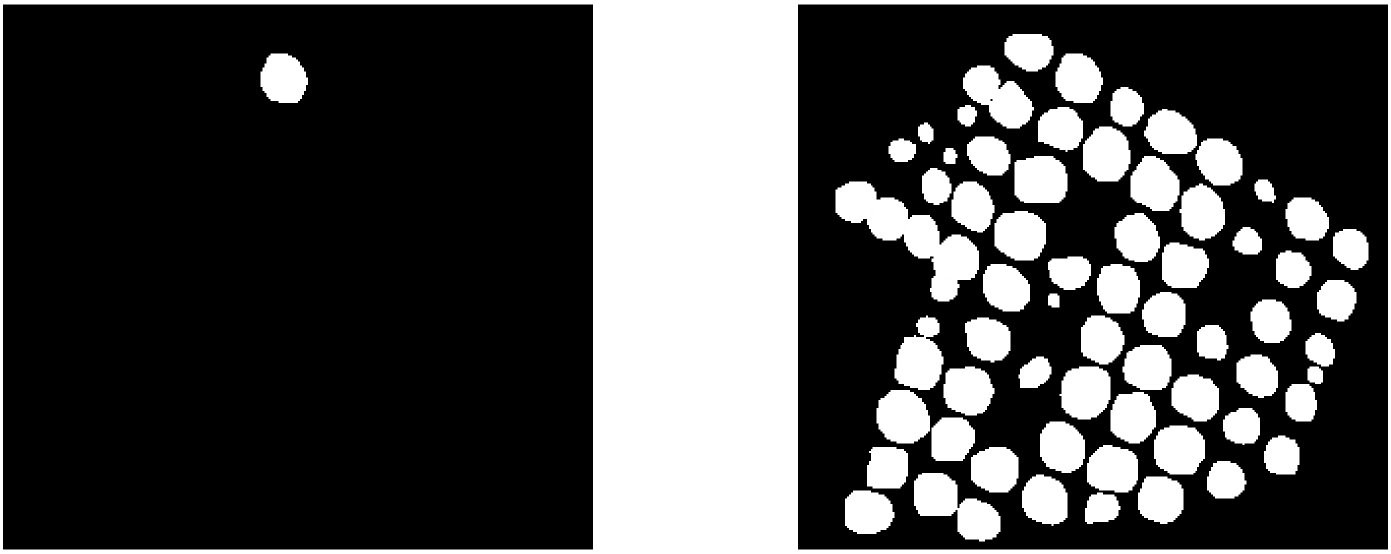
(**a**) Orchard with adult trees.

(**b**) Orchard with young trees.

(**c**) Orchard with high weed and grass coverage.

Figure 14. Examples of detected cherry trees in orchards: (**a**) Orchard that has mostly adult trees and moderate coverage in grass and weeds, (**b**) Orchard with young trees, (**c**) Orchard with high presence of weeds and grass.

Both algorithms generate a corresponding mask for the crowns of the detected trees. As an example, Figure 15a shows the corresponding mask of a specific tree detected with Detectron2 from the orchard of Figure 14a, while Figure 15b presents a composite image of all masks from the detected trees in the orchard.



(**a**) Final mask of a cherry tree.

(**b**) All masks combined in one image.

Figure 15. Detected masks from Detectron2: (**a**) Corresponding mask of a cherry tree. (**b**) All masks of the orchard combined in one image.

5.2. Results of Mask Improvement

To evaluate the effectiveness of our proposed additional algorithm, we manually delineated the crowns of 812 cherry trees from three different orchards with low weed and grass coverage. Each orchard was used four times, corresponding to four flights in two years. The masks derived from this precise annotation were considered the ground truth masks of the tree's crowns. To evaluate the masks provided by Detectron2, YOLOv8 and those after the OTSU thresholding, we used the Intersection over Union (IoU) metric between them and their corresponding ground truth mask. Intersection over Union was calculated with Equation (9), as the ratio of the intersection between the ground truth mask and the detected mask of the object to the union of the two masks. Alternatively, IoU can be defined with Equation (10), where TP_a (True Positive Area) is the area detected and belongs to the ground truth mask, FP_a (False Positive Area) is the falsely detected area, and FN_a (False Negative Area) is the area belonging to the ground truth but not detected.

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \tag{9}$$

$$IoU = \frac{TP_a}{TP_a + FP_a + FN_a} \tag{10}$$

Moreover, we defined $FPoU$ (Equation (11)) as the ratio between FP_a and the union of the predicted mask with the ground truth mask.

$$FPoU = \frac{FP_a}{TP_a + FP_a + FN_a} \tag{11}$$

Finally, we defined $FNoU$ (Equation (12)) as the ratio between FN_a and the union of the predicted mask with the ground truth mask.

$$FNoU = \frac{FN_a}{TP_a + FP_a + FN_a} \tag{12}$$

Table 7 shows the evaluation results between the ground truth masks and the masks provided by Detectron2, YOLOv8 and the additional improvement method based on OTSU thresholding.

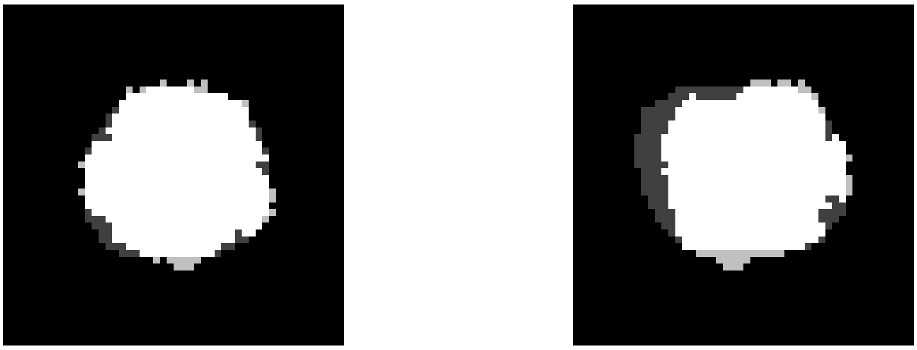
Table 7. Evaluation of mask accuracy for Detectron2, YOLOv8 and the additional algorithm based on OTSU thresholding.

Algorithm	IoU (%)	FPoU (%)	FNoU (%)
Detectron2	79.53	10.54	9.92
YOLOv8	75.36	14.07	10.56
OTSU	85.30	6.39	8.31

The proposed improvement method based on OTSU thresholding achieves better results in all metrics. More specifically, for IoU it reaches 85.30% while Detectron2 reaches 79.53% and YOLOv8 has 75.36. In addition, in the FPoU% metric, where lower values are better, our improvement gives 6.39%, Detectron2 10.54% and YOLOv8 14.07%. Finally, in the FNoU metric, where lower is better, the proposed method achieves 8.31%, while Detectron2 and YOLOv8 give 9.92% and 10.56%, respectively.

6. Discussion

Regarding our research on tree detection and crown extraction, Detectron2 and YOLOv8 can accurately detect cherry trees and extract a mask covering their crowns. Although most masks do not need too much correction, in some cases, a significant improvement is appropriate, and the proposed algorithm is suitable to deliver. For example, Figure 16a shows a slightly improved mask, while Figure 16b shows a mask significantly improved.



(**a**) Slightly improved mask.

(**b**) Significantly improved mask.

Figure 16. Examples of improved masks based on the OTSU thresholding. Dark grey pixels indicate the removed area, while light grey pixels indicate the added area.

The final masks correspond significantly better to each individual tree, enabling us to extract valuable information based on vegetation indices such as NDVI, NDRE or SAVI.

Equation (13) defines the NDVI index, which is frequently used for remote sensing. As an example, we used the generated masks by our method on the NIR and RED channels to calculate the NDVI for the corresponding cherry tree. The resulting NDVI pseudocolour images for four different flights from a cherry tree are displayed in Figure 17.

$$NDVI = \frac{NIR - Red}{NIR + Red} \tag{13}$$

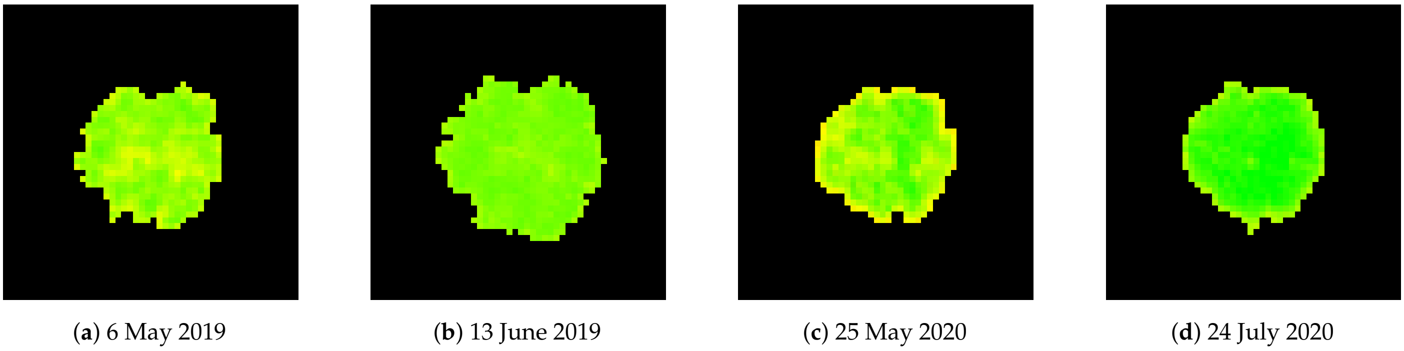

(**a**) 6 May 2019 (**b**) 13 June 2019 (**c**) 25 May 2020 (**d**) 24 July 2020

Figure 17. NDVI for a specific tree in four different flights. (**a**) On 6 May 2019. (**b**) On 13 June 2019. (**c**) On 25 May 2020. (**d**) On 24 July 2020.

Many research articles propose methods for detecting stress or identifying specific plant diseases using various vegetation indices. It is proved that multispectral cameras can provide additional information regarding the health status of the cultivation. In cultivations with full field coverage, such as wheat or barley, it is essential to analyse vegetation indices and divide the field into distinct areas based on the values of these indices. Such an approach may reveal potential diseases in areas with lower vegetation index values. In the case of orchards, where each tree can be defined as identical to the others, it may be helpful to examine the vegetation index separately for each tree. This can reveal trees with stress that should be further examined, as orthomosaic images from UAVs may not provide sufficient information at the leaf level. A closer examination, either with a ground multispectral camera or by an agronomist expert, is crucial for further estimation of potential diseases.

Furthermore, the proposed method demonstrates accurate performance in orchards with low to mid coverage of grass and weeds. In orchards where almost the entire free area is covered with grass, Detectron2 and YOLOv8 can still detect cherry trees with acceptable accuracy. However, the additional method for mask improvement may not yield optimal results. Nevertheless, we consider this to be non-crucial, as farmers can ensure the cleanliness of their orchards before a UAV flight if they intend to utilise such a service.

In addition, the proposed method for mask improvement is capable of removing shadows from the detected trees, even if they have been included in the generated masks by Detectron2 and YOLOv8. Figure 18 illustrates an example of the effective removal of a tree shadow. More specifically, Figure 18a shows part of the orthomosaic image where the shadow of the tree can be seen on the left. Figure 18b illustrates the generated mask from Detectron2, which includes part of the shadow. In Figure 18c, the NDVI index is displayed after gamma correction, indicating that the pixels belonging to the shadow have noticeably different values than those from the cherry tree. Thus, after applying the OTSU method, we separate them into different classes. Finally, Figure 18d, displays the improved mask derived from our method where it is clear that the shadow has been excluded.

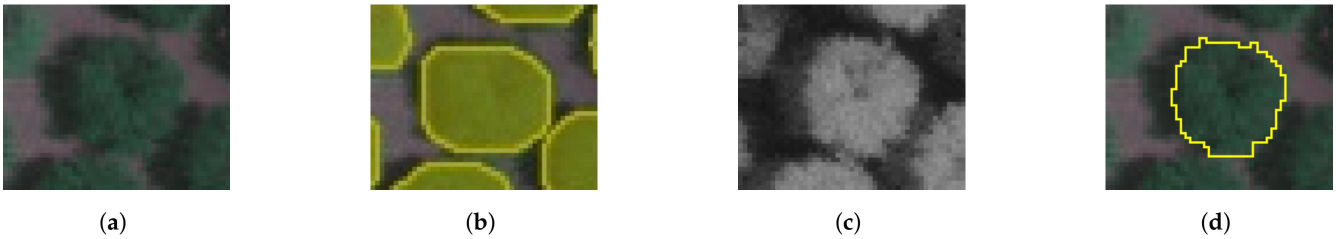

(**a**) (**b**) (**c**) (**d**)

Figure 18. Example of excluding the shadow of a cherry tree. (**a**) Part of the orthomosaic image; (**b**) Mask detected from Detectron2; (**c**) NDVI after gamma correction; (**d**) Final mask with no shadow.

Moreover, our method works well even when some of the trees are close enough. In a few instances, Detectron2 and YOLOv8 may create overlapped masks. The proposed method is designed to equally separate the common area to generate individual masks for each of the involved cherry trees. For example, Figure 19a shows the original image with two nearby cherry trees. Subsequently, Figure 19b displays the detected masks from Detectron2, while Figure 19c shows the masks after they have been separated using the proposed method.

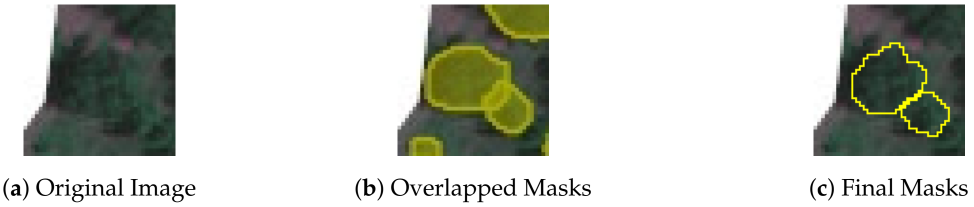

(**a**) Original Image (**b**) Overlapped Masks (**c**) Final Masks

Figure 19. Resolving overlapped masks. (**a**) Original image. (**b**) Overlapped masks from Detectron2. (**c**) Separated masks from the proposed method.

Although the trained models deliver satisfactory performance on the current orthomosaic images, augmenting the dataset is crucial. Our current dataset comprises orthomosaic photos from four UAV flights, lacking diversity. Additionally, the low resolution of the acquired orthomosaic images, attributed to flight height and camera limitations, poses challenges. Additional photos captured by different cameras and under various weather conditions may address these limitations. Moreover, flights with multicopters that are able to fly at lower heights can acquire images with higher resolution. This strategic enhancement aims to create a more comprehensive and generic dataset and increase the detection of cherry trees across a diverse range of orthomosaic photos.

Regarding using the proposed method on different tree species, the most crucial factor is the dataset selection. A high diversity of orthomosaic images with different weather conditions and different types of cameras and UAVs may help. Challenges may arise when dealing with tree species lacking rich and thick foliage or those with very small crowns, as mentioned earlier. Our method is specifically designed for orchards with moderately spaced trees. While it performs well when the crowns of some trees are connected, it may face limitations in situations where trees are closely planted side by side in rows.

7. Conclusions

Remote sensing has become one of the most valuable tasks in smart farming, enabling the observation of crop stress and facilitating timely decision-making for farmers and agronomists. UAV-captured images, particularly those with multispectral or hyperspectral information, serve as the primary source for this purpose.

In addition, various vegetation indices can extract valuable information from UAV images. In many cases, where the cultivation covers the entire field, such as with wheat or barley, we can use orthomosaic images captured from UAVs without any other preprocessing. However, in the case of orchards, it is essential to extract the crown for each individual tree before using it for further processing, such as calculating specific vegetation indices.

In this paper, we present a method for detecting the crown of each individual tree in an orchard and then improve the provided mask. We evaluate our method on orchards with cherry trees and show that it is feasible to detect them and provide individual masks with high accuracy. More specifically, we evaluated Detectron2 and YOLOv8 on cherry tree detection, each returning precise masks of the detected trees. Moreover, we propose a method to improve the provided mask, aiming for a more precise coverage of the tree crowns. The evaluation of our method shows that both Detectron2 and YOLOv8 can accurately detect cherry trees in orchards, achieving F1 scores up to 94.85% and 87.51%, respectively. Furthermore, the proposed improvement for the provided masks reaches an effectiveness of up to 85.30% on the Intersection over Union metric, when Detectron2 gives 79.53%, and YOLOv8 gives 75.36%.

Finally, we present an example for calculating vegetation indices like NDVI based on the provided masks by our method. In addition, the proposed method has been evaluated on cherry trees but can be easily adapted to any tree species. Therefore, we believe that our method can be a valuable tool for crop stress identification using aerial images for remote sensing in orchards.

In future work, we intend to use our method as a tool to detect possible stress on cherry trees. Differences in vegetation indices between trees of the same orchard or the same tree in different seasons may unveil early signs of possible diseases. Subsequently, additional information from ground multispectral cameras can further aid in identifying potential diseases affecting the specified trees.

Author Contributions: Conceptualization, V.M. and P.S.; methodology, V.M. and G.K.; software, V.M.; validation, G.K., V.A. and S.K.G.; formal analysis, V.M. and T.L.; investigation, V.M.; resources, V.M. and I.S.; data curation, V.M. and V.A.; writing—original draft preparation, V.M.; writing—review and editing, I.S. and T.L.; visualization, V.M. and S.K.G.; supervision, P.S.; project administration, P.S.; funding acquisition, P.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received funding from the European Union’s Horizon Europe Research and Innovation programme under Grant Agreement No. 101135800 (RAIDO). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to restrictions on privacy.

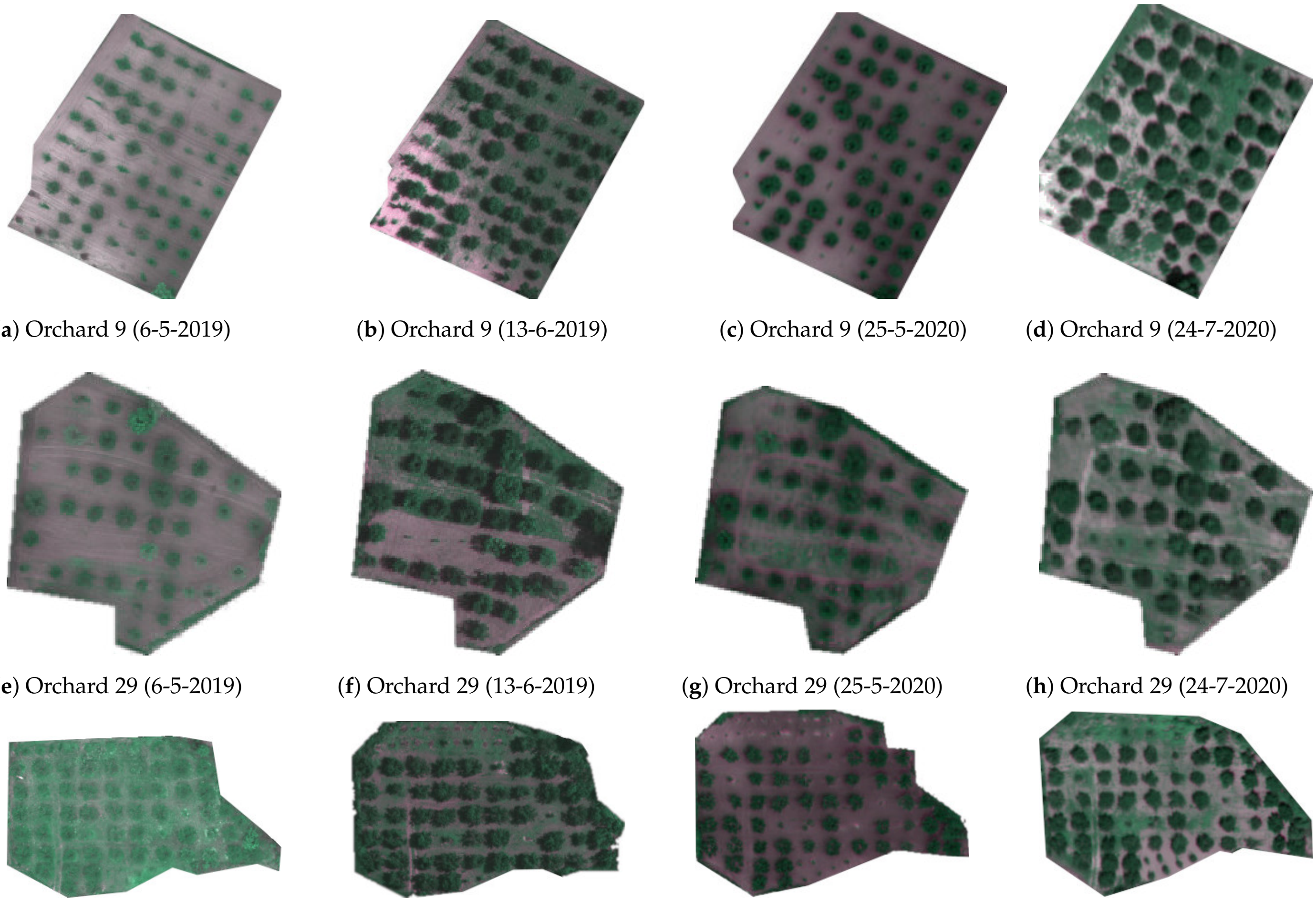
Conflicts of Interest: Established in May 2021, MetaMind Innovations is the first spin-off of the University of Western Macedonia. To the knowledge of the authors, there are no immediate conflicts of interest with this work, the reviewing process or with respect to the prestigious Journal’s procedures. There is a loyalty agreement between the University of Western Macedonia and the MetaMind Innovations, which defines the copyright dependencies between the two members while resolving any conflict of interest issues.

Abbreviations

The following abbreviations are used in this manuscript:

CCL	Connected Components Labelling
CNN	Convolutional Neural Network
FN	False Negative
FP	False Positive
FPN	Feature Pyramid Network
IoU	Intersection over Union
LiDAR	Light Detection And Ranging
LM	Local Maxima
MCWS	Marker-Controlled Watershed Segmentation
NDRE	Normalised Difference Red Edge
NDVI	Normalised Difference Vegetation Index
P	Precision
R	Recall
RoI	Regions of Interest
RPN	Region Proposal Network
SAVI	Soil-Adjusted Vegetation Index
SfM	Structure from Motion
SSD	Single Shot Detector
SVM	Support Vector Machine
TP	True Positive
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicles
VIA	VGG Image Annotator
YOLO	You Only Look Once

Appendix A



(**a**) Orchard 9 (6-5-2019) (**b**) Orchard 9 (13-6-2019) (**c**) Orchard 9 (25-5-2020) (**d**) Orchard 9 (24-7-2020)

(**e**) Orchard 29 (6-5-2019) (**f**) Orchard 29 (13-6-2019) (**g**) Orchard 29 (25-5-2020) (**h**) Orchard 29 (24-7-2020)

(**i**) Orchard 52 (6-5-2019) (**j**) Orchard 52 (13-6-2019) (**k**) Orchard 52 (25-5-2020) (**l**) Orchard 52 (24-7-2020)

Figure A1. Samples from the dataset from three different orchards. Four orthomosaic images were captured for each orchard in two consecutive years.

References

1. Moysiadis, V.; Sarigiannidis, P.; Vitsas, V.; Khelifi, A. Smart farming in Europe. *Comput. Sci. Rev.* **2021**, *39*, 100345. [CrossRef]
2. Diez, Y.; Kentsch, S.; Fukuda, M.; Caceres, M.L.L.; Moritake, K.; Cabezas, M. Deep learning in forestry using uav-acquired rgb data: A practical review. *Remote Sens.* **2021**, *13*, 2837. [CrossRef]
3. Santos, A.A.d.; Marcato Junior, J.; Araújo, M.S.; Di Martini, D.R.; Tetila, E.C.; Siqueira, H.L.; Aoki, C.; Eltner, A.; Matsubara, E.T.; Pistori, H.; et al. Assessment of CNN-based methods for individual tree detection on images captured by RGB cameras attached to UAVs. *Sensors* **2019**, *19*, 3595. [CrossRef]
4. Gini, R.; Sona, G.; Ronchetti, G.; Passoni, D.; Pinto, L. Improving tree species classification using UAS multispectral images and texture measures. *ISPRS Int. J. -Geo-Inf.* **2018**, *7*, 315. [CrossRef]
5. Hanapi, S.S.; Shukor, S.; Johari, J. A review on remote sensing-based method for tree detection and delineation. *IOP Conf. Ser. Mater. Sci. Eng.* **2019**, *705*, 012024. [CrossRef]
6. Qiu, L.; Jing, L.; Hu, B.; Li, H.; Tang, Y. A New Individual Tree Crown Delineation Method for High Resolution Multispectral Imagery. *Remote Sens.* **2020**, *12*, 585. [CrossRef]
7. Naveed, F.; Hu, B.; Wang, J.; Hall, G.B. Individual Tree Crown Delineation Using Multispectral LiDAR Data. *Sensors* **2019**, *19*, 5421. [CrossRef]
8. Rizeei, H.M.; Shafri, H.Z.; Mohamoud, M.A.; Pradhan, B.; Kalantar, B. Oil palm counting and age estimation from WorldView-3 imagery and LiDAR data using an integrated OBIA height model and regression analysis. *J. Sensors* **2018**, *2018*, 2536327. [CrossRef]
9. Dong, X.; Zhang, Z.; Yu, R.; Tian, Q.; Zhu, X. Extraction of Information about Individual Trees from High-Spatial-Resolution UAV-Acquired Images of an Orchard. *Remote Sens.* **2020**, *12*, 133. [CrossRef]
10. Wallace, L.; Lucieer, A.; Malenovský, Z.; Turner, D.; Vopěnka, P. Assessment of Forest Structure Using Two UAV Techniques: A Comparison of Airborne Laser Scanning and Structure from Motion (SfM) Point Clouds. *Forests* **2016**, *7*, 62. [CrossRef]

11. Li, L.; Dong, J.; Njeudeng Tenku, S.; Xiao, X. Mapping Oil Palm Plantations in Cameroon Using PALSAR 50-m Orthorectified Mosaic Images. *Remote Sens.* **2015**, *7*, 1206–1224. [CrossRef]
12. G. Braga, J.R.; Peripato, V.; Dalagnol, R.; P. Ferreira, M.; Tarabalka, Y.; O. C. Aragão, L.E.; F. de Campos Velho, H.; Shiguemori, E.H.; Wagner, F.H. Tree Crown Delineation Algorithm Based on a Convolutional Neural Network. *Remote Sens.* **2020**, *12*, 1288. [CrossRef]
13. Sultana, F.; Sufian, A.; Dutta, P. A review of object detection models based on convolutional neural network. In *Intelligent Computing: Image Processing Based Applications*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 1–16.
14. Chang, A.; Yeom, J.; Jung, J.; Landivar, J. Comparison of Canopy Shape and Vegetation Indices of Citrus Trees Derived from UAV Multispectral Images for Characterization of Citrus Greening Disease. *Remote Sens.* **2020**, *12*, 4122. [CrossRef]
15. Zhao, T.; Yang, Y.; Niu, H.; Wang, D.; Chen, Y. Comparing U-Net convolutional network with mask R-CNN in the performances of pomegranate tree canopy segmentation. In *Multispectral, Hyperspectral, and Ultraspectral Remote Sensing Technology, Techniques and Applications VII*; SPIE: Bellingham, WA, USA, 2018; Volume 10780; pp. 210–218.
16. Yu, K.; Hao, Z.; Post, C.J.; Mikhailova, E.A.; Lin, L.; Zhao, G.; Tian, S.; Liu, J. Comparison of Classical Methods and Mask R-CNN for Automatic Tree Detection and Mapping Using UAV Imagery. *Remote Sens.* **2022**, *14*, 295. [CrossRef]
17. Otsu, N. A Threshold Selection Method from Gray-Level Histograms. *IEEE Trans. Syst. Man Cybern.* **1979**, *9*, 62–66. [CrossRef]
18. Donmez, C.; Villi, O.; Berberoglu, S.; Cilek, A. Computer vision-based citrus tree detection in a cultivated environment using UAV imagery. *Comput. Electron. Agric.* **2021**, *187*, 106273. [CrossRef]
19. Ocer, N.E.; Kaplan, G.; Erdem, F.; Kucuk Matci, D.; Avdan, U. Tree extraction from multi-scale UAV images using Mask R-CNN with FPN. *Remote Sens. Lett.* **2020**, *11*, 847–856. [CrossRef]
20. Li, W.; Dong, R.; Fu, H.; Yu, L. Large-scale oil palm tree detection from high-resolution satellite images using two-stage convolutional neural networks. *Remote Sens.* **2018**, *11*, 11. [CrossRef]
21. Zortea, M.; Macedo, M.M.; Mattos, A.B.; Ruga, B.C.; Gemignani, B.H. Automatic citrus tree detection from UAV images based on convolutional neural networks. In Proceedings of the 31th Sibgrap/WIA—Conference on Graphics, Patterns and Images, SIBGRAPI, Paraná, Brazil, 29 October–1 November 2018; Volume 18.
22. Iqbal, M.S.; Ali, H.; Tran, S.N.; Iqbal, T. Coconut trees detection and segmentation in aerial imagery using mask region-based convolution neural network. *IET Comput. Vis.* **2021**, *15*, 428–439. [CrossRef]
23. Zarea, A.; Mohammadzadeh, A. A novel building and tree detection method from LiDAR data and aerial images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *9*, 1864–1875. [CrossRef]
24. Eysn, L.; Hollaus, M.; Lindberg, E.; Berger, F.; Monnet, J.M.; Dalponte, M.; Kobal, M.; Pellegrini, M.; Lingua, E.; Mongus, D.; et al. A benchmark of lidar-based single tree detection methods using heterogeneous forest data from the alpine space. *Forests* **2015**, *6*, 1721–1747. [CrossRef]
25. Chaschatzis, C.; Karaiskou, C.; Mouratidis, E.G.; Karagiannis, E.; Sarigiannidis, P.G. Detection and Characterization of Stressed Sweet Cherry Tissues Using Machine Learning. *Drones* **2021**, *6*, 3. [CrossRef]
26. Gonzalez-Huitron, V.; León-Borges, J.A.; Rodriguez-Mata, A.; Amabilis-Sosa, L.E.; Ramírez-Pereda, B.; Rodriguez, H. Disease detection in tomato leaves via CNN with lightweight architectures implemented in Raspberry Pi 4. *Comput. Electron. Agric.* **2021**, *181*, 105951. [CrossRef]
27. Shin, J.; Chang, Y.K.; Heung, B.; Nguyen-Quang, T.; Price, G.W.; Al-Mallahi, A. A deep learning approach for RGB image-based powdery mildew disease detection on strawberry leaves. *Comput. Electron. Agric.* **2021**, *183*, 106042. [CrossRef]
28. Razfar, N.; True, J.; Bassiouny, R.; Venkatesh, V.; Kashef, R. Weed detection in soybean crops using custom lightweight deep learning models. *J. Agric. Food Res.* **2022**, *8*, 100308. [CrossRef]
29. Zhuang, J.; Li, X.; Bagavathiannan, M.; Jin, X.; Yang, J.; Meng, W.; Li, T.; Li, L.; Wang, Y.; Chen, Y.; et al. Evaluation of different deep convolutional neural networks for detection of broadleaf weed seedlings in wheat. *Pest Manag. Sci.* **2022**, *78*, 521–529. [CrossRef]
30. Wu, L.; Ma, J.; Zhao, Y.; Liu, H. Apple detection in complex scene using the improved YOLOv4 model. *Agronomy* **2021**, *11*, 476. [CrossRef]
31. Li, X.; Pan, J.; Xie, F.; Zeng, J.; Li, Q.; Huang, X.; Liu, D.; Wang, X. Fast and accurate green pepper detection in complex backgrounds via an improved Yolov4-tiny model. *Comput. Electron. Agric.* **2021**, *191*, 106503. [CrossRef]
32. Dong, S.; Du, J.; Jiao, L.; Wang, F.; Liu, K.; Teng, Y.; Wang, R. Automatic Crop Pest Detection Oriented Multiscale Feature Fusion Approach. *Insects* **2022**, *13*, 554. [CrossRef]
33. Tetila, E.C.; Machado, B.B.; Astolfi, G.; de Souza Belete, N.A.; Amorim, W.P.; Roel, A.R.; Pistori, H. Detection and classification of soybean pests using deep learning with UAV images. *Comput. Electron. Agric.* **2020**, *179*, 105836. [CrossRef]
34. Moysiadis, V.; Kokkonis, G.; Bibi, S.; Moscholios, I.; Maropoulos, N.; Sarigiannidis, P. Monitoring Mushroom Growth with Machine Learning. *Agriculture* **2023**, *13*, 223. [CrossRef]
35. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
36. Wu, Y.; Kirillov, A.; Massa, F.; Lo, W.Y.; Girshick, R. Detectron2, Version 2.0, Facebook Inc., 2019. Available online: https://github.com/facebookresearch/detectron2 (accessed on 10 October 2023).
37. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.

38. Jocher, G.; Chaurasia, A.; Qiu, J. YOLOv8, Version 8.0, Ultralitics, 2023. Available online: https://github.com/ultralytics/ultralytics (accessed on 10 October 2023).
39. Dutta, A.; Zisserman, A. The VIA annotation software for images, audio and video. In Proceedings of the 27th ACM International Conference on Multimedia, Nice, France, 21–25 October 2019; pp. 2276–2279. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.