remote sensing

MDPI

Article

Automated Health Estimation of *Capsicum annuum* L. Crops by Means of Deep Learning and RGB Aerial Images

Jesús A. Sosa-Herrera [1,*,†], Nohemi Alvarez-Jarquin [1,†], Nestor M. Cid-Garcia [1,†], Daniela J. López-Araujo [1,†] and Moisés R. Vallejo-Pérez [2,†]

[1] Laboratorio Nacional de Geointeligencia, CONACYT-Centro de Investigación En Ciencias de Información Geospacial, Aguascalientes 20313, Mexico
[2] Coordinación para la Innovación y Aplicación de la Ciencia y la Tecnología (CIACYT), CONACYT-Universidad Autónoma de San Luis Potosí, San Luis Potosí 78000, Mexico
* Correspondence: jasosahe@conacyt.mx
† These authors contributed equally to this work.

Abstract: Recently, the use of small UAVs for monitoring agricultural land areas has been increasingly used by agricultural producers in order to improve crop yields. However, correctly interpreting the collected imagery data is still a challenging task. In this study, an automated pipeline for monitoring *C. Annuum* crops based on a deep learning model is implemented. The system is capable of performing inferences on the health status of individual plants, and to determine their locations and shapes in a georeferenced orthomosaic. Accuracy achieved on the classification task was 94.5. AP values among classes were in the range of $[63, 100]$ for plant location boxes, and in $[40, 80]$ for foliar area predictions. The methodology requires only RGB images, and so, it can be replicated for the monitoring of other types of crops by only employing consumer-grade UAVs. A comparison with random forest and large-scale mean shift segmentation methods which use predetermined features is presented. NDVI results obtained with multispectral equipment are also included.

Keywords: deep learning; Mask RCNN; precision agriculture; UAVs

check for updates

Citation: Sosa-Herrera, J.A.; Álvarez-Jarquín, N.; Cid-García, N.M.; López-Araujo D.J.; Vallejo-Pérez R.M. Automated Health Estimation of *Capsicum annuum* L. Crops by Means of Deep Learning and RGB Aerial Images. *Remote Sens.* **2022**, *14*, 4943. https://doi.org/10.3390/rs14194943

Academic Editors: Hao Gan, Chao Chen, Yunchao Tang, Lufeng Luo, Liang Gong, Junfeng Gao, Ya Xiong and Huaibo Song

Received: 1 August 2022
Accepted: 26 September 2022
Published: 3 October 2022

1. Introduction

Chili pepper (*Capsicum* sp.) is one of the most widely known condiments worldwide. In particular, the *Capsicum annuum* L. genus is of great economic importance in Mexico, since it has the largest distribution in the country [1]. During the nursery stage, and throughout the production process, the crop is affected by several types of microorganisms that cause diseases in seedlings. As a consequence, plant counts and fruit production volume are reduced, which represents a major problem for farmers. The epidemiological monitoring of crops allows one to know the health status of the total plant population, helping with timely implementation of preventive or corrective agronomic practices, and thus, allowing us to obtain maximum yields at the lowest cost. For the implementation of a technified on-field monitoring, it is necessary to develop an effective inspection plan, including sampling patterns, determination of unit sizes, number of samples, and the establishment of a set of severity scales to evaluate standard area diagrams (SADs) [2]. Then, a database with the records of individual cases must be kept. To perform these tasks, it is required to have trained personnel that can generate reliable results for disease severity estimates. Usually, a lot of human labor is needed to perform the entire monitoring process as described above, which increments total production costs; therefore, effective automation methods for such labor have a major impact on the adequate use of production resources [3]. Technological help for continuously monitoring *C. Annuum* crop at an early plant life is necessary to optimize the management of detected diseases. There exist many techniques to approach the problem of disease detection in crops using remote sensing. Nowadays, equipping UAVs with different types of sensors to acquire crop images is a common practice. Low cost, widespread utilization, high resolution, and flexibility have made it one of the best options for this kind of data acquisition [4]. Although satellite images allow for greater coverage of crop areas, UAVs have become one of the most widely used remote applications due to their availability, and to their compliance with data accuracy required for digital image processing [5]. Analyzing the data and correctly interpreting the outcomes of such analyses still remains a challenge from different perspectives when looking for a practical solution. In the literature, it is usual to find works that interpret images obtained from hyperspectral, thermal, Near Infrared (NIR), and RGB sensors, among others, or a combination thereof; see, for instance, [6]. How the acquired images are processed has become the main issue. A recent publication asserts that the most used techniques are photogrammetry, vegetation indices, and machine learning strategies [7]. Many learning models have been applied to retrieve significant information from imagery. The most frequently used models are the Bayesian (which is flexible and fast, but requires human expertise in selecting an a priori model) [8,9], clustering (simple implementation, scalable and adaptable but dependent on initial conditions and has a high computational cost) [10,11], decision trees (their run time is logarithmic and the rules are easy to set and implement; however, they can yield different results with minimum data changes and the optimal tree may not be found) [12,13], instance-based models such as k-Nearest Neighbor (these are easy to implement and no training is required, but they are costly for large datasets and sensitive to noise) [14], artificial neural networks (can work with incomplete information, which gives them robustness to fault, but they may be highly costly and dependent on hardware) [15,16], regression (simple to implement and works well with different data sizes; however, results are dependent on data quality) [17,18], and Support Vector Machine or SVM (works well if classes are clearly identified but may underperform for large or noisy data sets) [19]. Among such tools, deep learning (DL) has been increasingly applied, mainly due to the affordability of powerful computers and open source frameworks for the implementations of learning algorithms [20]. DL deals better with noise and scale variations than classical techniques, and has been mainly applied in agriculture to detect pests, weeds, irrigation or drought levels, and plant diseases. Table 1 briefly summarizes some methodologies, focusing on the latter objects.

Table 1. Deep learning agricultural applications. Previous works applying deep learning in agricultural processes.

Detected Objects	Publications
pests	[21–25]
weeds	[26–30]
irrigation/drought levels	[31–36]
diseases	[37–39]

To aid in the analysis of vegetation images, there are different computer vision tasks that can be performed using deep neural networks (NN). One of them is image recognition, which assigns a label or a class to a digital image, while object detection is the process of locating entities inside an image in which a box is usually drawn to delimit the regions of interest. Aside from object detection operations, there is the semantic segmentation process, which occurs when each pixel of an image is labeled as belonging to a class, and it is possible to modify the number of classes. Another more accurate alternative is instance segmentation, in which the boundary of each object instance is drawn. This technique, unlike simple object detection, allows for the location and delimitation of objects that have irregular shapes. It can be seen that, as the challenges increase, the complexity of the techniques also increases.

Amidst the different DL methodologies focused on disease detection, the region-based convolutional neural network (RCNN) stands out due to its capacity to extract image features. Since its emergence, it has been improved, giving rise to Fast RCNN, Faster RCNN, and Mask RCNN, appearing in [40–42] respectively. Mask RCNN models have the property of not only being able to detect instances of objects; they are also able to delimit the area of occurrence for each instance, proving instance segmentation capabilities. These approaches have been used alone or in combination with other algorithms for disease detection. Table 2 shows the detection objectives, and the accuracy achieved in works found in the literature.

Table 2. Mask RCNN applications. Uses of Mask RCNN for plant disease detection.

Objective	Backbone	Accuracy	Publication
Fruit spot disease detection	ResNet-101	+96	[43]
Diverse strawberry disease detection	ResNet-50	81.37	[44]
	ResNet-101	82.43	
Apple rust disease detection	ResNet-50	80.5	[45]
	MobileNet V3 Large	68.3	
	Large Mobile	53.7	

Canopy classification presents a challenging problem when orchard areas overlap. In such cases, traditional classification algorithms need to label individual plants manually to validate the models. This drawback can be overcome using a NN to automatically extract the relevant features at the convolutional layers from just a set of examples given to the network as training. Other technologies that deal with the overlapping problem include ones based on airborne LIDAR systems, but even when they possess several advantages, such as functionality in both daytime and nighttime, and the ability to be combined with other sensors, the expensive acquisition, costly data processing (in both time and computational resources), and low performance in some weathers [46], are major drawbacks for their implementation in real-life systems.

This paper presents a methodology based on the Mask RCNN deep learning ensemble to detect every plant cluster in the crop that originated at the same seeding point. The procedure localizes the plant objects and performs an instance segmentation of an image used as input, which represents a segmented portion of a large orthomosaic of the crop's area under study. The present work goes beyond performing instance segmentation, as the technique described here also estimates the health state of each plant cluster based on visual phenotypic features, implicitly extracted by the Mask R-CNN model. Unlike general purpose object detection that use regular CNNs, which aim to detect several types of unrelated objects under different background conditions, the model presented here is fine-tuned to detect plants and their discerning features present in a crop field environment. This level of specialization is intended for the purpose of not only detecting vegetation objects, but also to distinguish some of the features and visible traits that are correlated with the plant's health.

In the proposed model, only RGB aerial images were used. To verify the consistency of our results, we compare the method with results obtained from large-scale mean shift segmentation (LSMSS) composed with spatial KMeans [47,48] and random forest classification over local image filters [49], which both are methods in which predefined features are used. Additionally, spectral reflectance signatures for the plant samples were collected to ensure that the defined classes can be characterized not only by the plant's morphology and phenotypic traits, but also by their reflectance spectrum [50], as this latter property is the basis of many plant health indices [51], including the widely accepted normalized differential vegetation index (NDVI) [52]. The introduced methodology can be implemented as an automated pipeline to quantitatively determine the health state of *C. Annuum* crops in precise geolocalized manner.

2. Materials and Methods

2.1. Study Area

The experiments for this research were conducted on a *C. Annuum* crop field located at a rectangular region delimited by the coordinates (2610818N, 11429693W) and (2610758N, 11429543W) at 2205 m.a.s.l, in the municipality of Morelos, Zacatecas, Mexico. This specific portion of the field, was chosen in the interest of having a fair number of samples of plants with variable health conditions to define the comparative plant health classes. The region for the study has a *Cwb* climate according to the Köppen–Geiger categories [53], with average annual average temperature of 17 °C with minimum and maximum temperatures around 3 °C and 30 °C, respectively, and the annual rainfall is 510 mm [54].

2.2. Data Collection and Prepossessing

Airborne images were captured using multirotor UAVs, the Phantom III Standard® (SZ DJI Technology Co., Ltd., Shenzhen, China), equipped with an RGB camera with image resolution of 12 Mpx. The RGB image dataset was taken at 15 m from the ground, generating pictures with a resolution of 5.1 cm/px. With the purpose of comparison of the deep learning method proposed here, with respect to standard techniques of vegetation health assessment of crops from airborne images, a second multirotor equipped with a Sequoia Parrot® (Parrot SA, Paris, France) multispectral camera was flown over the same area and at the same height in order to generate a NDVI map. The multispectral camera captured reflectance levels at the near infrared (NIR) band, with center at 790 nm and 40 nm width, red edge (REG) band, centered at 735 nm with 10 nm width, red band (RED), centered at 660 nm with 40 nm width, and a green (GRE) band, with center at 550 nm and 40 nm of width. The Sequoia Parrot has a resolution of 1.2 megapixels for each of the individual spectral channels, which gave multiespectral images with a resolution of 11 cm/px. The images were post processed with the Pix4DMapper® (Pix4D, Lucerne, Switzerland) software, which was responsible for performing orthogonal rectification, pose estimation, vignetting and radiometric corrections for each picture, and generated the RGB and multispectral orthomosaics. Figure 1 shows the post processed RGB orthomosaic obtained with the unmanned aerial vehicle (UAV) survey of the study area, over the corresponding satellite image of the same region as background.

In addition to the UAVs imagery collected, the health state of every plant cluster located along two plowing rows of the cropland was also registered, their respective locations relative to a set of ground control points (GCP) placed at 3 m intervals were recorded. Five plant health condition classes, labeled from $HC1$ up to $HC5$, were established according to observable trait combinations of visible features associated with plant health status. The attributes considered were plant height (cm), foliar area (m^2), and percentage of canopy areas presenting disease symptoms (leaf spots, chlorosis, curly and wilting leaves). The characterization mentioned above was based on the average SAD maps of individual leaves. The threshold values for every tracked trait are shown in Figure 2, where the ranges of the maximum plant height, maximum canopy area, and percentage of damaged leaves are depicted as radial bar graphs for each of the $HC1, \ldots, HC5$ classes.

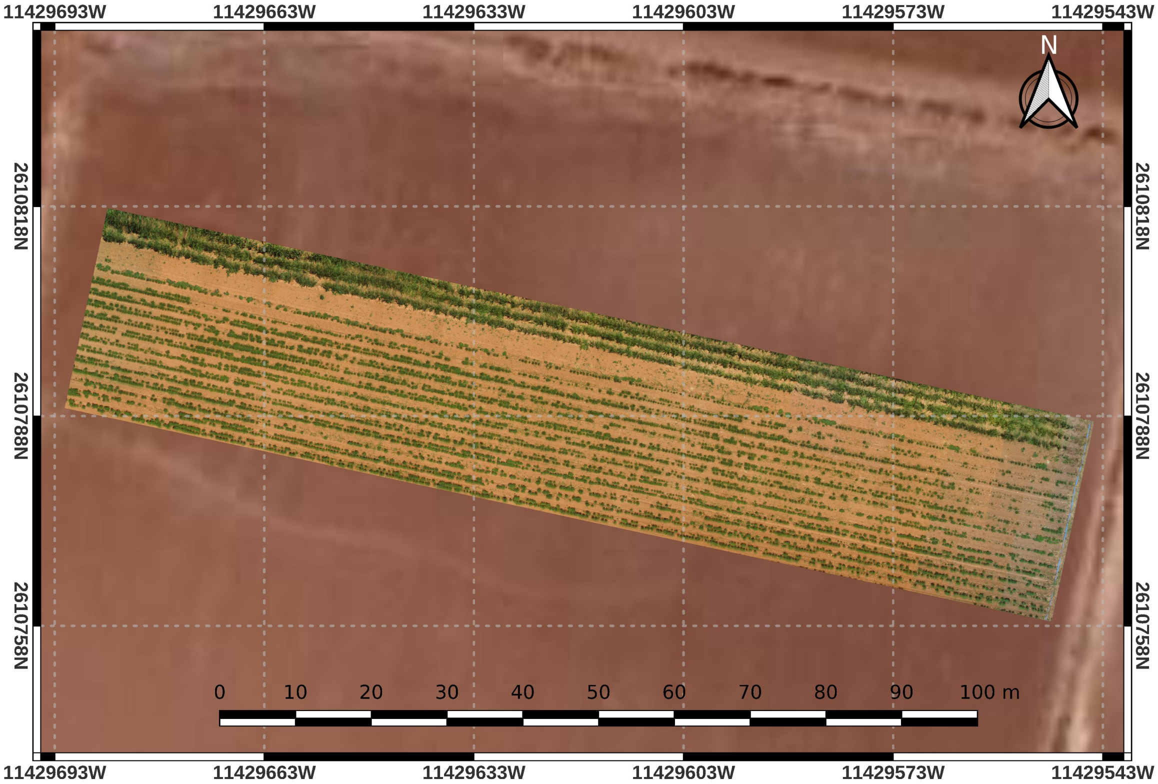


Figure 1. Study Area. RGB orthomosaic of the *C. Annuum* crop's portion under study.

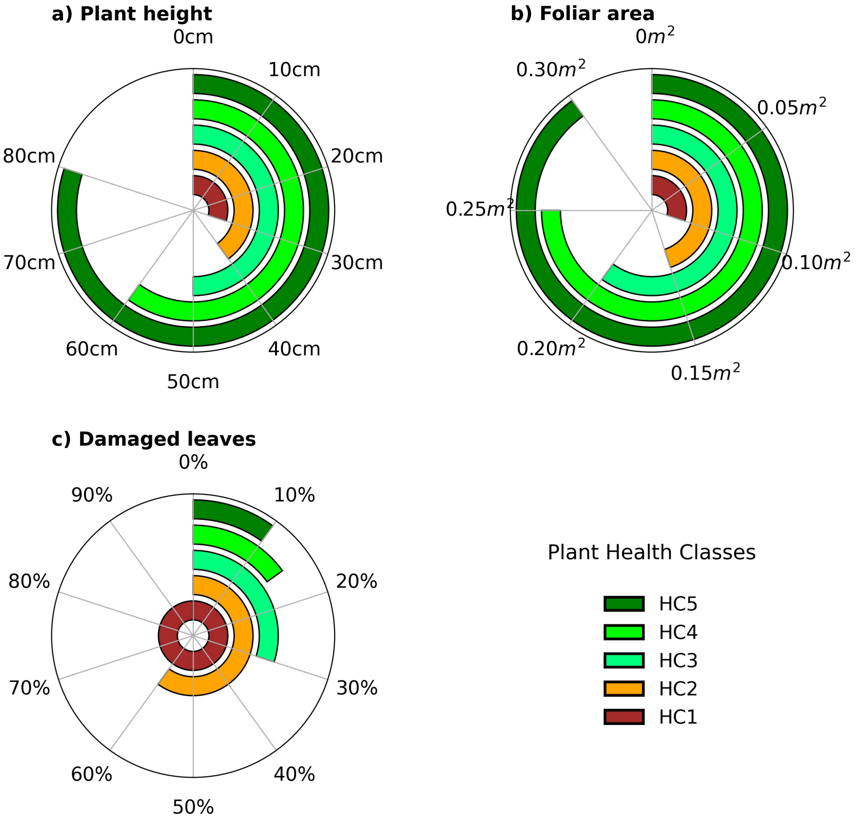


Figure 2. Plant health classes. Phenotype traits for each plant class.

For the task of collecting spectral signatures of plant samples, a custom portable spectrometer based on the $C12880MA$ (Hamamatsu Photonics, Shizuoka, Japan) sensor was used. The $C12880MA$ sensor is capable of detecting 288 spectral bands with centers at intervals of about 2 nm, in a wavelength range between 330 nm and 890 nm. The spectrometer was connected to a smartphone with GPS capabilities through the on-the-go (OTG) universal serial bus (USB) peripheral port [55]. An Arduino(Arduino.cc, Somerville, MA, USA) microcontroller was used to convert the inter integrated circuit (I2C) [56] bus signal from the sensor to USB serialized signal levels. A user interface programmed in Java language and the Android Development Studio® (Google Inc., Mountain View, CA, USA) tools was developed. In addition to recording and transmitting sensor signals, the application also took charge of attaching geotags to the spectral data, and of performing wavelength calibrations according to factory parameters [57]. Reflectance variations due to different intensity and illumination sources were also compensated by the program. The reference used for reflectance adjustments was a Micasense® (AgEagle Sensor Systems Inc., Wichita, KS, USA) calibration panel for which reflectance values in the range 300–900 nm at 2 nm intervals were provided by the manufacturer. The spectral signatures of 20 samples for every defined class were taken; the average spectrum for each class smoothed with a third degree polynomial is presented in Figure 3. The intervals corresponding to the four channels of the multispectral camera are used as x axis in order to highlight the reflectance features relevant to the health classes that can be captured by the multispectral camera from which the comparative NDVI map was built.

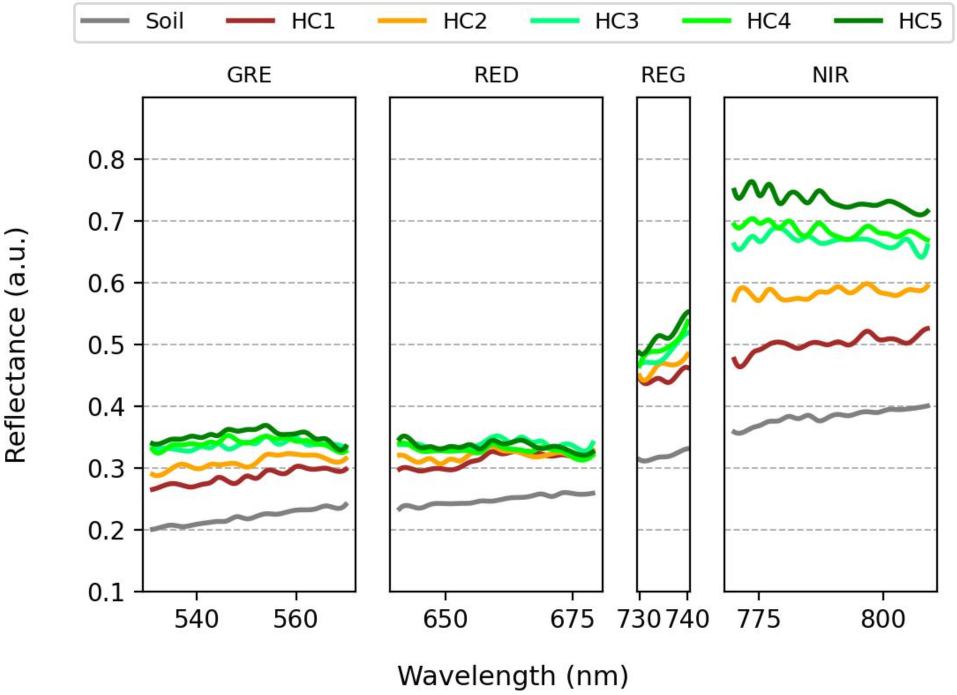


Figure 3. Plant health classes. Phenotype traits for each plant class.

Manual annotations regarding plant health classes, and canopy area pixel masks of the plant clusters appearing on each of the training and validation images were created. Labels and instance masks were outlined according to georeferenced data collected on-field. An example of image annotation masks for one of the aerial images from the crop can be seen in Figure 4a,b. Note that pixels at the edges of very small plant twigs for which their color were heavily mixed with background soil color, were not considered to be part of the plant's canopy, otherwise they would have induced noise in the reflectance features that differentiate health classes. A total of 60 images were annotated; 40 of them were used for training the model and 20 were used for validation and to estimate the Mask RCNN's hyperparameters. The annotated objects combined, added up to the amount of

2139 instances inside the images for which their respective contours, delimited by polygonal boundaries, were registered, thus providing a representative set of object instances of each health level to properly train the Mask RCNN ensemble.

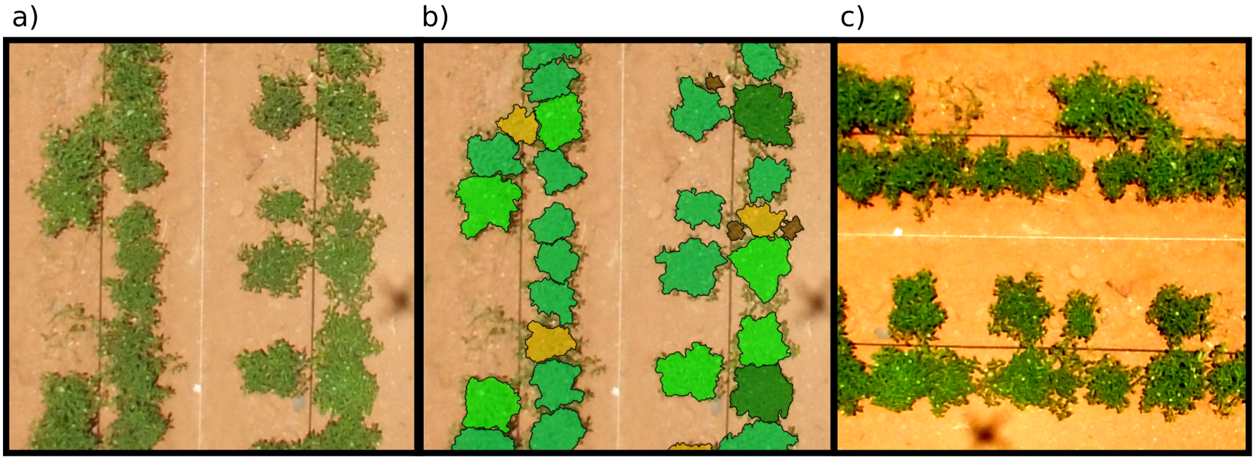


Figure 4. Training and validation datasets. (**a**) Original image of a 512 × 512 px section of the crop taken with a UAV. (**b**) Georeferenced instance segmentation masks elaborated from on field data. (**c**) Example of augmented data generated by applying randomly chosen transformations to the image.

2.3. Hardware and Software

Model implementation of the neural network ensemble was performed using the Detectron2® [58] deep learning framework, which has been released as open source [59] by Meta®AI Research. The original source code for Mask RCNN was made publicly available at the Detectron repository in [60] based on the Caffe [61] deep learning framework. Currently, a second revision of the framework based on Pytorch is available at the Detectron2 repository [59], which is the version of the code used here. Detectron2 acts as a wrapper for several Pytorch models, allowing them to interact together to conform ensembles of deep neural networks. Training, validation and testing of the model was executed using a workstation with CPU and GPU compute capabilities featuring an Intel Core®i7-9700FV (Intel Co., Santa Clara, CA, USA) CPU with 8 physical cores at 4.7 GHz and 32 GB of RAM. The GPU employed was a Nvidia GeForce RTX 3090® (Nvidia Co., Santa Clara, CA, USA) with 24 GB of video memory, with drivers configured with support for CUDA version 11.4. The Pytorch version used was 1.8. The main python scripts for data management and processing were elaborated on in the Visual Studio Code® (Microsoft, Redmond, WA, USA) integrated development environment (IDE). Georeferencing for use of the generated maps in standard GIS tools with a satellital base layer was carried out with the aid of QGIS [62] version 3.26.1 using the pseudo-Mercator projection [63]. Boundary polygons to generate instance training masks were made with the Computer Vision Annotation Tool [64]. In order to compare the instance segmentation and plant classification performed in this work with alternative traditional methods, we implemented a random forest classifier on local features (RFLF) extracted from the same images used to train the Mask RCNN ensemble. The local features images used the RFLF segmentation were extracted using the Laplacian-of-Gaussian detector [65], the Harris and Hessian affine regions [66] and a Gaussian blur filter to deal with features at different scales [67]. This was achieved using the Scikit-learn [68] libraries written in Python with the Scikit-image [69] extensions for digital image processing. Additionally to the RFLF reference segmentation, the LSMSS method using spatial KMeans was also compared with the technique presented in this research using the same validation set. The LSMSS version used here was taken from the orfeo toolbox [70] libraries, the spatial KMeans implementation was programmed as a

Python script based on Scikit-learn. The details for achieving object detection with LSMSS and KMeans are described in [48].

2.4. Data Augmentation and Class Balance

The original training image dataset was augmented by applying random transformations consisting of rotations at 90°, 180° and 270°, color saturation, contrast and brightness modifications at ranges between −20% and +20%. In this way, we provided a continuous stream of images for training, preventing overfitting problems at early training stages. This aspect is important for our model, as it provides a mechanism of adaptation for images taken under different lighting conditions and variation in camera settings. An example of such augmented images can be seen in Figure 4c, where synthetically generated transformations simulating such conditions have been applied. Batch normalization was avoided, as in most of the training images, several plant instances belonging to different classes appear at different positions, and the same is expected for the input images when the ensemble is operating at prediction stages. In addition to the random image augmentation, the stream of images was also modified by the repeat factor sampling (RFS), introduced in [71]. Mechanisms such as RFS are used to counteract the data imbalance present in training samples. Specifically, RSF consist of oversampling images that contain objects belonging to the less frequent classes by assigning to each class c a repeat factor

$$r_c = \max(1, \sqrt{\frac{t}{f_c}}), \tag{1}$$

where f_c is the fraction of images that contain at least one object belonging to class c. Considering that training images can contain several objects in different class categories, the repeat factor for an individual image is set to

$$r_i = \max_{c \in i}(r_c), \tag{2}$$

We used the parameter $t = 0.001$, as this resampling factor provides acceptable results for images containing multiple objects of different classes [71]. The distribution of class objects in our training and validation data is shown in Table 3. The GCP class was not used to train the Mask RCNN ensemble; however, it was detected using a grayscale-level histogram normalization and thresholding method for geolocation and photogrammetry purposes.

Table 3. Object distribution by class.

Class	HC1	HC2	HC3	HC4	HC5	GCP
Instances	550	454	653	258	165	60

2.5. Deep Learning Ensemble Model

The multi-stage architecture deep neural network Mask RCNN [42] is employed in this research as a feature extractor, classifier and instance detector. This architecture consists of an ensemble derived from the region-based convolutional neural network (RCNN) [72]; a simplified schematic diagram of Mask RCNN is shown in Figure 5.

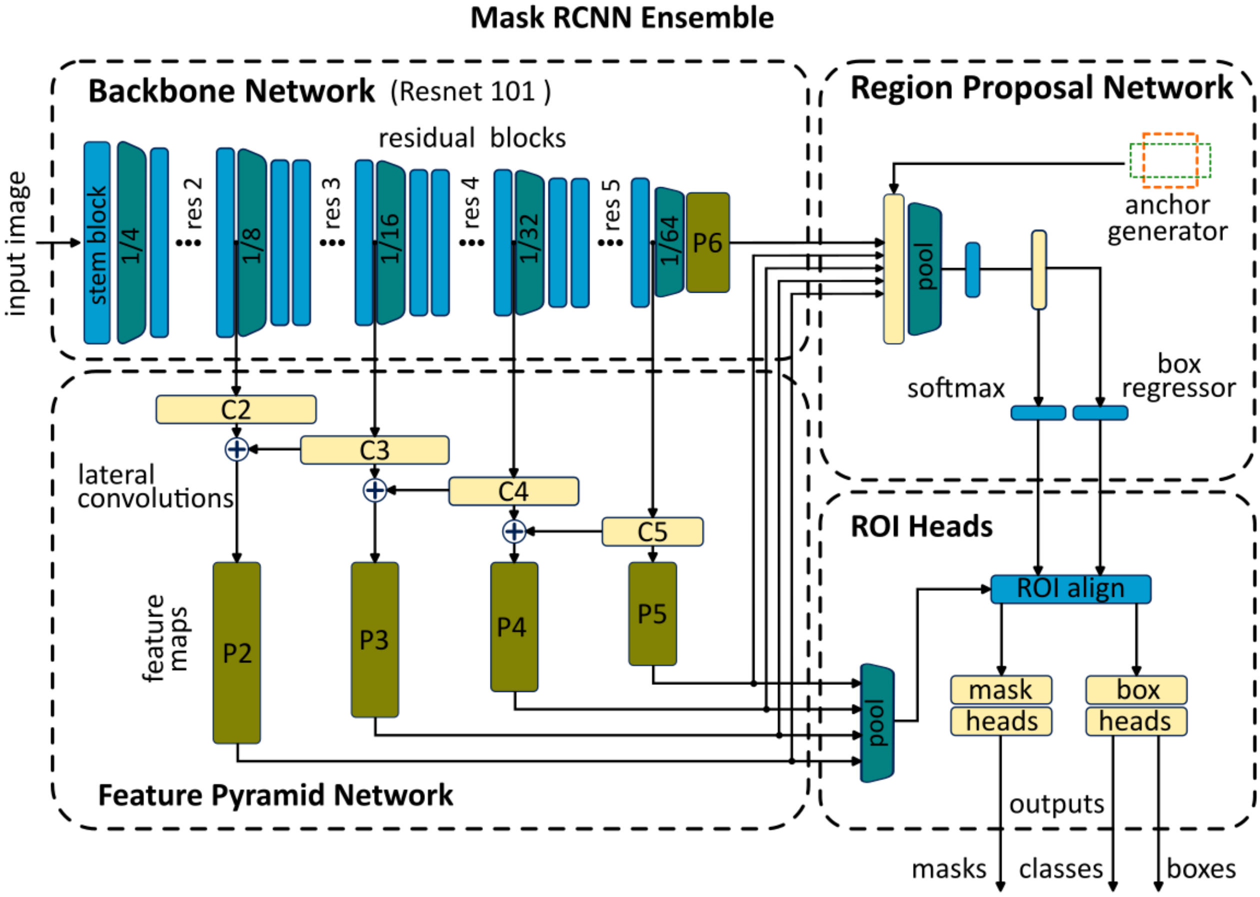


Figure 5. Basic schematics of a Mask RCNN ensemble.

In the Mask RCNN model, a backbone network is used as a feature extractor, Detectron2 allows several backbones to be used for this purpose. When implemented in conjunction with a feature pyramid network (FPN) [73], the outputs of the last residual blocks from the backbone network are linked to a series of $C2, \ldots, C5$, 1×1 convolutions that reduce the number of feature channels. These convolutions are concatenated with the previous features map that composes each stage of the map pyramid. Maps at every scale in the set $\{1/4, 1/8, 1/16, 1/32\}$ are identified with the $P2, \ldots, P5$ labels. A final feature map $P6$ scaled at $1/64$ is also taken at the end of the backbone network. The output of the stem layers of the residual network is ignored in favor of reducing memory footprint. Note that the FPN is not a necessary component for RCNNs, and the original Faster RCNN, the network on which Mask RCNN is based, does not implement it [41]. However, FPN improves detection and training speeds while reasonably maintaining accuracy [42]. The feature maps are fed as inputs to the region proposal network (RPN) [40,41] and to the region of interest (ROI) heads, which are primarily comprised of pooling and convolutional layers. The RPN component includes an anchor generator that produces predetermined locations and shapes for the initial proposals, returning the scores of each candidate region. The RPN output is a set of rectangular boxes with the respective scores as candidates for containing an object, along with class logits. Based on the feature maps, a box regressor and a softmax discriminator, the best candidate regions are given as inputs to the ROI heads module, whose main functions are to crop and pool regions taken from the proposals with higher objectness scores. These proposals have been previously relocated by an extra step called ROI alignment [42]. Final predictions for masks, locations and classes for each detected object are determined at this stage. For the particular instance segmentation problem investigated here, the backbone network used was a Resnet 101 model [72], which was previously initialized with pretrained weights and biases under three COCO epochs [74], as a way of having an initial state that included some connections related to semantic features involved in image classification tasks. This approach has been documented to speed up the training process by introducing some transfer learning operations [75]. The backbone choice was based on the fact that large residual networks are better for detecting fine grain features appearing in small objects [76]. When working with plant crop images, one problem that arises at the segmentation of contiguous plants is that in some cases, it is difficult to discern the boundaries of neighboring plant clusters. So, we tuned the model ensemble to learn the plant's morphology and phenology from the training examples to determine the borders of each cluster plant. The mechanism implemented to achieve this particular task is to give a set of fixed size and shapes of the predefined regions to the anchor generator at the RPN component of the ensemble. The set was adjusted to have areas at intervals from the average size of the foliar canopy of each of the plant health classes, plus and minus two standard deviations; aspect ratios for the anchors were also shaped in the same way. The loss function L for Mask RCNN models is composed of three elements:

$$L = L_{cls} + L_{box} + L_{mask} \tag{3}$$

where

$$L_{cls} = -log(p_u) \tag{4}$$

is the loss function for the label classification, with (p_u) representing the softmax operation for a ground-truth class labeled as u. For the case of pixel masks, L_{mask} is the average binary cross-entropy. L_{box} is the loss function that evaluates the precision of the location of the bounding boxes containing detected objects. For a class u with a ground-truth box v defined by the values $v = (v_x, v_y, v_w, v_h)$ in which v_x and v_y, are the coordinates of the upper left corner, and v_w, v_h correspond to its width and height, the loss of the regression for a predicted box $t^u = (t^u_x, t^u_y, t^u_w, t^u_h)$ uses the following loss function [40]:

$$L_{box} = \sum_{i \in \{x,y,w,h\}} S_{L_1}(t^u_i - v_i) \tag{5}$$

where

$$S_{L_1}(x) = \begin{cases} 0.5x^2 & |x| < 1 \\ |x| & |x| \end{cases}. \tag{6}$$

A custom script in the Python programming language was written to modify the default data loader of the Detectron2 libraries, with the intention of providing an input image with constant resolution to the first layer of the ensemble. To apply this script, it is necessary for the images to have units and values for the resolution tags at their exchangeable image file format (EXIF) [77] header. This represents no technical limitation, as most of the images gathered for the purpose of employing them in agriculture studies usually have them recorded [51]. Because most aerial imagery used in precision agriculture are in the format of georeferenced orthomosaics, the loading script feeds the input images in a mosaicking way, similar to the procedure described in [78]. To this end, orthomosaics are scanned by a sliding tile of fixed size. An overlap of size s is maintained between tiles that cover the orthomosaic, the computation of the value s and the locations of the tiles, which provides uniform cover for the entire area at a constant resolution for the input images performed by the custom data loader script. Regardless of the dimension of the analyzed image, the inputs passed to the MaskRCNN model are tensors with fixed dimensions of size $1 \times 512 \times 512 \times 3$. Repeated instances from partial detections occasionally appear at the tile borders. These instances are removed using a non maximum suppression (NMS) criteria with intersection over union (IOU) thresholds of 0.3 for intraclass objects and 0.7 for interclass objects. These values were based on the default anchor NMS values at the RPN network, as we wanted to preserve similar thresholding behavior for object filtering at the external process. A second criterion is applied to suppress duplicated and partial instances by setting limits on canopy area for each class based on the average values presented at

Figure 2b. The mechanism for scanning large orthomosaics by local detections on covering tiles explained above is depicted in Figure 6. Note that the phenotypic threshold filtering is only applied to detections of plants that do not appear completely on a tile, or that are duplicated because they are entirely located at the overlapping regions; all the other instance classifications are left the same as the output from the Mask RCNN.

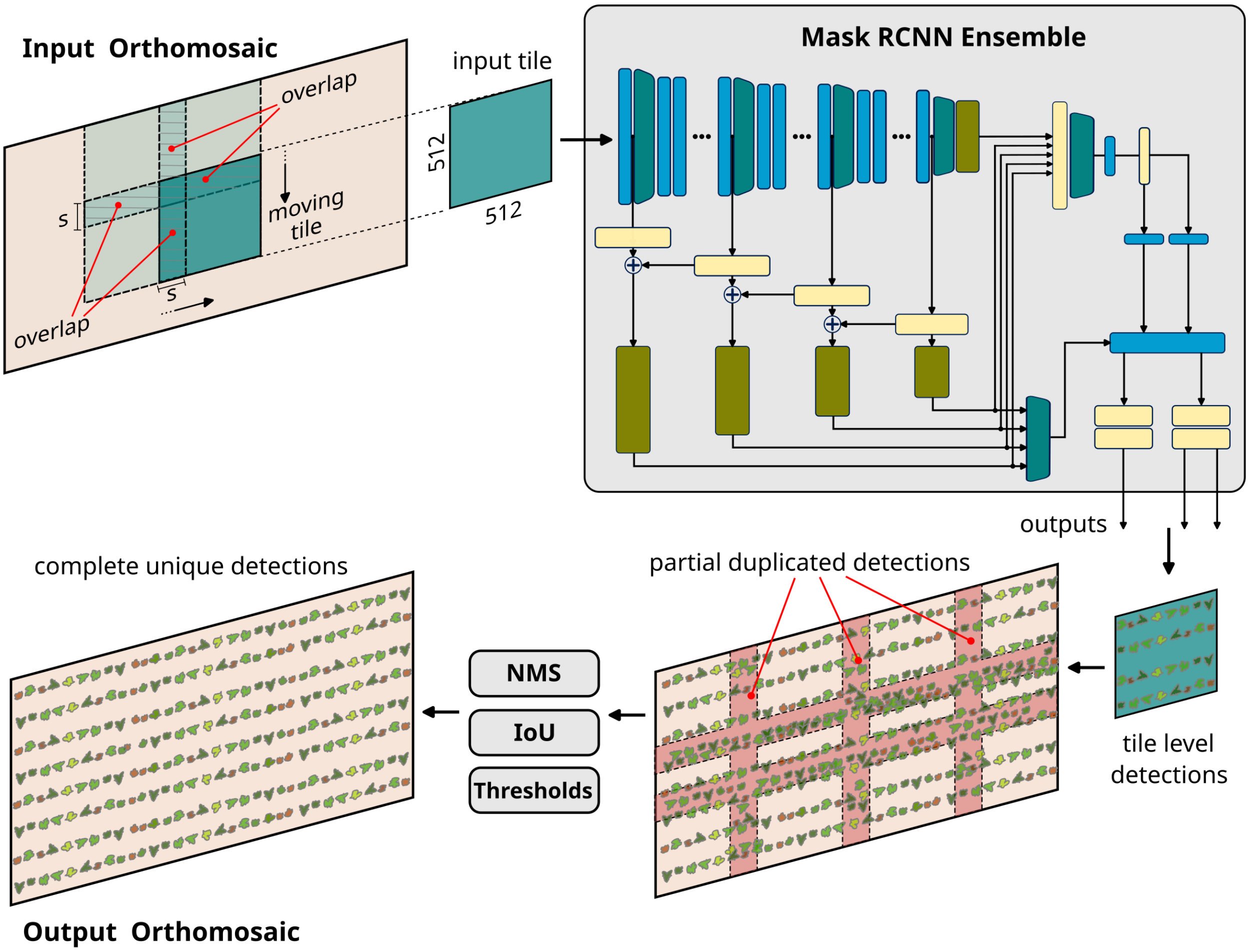


Figure 6. Basic schematics of a Mask RCNN ensemble.

The proposed processing pipeline allows the MaskRCNN to operate efficiently, as the NN model takes a fixed size input in the form of a multichannel tensor. Otherwise, previous image scaling would have been needed for images of arbitrary sizes. Note that if only image scaling was used to adjust the original input size, and the difference in scales is significant, as is the case of large orthomosaics, the recognition performance of Mask RCNN was heavily affected. Optimization of weights and biases for the training stage was executed using stochastic gradient descent (SGD) [79] with a momentum value of $m = 0.9$. A multi-step learning rate, starting at $lr = 0.001$ with discrete exponential adjustments with a factor $\gamma = 0.5$ was applied every 2500 epochs. Image batches consisted of 16 augmented images, with 512 ROIs being analyzed by the solver for each image.

3. Results

Using the equipment described in the previous section, it took 7.71 h to train the ensemble up to 25,000 epochs. The prediction time needed to process a large orthomosaic of 16,384 × 3584 pixels, covered with 360 overlapping tiles took, on average, 56 s; this time includes the overhead of removing partial detections at the overlapping zones, dividing the input orthomosaic and reconstructing it again with the predicted outputs. Note that the communication of tile tensors and prediction results between GPU and CPU memory also affects the total time taken to obtain a final output orthomosaic. The loss function behavior during the training epochs is shown in Figure 7. Loss function evolution through the training epochs of the Mask RCNN ensemble depicted in Figure 7 indicate that the optimization process follows a step decreasing trend for the total loss L metric at the first 15 K steps, from which changes were less marked up to 25 K iterations of the SGD algorithm. At this point, the training was stopped to avoid overfitting, as few improvements were recorded at this step.

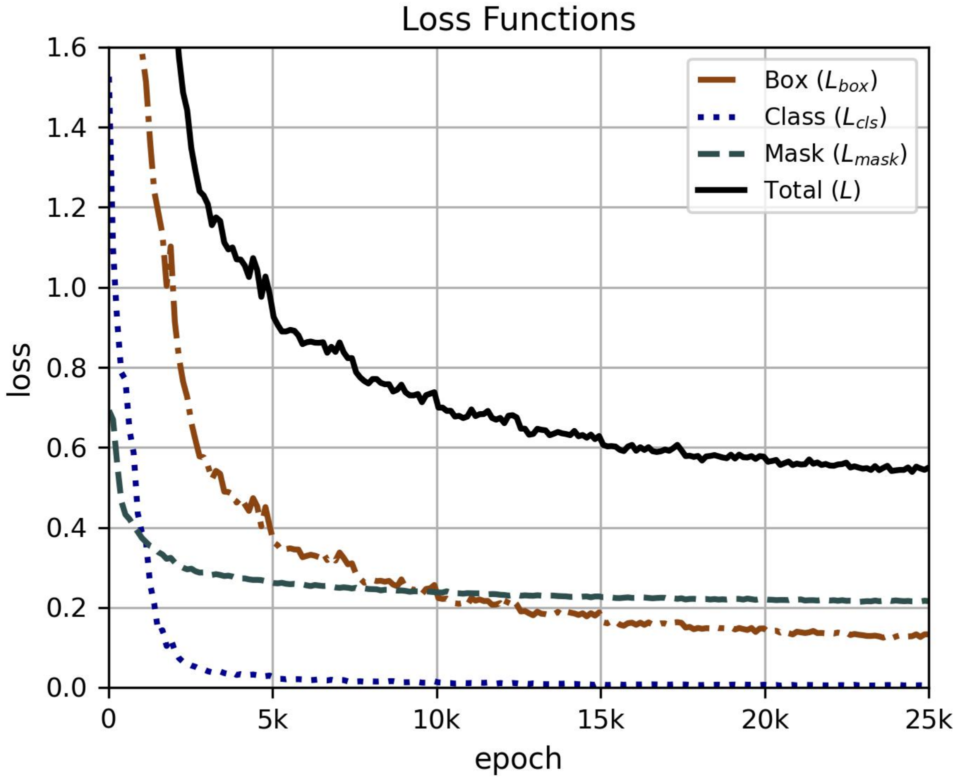


Figure 7. Loss function's behavior throughout training epochs.

To evaluate the performance of the classifier module of the Mask RCNN, we calculate the accuracy for all the classes, given by:

$$acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

where TP, TN, FP, FN are, respectively, the true and false positives and negatives using a set of IoU values in the interval $(0.5, 0.95)$ separated at 0.05 increments, sometimes denoted as IoU ∈ [0.50:0.05:0.95]. Accuracy levels throughout the epochs along with the portion of FP and FN are shown in Figure 8; the ability of the network to detect objects of interest is represented in this figure. The confusion matrix for the detection of each object in the validation set is shown in Figure 9.

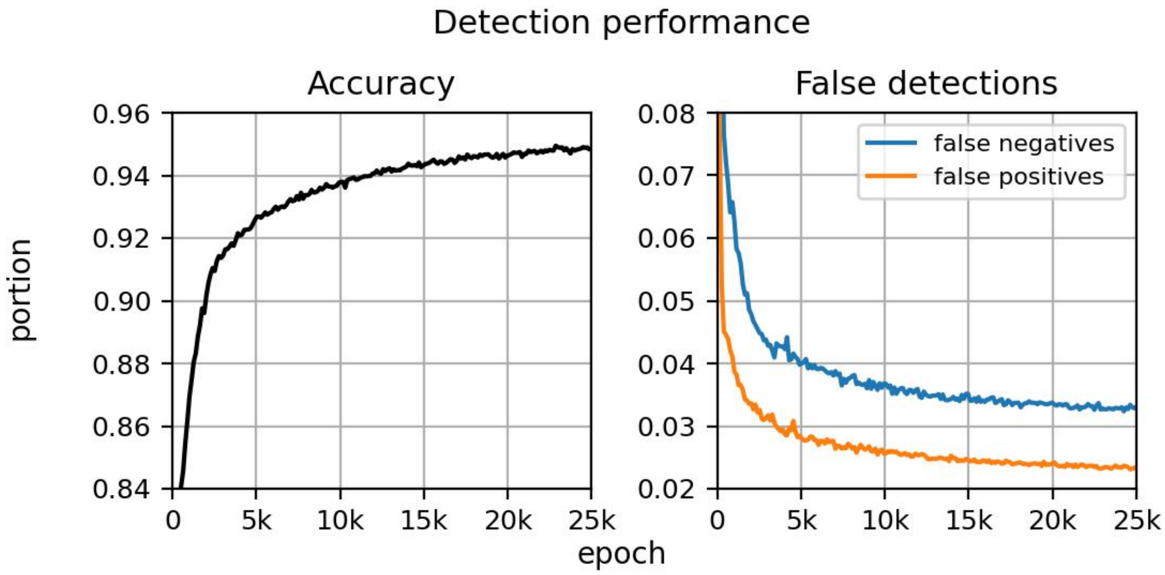
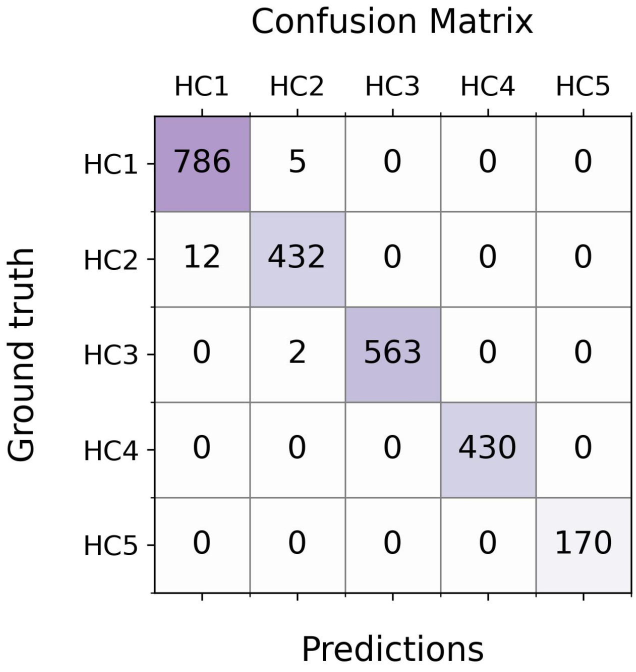


Figure 8. Accuracy and portion of incorrect detections during the NN training stage.



	HC1	HC2	HC3	HC4	HC5
HC1	786	5	0	0	0
HC2	12	432	0	0	0
HC3	0	2	563	0	0
HC4	0	0	0	430	0
HC5	0	0	0	0	170

Figure 9. Confusion matrix of object detection for the validation dataset.

To evaluate the precision of the masks and the locations of the instances, we employed the average precision (AP) metric as defined in the PASCAL visual object classes challenge [80], by taking for each validation image metrics of precision and recall, defined as:

$$\text{precision} = \frac{TP}{TP + FP} \tag{8}$$

$$\text{recall} = \frac{TP}{TP + FN} \tag{9}$$

Then, the measurements are sorted in a monotonically descending order, defining the AP as the area under precision-recall curve:

$$\text{AP} = \int_0^1 p(r)dr \tag{10}$$

with $p(r)$ representing the precision p as a function of the recall r. To avoid approximations introduced by the use of discrete data to estimate the AP values from the equation above, the precision for a given recall r is set to the maximum precision for every recall $r' \geq r$. The values of AP obtained for the predictions of the validation set for the boxes that locate each instance, and for the mask regions generated by the network are shown in Figure 10. Results are visualized for each class individually at this figure expressed as a percentage. Again, the AP values are based on the set of IoU $\in$ [0.50:0.05:0.95] thresholds to determine TP, TN, FP, and FN.

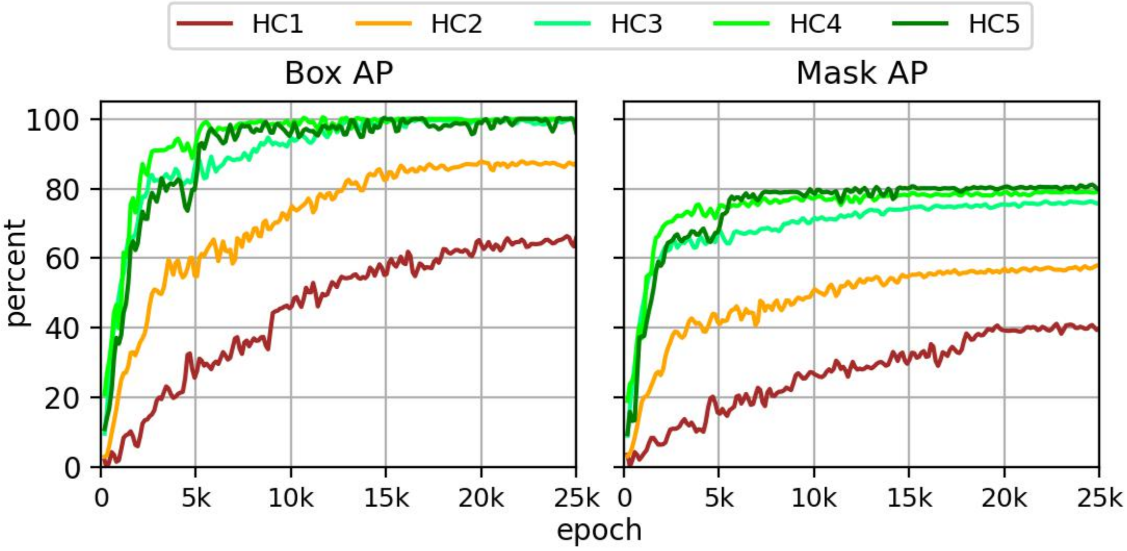


Figure 10. Average precision for box locations and mask areas delimitation for each health class at different training epochs.

Figure 10 shows that inferences are affected at different magnitudes for each $HC1, \ldots, HC5$, with $HC3$, $H4$, and $HC5$ APs being only slightly affected at the pixel labeling stage that generates object masks, while $HC1$ and $HC2$ APs reach significantly lower AP values. For the case of object localization estimated by bounding boxes, only the classes $HC1$ and $HC2$ are affected. This phenomenon has a reduced effect for overall classification accuracy, where a value of acc = 0.945 is reached at the end of the training epochs, as can be seen in Figure 8. Final AP values were smaller for low health level classes. The reason for this is that plants belonging to these groups have a smaller foliar area, and their shapes show, in many cases, branch-like structures; therefore, pixels of the image belonging to these objects are mixed with background pixels at a larger proportion than for the other classes.

To illustrate the tile-level outputs generated by the Mask RCNN ensemble, and the reference methods, we present the instance predictions, as well as the generated masks obtained for one of the tile samples at Figure 11a. Figure 11b shows the segmentation and pixel labeling performed by the RFLF classifier. Note that RFLF cannot distinguish between the background soil and the $HC1$ class. This might be due to the fact that plants with a lower health condition are mainly composed of dry matter, which, according to Figure 3, exhibits lower reflectance values at the NIR wavelengths than the other categories. RFLF is also unable to locate objects, as this algorithm was not designed for such function. One common strategy to delimit objects is watershed segmentation [81]; however, by using the same input data that were used to train the Mask RCNN model, the watershed segmentation does not match with the RFLF output, as can be seen in Figure 11c. Neither of these methods provide a satisfactory solution for the plant location problem. On the other hand, the LSMSS can detect and classify plant leaves and their health condition in a better way than RFLF, as shown in Figure 11d. However, object detection for LSMSS has to be performed statistically by grouping adjacent segments originated at a centroid generated by a KMeans spatial partition, and then the condition of the segments is averaged over a predefined radius. NDVI provides a nice segmentation for vegetation covered areas when thresholding is applied, but boundaries between plants cannot be determined in this way, as can be concluded by examining Figure 11f.

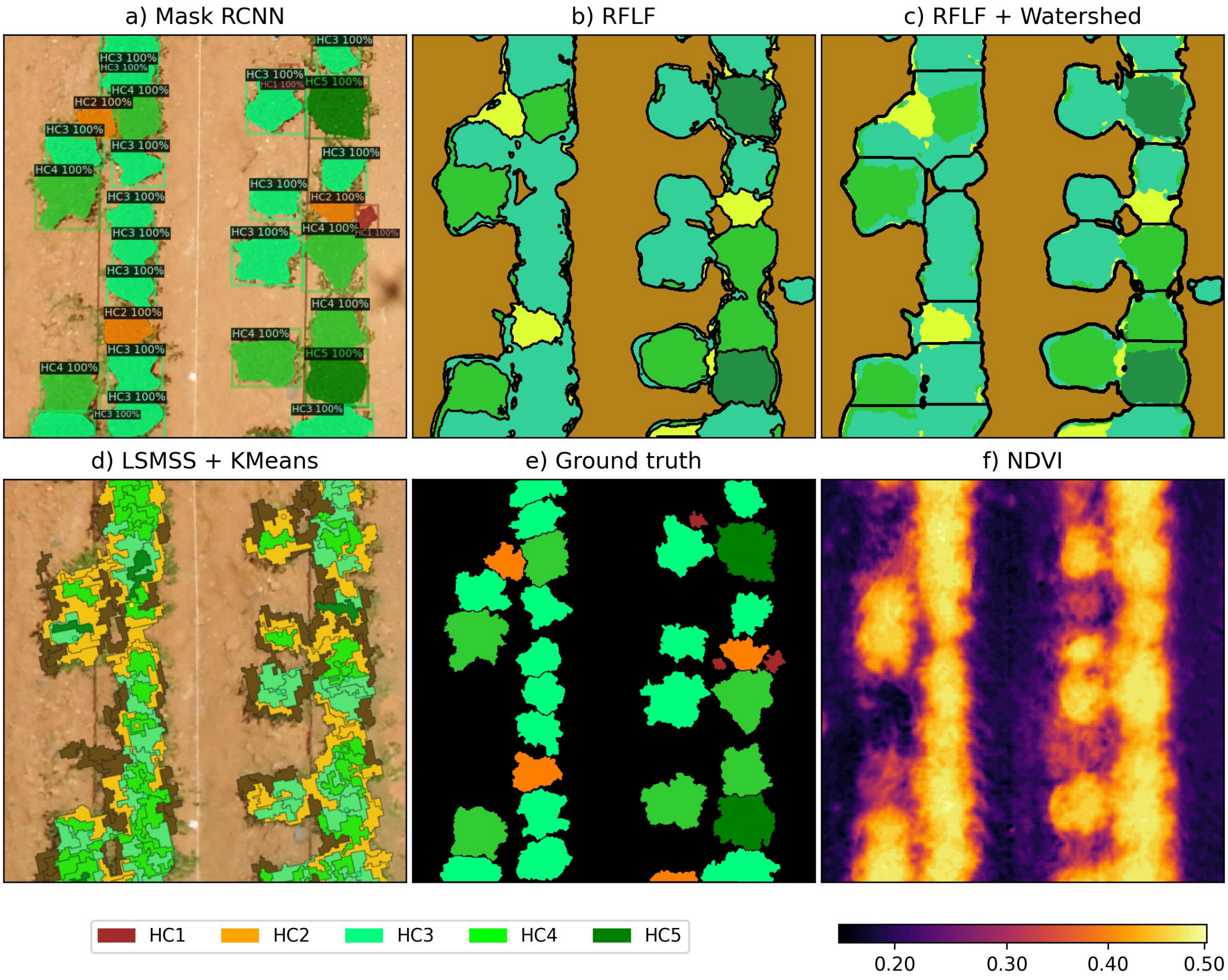


Figure 11. Tile level predictions for different segmentation procedures. Mask RCNN, and RFLF methods were trained with the same dataset samples. Classifiers for LSMSS and NDVI are non-supervised.

A larger section of the segmented orthomosaic obtained with Mask RCNN using the tiling procedure is shown in Figure 12. Figures 13 and 14 show, respectively, the RFLF, and LSMSS predictions over the same orthomosaic. Figure 15 presents the NDVI map obtained with the data collected with the multispectral camera. The five plant groups $HC1, \ldots, HC5$ can be accurately detected by the Mask RCNN procedure, as can be seen in Figures 11 and 12. The confusion matrix at Figure 9 shows that for the validation set, only 5 plants in the category $HC1$ were labeled as $HC2$ and 12 plants in the group $HC2$ were assigned to the $HC1$ type. All other plants were classified exactly in their corresponding categories. On the other hand, NDVI maps reveal values greater than 4.0 for healthy plants, values between 3.0 and 4.0 for unhealthy plants, and values lower than 3.0 for non vegetation objects, as shown in Figures 11 and 15. Thus, NDVI alone can only determine two plant health categories when applied to the dataset gathered for this research. Unlike the process described here, NDVI cannot be used to infer plant boundaries of overlapping canopies, or to locate instances of individual plants. This is because the indices extracted by NDVI do not consider phenotypic traits of the plants, which is a key factor used to determine the health state of vegetation samples, and to calculate their shape, location and extension, which was successfully performed in this work. The LSMSS can also distinguish all the $HC1, \ldots, HC5$ classes when is post-processed with the spatial KMeans algorithm that identifies centroids of Voronoi's regions [82] determining the object locations. The RFLF method cannot differentiate $HC1$ plants from background soil, and as it only performs semantic segmentation, it tends to assign a different class on the leaves at the plant's boundary disregarding the category assigned to the center mass of the objects.

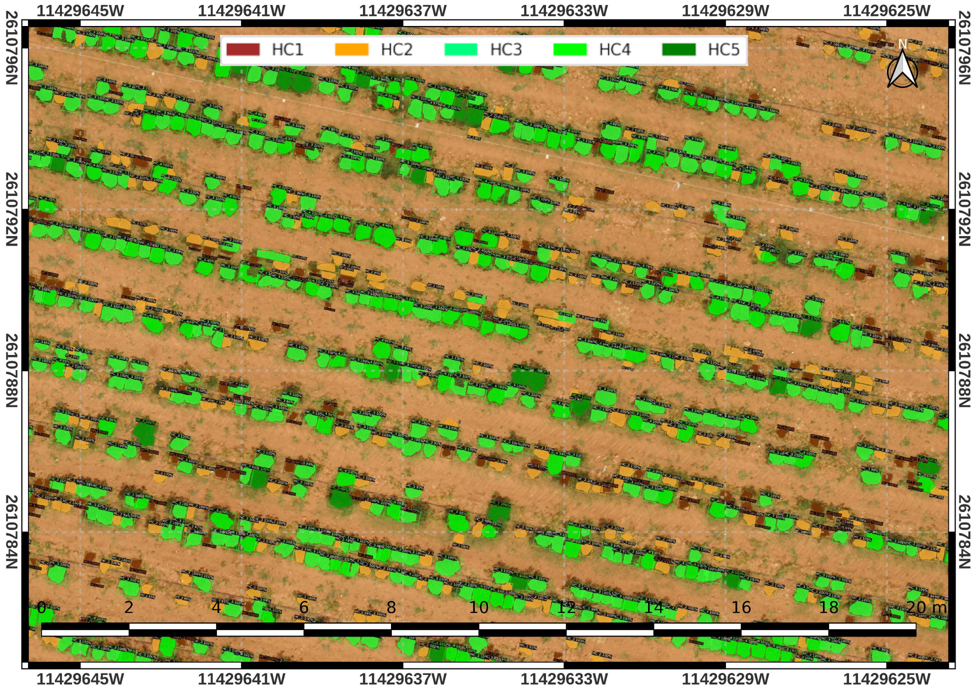


Figure 12. Instance detections and segmentation at orthomosaic level area.

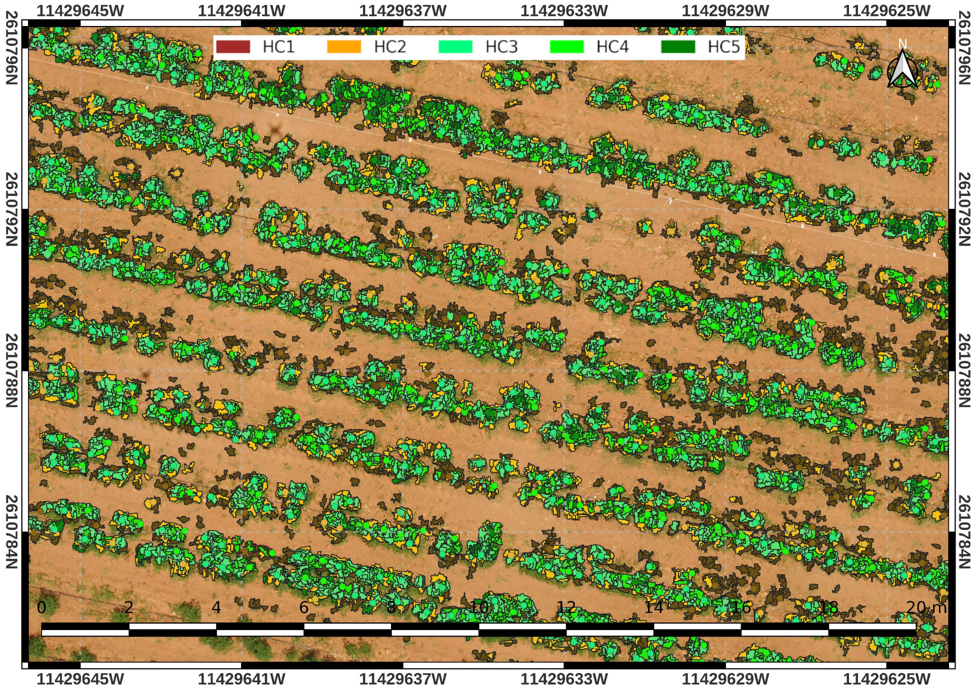


Figure 13. LSMSS with spatial KMeans plant detection and leaf segmentation at orthomosaic level area.

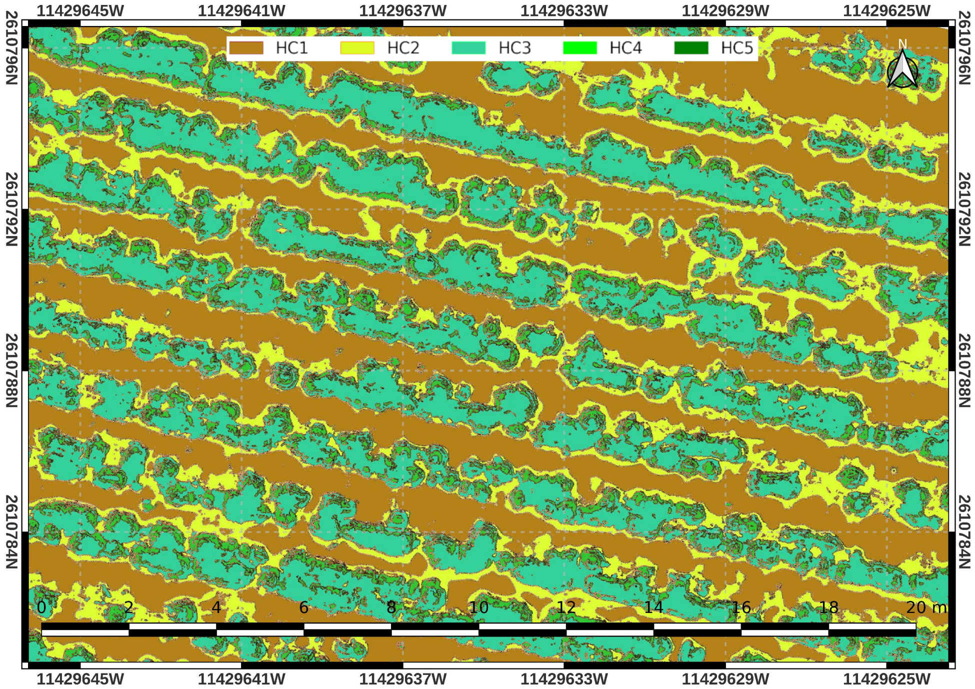


Figure 14. Segmentation with random forest classifier on local image features at orthomosaic level area.

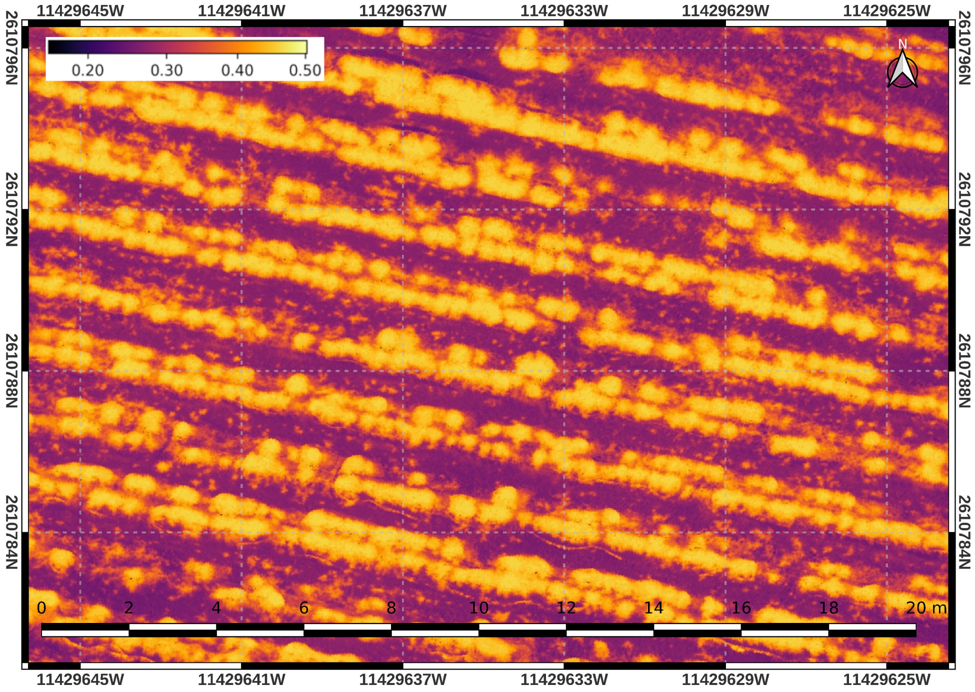


Figure 15. NDVI map at orthomosaic level area.

The computational complexity for the Mask RCNN ensemble performing predictions on an image depends on the individual layers that execute the forward propagation operation. Mask RCNNs are primarily composed of convolutional, fully connected and pooling layers, all of them known to have a computational complexity of $\mathcal{O}(n)$ [83], with n being the number of input pixels. The proposed tiling process also has computational complexity of $\mathcal{O}(n)$, since in the implementation of this paper, only matrix operations are applied to the orthomosaic to perform such a task. Table 4 compares the segmentation obtained by the Mask RCNN, RLFLF, LSMSS, and NDVI methods as described in this work. All were applied on the same input orthomosaic in an end-to-end fashion. The time data shown in this table were averaged sampled on 10 runs for each method; time figures correspond to wall time. Program sections that were executed on the CPU for all methods are parallelizable. The multiproccess programming for these cases was implemented by the native python multiprocessing module. 8 CPU processes, and an input orthomosaic of 16,384 ×3584 pixels were used in all cases.

Table 4. Comparison of segmentation capabilities performed by Mask RCNN, RFLF, LSMSS, and NDVI, executed on the same orthomosaic.

Feature	Mask RCNN (Tiled)	RFLF	LSMSS (KMeans)	NDVI
Detected classes	5	4	5	2
Execution device	GPU + CPU	CPU	CPU	CPU
Image channels	RGB	RGB	RGB, REG, NIR	RED, NIR
Training time (s)	27,756	2400	–	–
Inference time (s)	56	1184	4644	3.5

Table 5 shows the number of plant clusters of each class identified and segmented by the strategy described here, applied to the entire orthomosaic of the study area. Average scores presented correspond to the objectness obtained at the last layer of the network for all instances belonging to the same class. Canopy covered areas for each class and their percentage in relation to the total vegetation objects detected are also presented.

Table 5. Automated plant detection counts and per class and segmented canopy area estimates.

Class	Instances	Avg. Score	Portion (%)	Foliar Area (m^2)
$HC1$	749	0.9594	14.87	359.12
$HC2$	1396	0.9785	27.72	1110.76
$HC3$	2115	0.9839	42.00	2733.30
$HC4$	605	0.9714	12.07	1057.29
$HC5$	167	0.9722	3.31	380.31
All	5035	0.9731	100.00	5640.81

4. Discussion

The plant health classes defined in Section 2 are consistent with SAD maps for leaf samples collected in-field, and they are in accordance with the average spectral signatures for the reflectance of each category. In Figure 3, the differences in the average reflectance signatures of each health category can be distinguished even at the visible bands GRE, and RED. As expected, the signatures are much more easily differentiated considering the NIR band, whose comparison with the RED band is quantitatively expressed by NDVI. The behavior of signatures at the REG band shows that the spectrum of healthy plants have a more accentuated slope than unhealthy plants in the same region; therefore, the differences in healthy and unhealthy plants are also exposed in this spectral transition region.

Many image segmentation algorithms based on classic methods such as threshold, dilation and eroding perform only for semantic segmentation, and in many cases, algorithm parameters and the selection of the features employed need to be tuned for specific images [84]. The implementation of the RFLF method, which is widely used to analyze images obtained from multispectral sensors, is one of these examples. We used it here for comparison with the proposed mechanism. The main disadvantage of the RFLF approach is that it can only perform semantic segmentation, and even when the results are post-processed with a watershed detector, as described in the previous section, the location and shapes of the plants are not accurately matched by these techniques with plants' morphology, as shown in Figure 11. On the other hand, the approach introduced here allows us to detect individual instances of *C. annuum* plants under different contexts, given the image augmented images fed to the network in the training stage.

Techniques such as large-scale mean shift segmentation (LSMSS) [47] and object-based image analysis (OBIA) [85] are among the algorithms that can also perform instance segmentation on images using manually engineered feature extraction. Specifically, instance segmentation using OBIA implemented by the spatial KMeans algorithm was compared with the proposed pipeline. Although training time for LSMSS + KMeans does not need prior training, as it is based on unsupervised methods, its prediction times are rather long, according to Table 4. In the present work, using the Mask RCNN in a tiled fashion, the processing of a 58.7 Mpx orthomosaic is performed in just 56 s, which represents an improvement of two orders of magnitude on the execution speed at analyzing large orthomosaics.

The input for the Mask RCNN classifier used here only needs to be in RGB image format, which is standard for many commercial UAVs used for crop monitoring. Among the advantages of working RGB images, modern RGB cameras designed for UAVs are cheaper, have larger spatial resolution and are lighter than their multispectral counterparts. Spectral cameras enable us to establish several vegetation indices with precision by applying simple operations. Nonetheless, in this study, the lack of multispectral features is compensated by the phenotype traits learned by the Mask RCNN ensemble for the accurate estimation of vegetation health, providing additional instance segmentation capabilities. These properties are very useful at crop-monitoring-related tasks, as presented in Table 5. The introduced pipeline is capable of counting the total amount of plants in a crop, detecting the health state of each plant, estimating the foliar area covered by the plants of each category and locating the pathological cases in an accurate and fast georeferenced way.

5. Conclusions

In this work, we developed a sequence of steps that allowed efficient processing of aerial imagery for the task of monitoring *C. Annum* crops. By training a Mask RCNN deep learning model with the proper annotated imagery corresponding to vegetation health classes, and polygonal boundaries of individual plants, it was possible to provide high-accuracy automated detection of up to five health classes of vegetation, and to determine their locations and shapes with acceptable precision. The classes defined for training were based on spectral signatures and phenotypic features of the vegetation under study. The model was fed with fixed tiled inputs representing the partition of larger images. In this way, the Mask RCNN performed reasonably well, without showing scaling issues when dealing with large orthomosaics representing vegetation fields. Comparison with methods such as RFLF and LSMSS, shows advantages of the proposed pipeline of using a Mask RCNN ensemble in a tiled way. Such improvements arise from the ability to perform instance segmentation on large orthomosaics with low execution times, as once the model was trained, the execution time taken to predict the classes, locations, and plant shapes was less than one minute when examining an orthomosaic composed of multiple images. For the goal of health state determination of *C. Annum* crops, the inferences obtained by the model proposed here using RGB imagery give more detailed and informative results than the standard NDVI method using multispectral instruments, and also outperforms the models using a set of predefined features, such as RFLF and LSMSS.

The methodology presented here can be easily adapted to other crops, which can be implemented in future works. Therefore, it represents a viable alternative for automated crop monitoring using RGB airborne images.

Author Contributions: All authors have contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by project “Cátedras CONACYT” no. 2643, 621, 843, 2210.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AP	Average precision.
DL	Deep learning.
EXIF	Exchangeable image file format.
FPN	Feature pyramid network.
GIS	Geographic information system.
GRE	Green.
I2C	Inter integrated circuit.
IoU	Intersection over union.
LSMSS	Large scale mean shift segmentation.
NDVI	Normalized differential vegetation index.
NIR	Near infrared.
NMS	Non maximum suppression.
NN	Neural network.
OBIA	Object-based image analysis.
OTG	On the go.
RCNN	Region-based neural network.
RED	Red.
REG	Red edge.
ROI	Region of interest.
RPN	Region proposal network.
SAD	Standard area diagrams.
SGD	Stochastic gradient descent.
USB	Universal serial bus.

References

1. Moreno-Pérez, E.D.C.; Avendaño-Arrazate, C.H.; Mora-Aguilar, R.; Cadena-Iñiguez, J.; Aguilar-Rincón, V.H.; Aguirre-Medina, J.F. Diversidad morfológica en colectas de chile guajillo (*Capsicum annuum* L.) del centro-norte de México. *Rev. Chapingo. Ser. Hortic.* **2011**, *17*, 23–30. [CrossRef]
2. Del Ponte, E.M.; Pethybridge, S.J.; Bock, C.H.; Michereff, S.J.; Machado, F.J.; Spolti, P. Standard area diagrams for aiding severity estimation: Scientometrics, pathosystems, and methodological trends in the last 25 years. *Phytopathology* **2017**, *107*, 1161–1174. [CrossRef]
3. Bock, C.H.; Chiang, K.S.; Del Ponte, E.M. Plant disease severity estimated visually: A century of research, best practices, and opportunities for improving methods and practices to maximize accuracy. *Trop. Plant Pathol.* **2022**, *47*, 25–42. [CrossRef]
4. Ahmadi, P.; Mansor, S.; Farjad, B.; Ghaderpour, E. Unmanned Aerial Vehicle (UAV)-based remote sensing for early-stage detection of Ganoderma. *Remote Sens.* **2022**, *14*, 1239. [CrossRef]
5. Dunford, R.; Michel, K.; Gagnage, M.; Piégay, H.; Trémelo, M.L. Potential and constraints of Unmanned Aerial Vehicle technology for the characterization of Mediterranean riparian forest. *Int. J. Remote Sens.* **2009**, *30*, 4915–4935. [CrossRef]
6. Maimaitijiang, M.; Ghulam, A.; Sidike, P.; Hartling, S.; Maimaitiyiming, M.; Peterson, K.; Shavers, E.; Fishman, J.; Peterson, J.; Kadam, S.; et al. Unmanned Aerial System (UAS)-based phenotyping of soybean using multi-sensor data fusion and extreme learning machine. *ISPRS J. Photogramm. Remote Sens.* **2017**, *134*, 43–58. [CrossRef]
7. Tsouros, D.C.; Bibi, S.; Sarigiannidis, P.G. A review on UAV-based applications for precision agriculture. *Information* **2019**, *10*, 349. [CrossRef]
8. Ammad Uddin, M.; Mansour, A.; Le Jeune, D.; Ayaz, M.; Aggoune, E.H.M. UAV-assisted dynamic clustering of wireless sensor networks for crop health monitoring. *Sensors* **2018**, *18*, 555. [CrossRef]
9. Singh, N.; Gupta, N. Decision-Making in Integrated Pest Management and Bayesian Network. *Int. J. Comput. Sci. Inf. Technol.* **2017**, *9*, 31–37. [CrossRef]
10. Hamada, M.A.; Kanat, Y.; Abiche, A.E. Multi-spectral image segmentation based on the K-means clustering. *Int. J. Innov. Technol. Explor. Eng.* **2019**, *9*, 1016–1019. [CrossRef]
11. Das, S.; Christopher, J.; Apan, A.; Choudhury, M.R.; Chapman, S.; Menzies, N.W.; Dang, Y.P. UAV-Thermal imaging and agglomerative hierarchical clustering techniques to evaluate and rank physiological performance of wheat genotypes on sodic soil. *ISPRS J. Photogramm. Remote Sens.* **2021**, *173*, 221–237. [CrossRef]
12. Rajeswari, S.; Suthendran, K. C5.0: Advanced Decision Tree (ADT) classification model for agricultural data analysis on cloud. *Comput. Electron. Agric.* **2019**, *156*, 530–539. [CrossRef]

13. Tariq, A.; Yan, J.; Gagnon, A.S.; Riaz Khan, M.; Mumtaz, F. Mapping of cropland, cropping patterns and crop types by combining optical remote sensing images with decision tree classifier and random forest. *Geo-Spat. Inf. Sci.* **2022**, 1–19. [CrossRef]
14. Muliady, M.; Lim, T.S.; Koo, V.C.; Patra, S. Classification of rice plant nitrogen nutrient status using k-nearest neighbors (k-NN) with light intensity data. *Indones. J. Electr. Eng. Comput. Sci.* **2021**, *22*, 179–186. [CrossRef]
15. Dharmaraj, V.; Vijayanand, C. Artificial intelligence (AI) in agriculture. *Int. J. Curr. Microbiol. Appl. Sci.* **2018**, *7*, 2122–2128. [CrossRef]
16. Kujawa, S.; Niedbała, G. Artificial neural networks in agriculture. *Agriculture* **2021**, *11*, 497. [CrossRef]
17. Song, Y.; Teng, G.; Yuan, Y.; Liu, T.; Sun, Z. Assessment of wheat chlorophyll content by the multiple linear regression of leaf image features. *Inf. Process. Agric.* **2021**, *8*, 232–243. [CrossRef]
18. Sahoo, R.N. Sensor-Based Monitoring of Soil and Crop Health for Enhancing Input Use Efficiency. In *Food, Energy, and Water Nexus*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 129–147.
19. Banerjee, I.; Madhumathy, P. IoT Based Agricultural Business Model for Estimating Crop Health Management to Reduce Farmer Distress Using SVM and Machine Learning. In *Internet of Things and Analytics for Agriculture*; Springer: Berlin/Heidelberg, Germany, 2022; Volume 3, pp. 165–183.
20. Kamilaris, A.; Prenafeta-Boldú, F.X. Deep learning in agriculture: A survey. *Comput. Electron. Agric.* **2018**, *147*, 70–90. [CrossRef]
21. Türkoğlu, M.; Hanbay, D. Plant disease and pest detection using deep learning-based features. *Turk. J. Electr. Eng. Comput. Sci.* **2019**, *27*, 1636–1651. [CrossRef]
22. Chen, C.J.; Huang, Y.Y.; Li, Y.S.; Chang, C.Y.; Huang, Y.M. An AIoT based smart agricultural system for pests detection. *IEEE Access* **2020**, *8*, 180750–180761. [CrossRef]
23. 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]
24. Chen, C.J.; Huang, Y.Y.; Li, Y.S.; Chen, Y.C.; Chang, C.Y.; Huang, Y.M. Identification of fruit tree pests with deep learning on embedded drone to achieve accurate pesticide spraying. *IEEE Access* **2021**, *9*, 21986–21997. [CrossRef]
25. Feng, J.; Sun, Y.; Zhang, K.; Zhao, Y.; Ren, Y.; Chen, Y.; Zhuang, H.; Chen, S. Autonomous Detection of Spodoptera frugiperda by Feeding Symptoms Directly from UAV RGB Imagery. *Appl. Sci.* **2022**, *12*, 2592. [CrossRef]
26. Bah, M.D.; Hafiane, A.; Canals, R. Deep learning with unsupervised data labeling for weed detection in line crops in UAV images. *Remote Sens.* **2018**, *10*, 1690. [CrossRef]
27. Etienne, A.; Saraswat, D. Machine learning approaches to automate weed detection by UAV based sensors. Autonomous Air and Ground Sensing Systems for Agricultural Optimization and Phenotyping IV. *Int. Soc. Opt. Photonics* **2019**, *11008*, 110080R.
28. Veeranampalayam Sivakumar, A.N.; Li, J.; Scott, S.; Psota, E.; J Jhala, A.; Luck, J.D.; Shi, Y. Comparison of object detection and patch-based classification deep learning models on mid-to late-season weed detection in UAV imagery. *Remote Sens.* **2020**, *12*, 2136. [CrossRef]
29. Khan, S.; Tufail, M.; Khan, M.T.; Khan, Z.A.; Anwar, S. Deep learning-based identification system of weeds and crops in strawberry and pea fields for a precision agriculture sprayer. *Precis. Agric.* **2021**, *22*, 1711–1727. [CrossRef]
30. Beeharry, Y.; Bassoo, V. Drone-Based Weed Detection Architectures Using Deep Learning Algorithms and Real-Time Analytics. In *Computer Vision and Machine Learning in Agriculture*; Springer: Berlin/Heidelberg, Germany, 2022; Volume 2, pp. 15–33.
31. Ge, X.; Wang, J.; Ding, J.; Cao, X.; Zhang, Z.; Liu, J.; Li, X. Combining UAV-based hyperspectral imagery and machine learning algorithms for soil moisture content monitoring. *PeerJ* **2019**, *7*, e6926. [CrossRef]
32. Su, J.; Coombes, M.; Liu, C.; Zhu, Y.; Song, X.; Fang, S.; Guo, L.; Chen, W.H. Machine learning-based crop drought mapping system by UAV remote sensing RGB imagery. *Unmanned Syst.* **2020**, *8*, 71–83. [CrossRef]
33. Zhou, Z.; Majeed, Y.; Naranjo, G.D.; Gambacorta, E.M. Assessment for crop water stress with infrared thermal imagery in precision agriculture: A review and future prospects for deep learning applications. *Comput. Electron. Agric.* **2021**, *182*, 106019. [CrossRef]
34. Cheng, M.; Jiao, X.; Liu, Y.; Shao, M.; Yu, X.; Bai, Y.; Wang, Z.; Wang, S.; Tuohuti, N.; Liu, S.; et al. Estimation of soil moisture content under high maize canopy coverage from UAV multimodal data and machine learning. *Agric. Water Manag.* **2022**, *264*, 107530. [CrossRef]
35. Mithra, S.; Nagamalleswari, T. An analysis of deep learning models for dry land farming applications. *Appl. Geomat.* **2022**, 1–7. [CrossRef]
36. Reedha, R.; Dericquebourg, E.; Canals, R.; Hafiane, A. Transformer Neural Network for Weed and Crop Classification of High Resolution UAV Images. *Remote Sens.* **2022**, *14*, 592. [CrossRef]
37. Zhang, X.; Han, L.; Dong, Y.; Shi, Y.; Huang, W.; Han, L.; González-Moreno, P.; Ma, H.; Ye, H.; Sobeih, T. A deep learning-based approach for automated yellow rust disease detection from high-resolution hyperspectral UAV images. *Remote Sens.* **2019**, *11*, 1554. [CrossRef]
38. Tetila, E.C.; Machado, B.B.; Menezes, G.K.; Oliveira, A.D.S.; Alvarez, M.; Amorim, W.P.; Belete, N.A.D.S.; Da Silva, G.G.; Pistori, H. Automatic recognition of soybean leaf diseases using UAV images and deep convolutional neural networks. *IEEE Geosci. Remote Sens. Lett.* **2019**, *17*, 903–907. [CrossRef]
39. Hu, G.; Wang, T.; Wan, M.; Bao, W.; Zeng, W. UAV remote sensing monitoring of pine forest diseases based on improved Mask R-CNN. *Int. J. Remote Sens.* **2022**, *43*, 1274–1305. [CrossRef]

40. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
41. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [CrossRef] [PubMed]
42. 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.
43. Wang, H.; Mou, Q.; Yue, Y.; Zhao, H. Research on detection technology of various fruit disease spots based on mask R-CNN. In Proceedings of the 2020 IEEE International Conference on Mechatronics and Automation (ICMA), Beijing, China, 13–16 October 2020; pp. 1083–1087.
44. Afzaal, U.; Bhattarai, B.; Pandeya, Y.R.; Lee, J. An Instance Segmentation Model for Strawberry Diseases Based on Mask R-CNN. *Sensors* **2021**, *21*, 6565. [CrossRef] [PubMed]
45. Storey, G.; Meng, Q.; Li, B. Leaf Disease Segmentation and Detection in Apple Orchards for Precise Smart Spraying in Sustainable Agriculture. *Sustainability* **2022**, *14*, 1458. [CrossRef]
46. Farzadpour, F.; Church, P.; Chen, X. Modeling and optimizing the coverage performance of the lidar sensor network. In Proceedings of the 2018 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Auckland, New Zealand, 9–12 July 2018; pp. 504–509.
47. Michel, J.; Youssefi, D.; Grizonnet, M. Stable mean-shift algorithm and its application to the segmentation of arbitrarily large remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2014**, *53*, 952–964. [CrossRef]
48. Sosa-Herrera, J.A.; Vallejo-Pérez, M.R.; Álvarez-Jarquín, N.; Cid-García, N.M.; López-Araujo, D.J. Geographic object-based analysis of airborne multispectral images for health assessment of *Capsicum annuum* L. crops. *Sensors* **2019**, *19*, 4817. [CrossRef]
49. Kwenda, C.; Gwetu, M.; Dombeu, J.V.F. Machine Learning Methods for Forest Image Analysis and Classification: A Survey of the State of the Art. *IEEE Access* **2022**, *10*, 45290–45316. [CrossRef]
50. Bahrami, M.; Mobasheri, M.R. Plant species determination by coding leaf reflectance spectrum and its derivatives. *Eur. J. Remote Sens.* **2020**, *53*, 258–273. [CrossRef]
51. Sishodia, R.P.; Ray, R.L.; Singh, S.K. Applications of remote sensing in precision agriculture: A review. *Remote Sens.* **2020**, *12*, 3136. [CrossRef]
52. Huang, S.; Tang, L.; Hupy, J.P.; Wang, Y.; Shao, G. A commentary review on the use of normalized difference vegetation index (NDVI) in the era of popular remote sensing. *J. For. Res.* **2021**, *32*, 1–6. [CrossRef]
53. Peel, M.C.; Finlayson, B.L.; McMahon, T.A. Updated world map of the Köppen-Geiger climate classification. *Hydrol. Earth Syst. Sci.* **2007**, *11*, 1633–1644. [CrossRef]
54. Medina-García, G.; Ruiz Corral, J.A. *Estadísticas Climatológicas Básicas del Estado de Zacatecas (Periodo 1961–2003)*; Instituto Nacional de Investigaciones Forestales, Agrícolas y Pecuarias: Mexico City, Mexico, 2004.
55. USB.org. On-The-Go and Embedded Host Supplement to the USB Revision 3.0 Specification; Rev. 3.0.; 2012. Available online: https://www.usb.org/sites/default/files/documents/usb_otg_and_eh_3-0_release_1_1_10may2012.pdf (accessed on 1 July 2022).
56. NXP Semiconductors. I2C Bus Specification and User Manual; Rev. 7.0.; 2021. Available online: https://www.nxp.com/docs/en/user-guide/UM10204.pdf (accessed on 1 July 2022).
57. Hamamatsu Photonics. *C12880MA Final Inspection Sheet*; Hamamatsu Photonics: Shizuoka, Japan, 2018.
58. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*; Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2019; pp. 8024–8035.
59. Wu, Y.; Kirillov, A.; Massa, F.; Lo, W.Y.; Girshick, R. Detectron2. 2019. Available online: https://github.com/facebookresearch/detectron2 (accessed on 1 January 2022).
60. Girshick, R.; Radosavovic, I.; Gkioxari, G.; Dollár, P.; He, K. Detectron. 2018. Available online: https://github.com/facebookresearch/detectron (accessed on 1 January 2022).
61. Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; Darrell, T. Caffe: Convolutional Architecture for Fast Feature Embedding. *arXiv* **2014**, arXiv:1408.5093.
62. QGIS Development Team. *QGIS Geographic Information System*; Open Source Geospatial Foundation: Beaverton, OR, USA, 2009.
63. Snyder, J.P. The space oblique Mercator projection. *Photogramm. Eng. Remote Sens.* **1978**, *44*, 140.
64. Sekachev, B.; Manovich, N.; Zhiltsov, M.; Zhavoronkov, A.; Kalinin, D.; Hoff, B.; TOsmanov.; Kruchinin, D.; Zankevich, A.; Sidnev, D.; et al. opencv/cvat: v1.1.0, 2020. Available online: https://zenodo.org/record/4009388#.YzJst3ZByUk (accessed on 1 January 2022).
65. Sotak, G.E., Jr.; Boyer, K.L. The Laplacian-of-Gaussian kernel: A formal analysis and design procedure for fast, accurate convolution and full-frame output. *Comput. Vis. Graph. Image Process.* **1989**, *48*, 147–189. [CrossRef]
66. Mikolajczyk, K.; Tuytelaars, T.; Schmid, C.; Zisserman, A.; Matas, J.; Schaffalitzky, F.; Kadir, T.; Gool, L.V. A comparison of affine region detectors. *Int. J. Comput. Vis.* **2005**, *65*, 43–72. [CrossRef]
67. Gedraite, E.S.; Hadad, M. Investigation on the effect of a Gaussian Blur in image filtering and segmentation. In Proceedings of the ELMAR-2011, Zadar, Croatia, 14–16 September 2011; pp. 393–396.

68. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
69. Van der Walt, S.; Schönberger, J.L.; Nunez-Iglesias, J.; Boulogne, F.; Warner, J.D.; Yager, N.; Gouillart, E.; Yu, T. scikit-image: Image processing in Python. *PeerJ* **2014**, *2*, e453. [CrossRef] [PubMed]
70. Grizonnet, M.; Michel, J.; Poughon, V.; Inglada, J.; Savinaud, M.; Cresson, R. Orfeo ToolBox: Open source processing of remote sensing images. *Open Geospat. Data, Softw. Stand.* **2017**, *2*, 15. [CrossRef]
71. Gupta, A.; Dollar, P.; Girshick, R. LVIS: A dataset for large vocabulary instance segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5356–5364.
72. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
73. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
74. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
75. Minaee, S.; Boykov, Y.Y.; Porikli, F.; Plaza, A.J.; Kehtarnavaz, N.; Terzopoulos, D. Image segmentation using deep learning: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 3523–3542. [CrossRef] [PubMed]
76. Fu, C.Y.; Liu, W.; Ranga, A.; Tyagi, A.; Berg, A.C. Dssd: Deconvolutional single shot detector. *arXiv* **2017**, arXiv:1701.06659.
77. Camera & Imaging Products Association. *Exchangeable Image File Format for Digital still Cameras: Exif Version 2.3*; Camera & Imaging Products Association: Tokyo, Japan, 2012.
78. Carvalho, O.L.F.d.; de Carvalho Junior, O.A.; Albuquerque, A.O.d.; Bem, P.P.d.; Silva, C.R.; Ferreira, P.H.G.; Moura, R.d.S.d.; Gomes, R.A.T.; Guimaraes, R.F.; Borges, D.L. Instance segmentation for large, multi-channel remote sensing imagery using mask-RCNN and a mosaicking approach. *Remote Sens.* **2020**, *13*, 39. [CrossRef]
79. Amari, S.I. Backpropagation and stochastic gradient descent method. *Neurocomputing* **1993**, *5*, 185–196. [CrossRef]
80. Everingham, M.; Eslami, S.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes challenge: A retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136. [CrossRef]
81. Kornilov, A.S.; Safonov, I.V. An overview of watershed algorithm implementations in open source libraries. *J. Imaging* **2018**, *4*, 123. [CrossRef]
82. Aurenhammer, F.; Klein, R. Voronoi Diagrams. *Handb. Comput. Geom.* **2000**, *5*, 201–290.
83. Dumoulin, V.; Visin, F. A guide to convolution arithmetic for deep learning. *arXiv* **2016**, arXiv:1603.07285.
84. Kang, W.X.; Yang, Q.Q.; Liang, R.P. The comparative research on image segmentation algorithms. In Proceedings of the 2009 First International Workshop on Education Technology and Computer Science, Wuhan, China, 7–8 March 2009; Volume 2, pp. 703–707.
85. Blaschke, T. Object based image analysis for remote sensing. *ISPRS J. Photogramm. Remote Sens.* **2010**, *65*, 2–16. [CrossRef]