



Article

UAV-Based Computer Vision System for Orchard Apple Tree Detection and Health Assessment

Hela Jemaa ^{1,2,*}, Wassim Bouachir ¹ , Brigitte Leblon ³, Armand LaRocque ⁴ , Ata Haddadi ⁵ and Nizar Bouguila ²

¹ Data Science Laboratory, University of Québec (TÉLUQ), Montréal, QC H2S 3L4, Canada; wassim.bouachir@teluq.ca

² Concordia Institute for Information Systems Engineering, Concordia University, Montréal, QC H3G 2E9, Canada; nizar.bouguila@concordia.ca

³ Faculty of Natural Resources Management, Lakehead University, Thunder Bay, ON P7B 5E1, Canada; bleblon@lakeheadu.ca

⁴ Faculty of Forestry, University of New Brunswick, Fredericton, NB E3B 5A3, Canada; armand.larocque@unb.ca

⁵ A&L Canada Laboratories, London, ON N5V 3P5, Canada; ahaddadi@alcanada.com

* Correspondence: hela.jemaa@mail.concordia.ca

Abstract: Accurate and efficient orchard tree inventories are essential for acquiring up-to-date information, which is necessary for effective treatments and crop insurance purposes. Surveying orchard trees, including tasks such as counting, locating, and assessing health status, plays a vital role in predicting production volumes and facilitating orchard management. However, traditional manual inventories are known to be labor-intensive, expensive, and prone to errors. Motivated by recent advancements in UAV imagery and computer vision methods, we propose a UAV-based computer vision framework for individual tree detection and health assessment. Our proposed approach follows a two-stage process. Firstly, we propose a tree detection model by employing a hard negative mining strategy using RGB UAV images. Subsequently, we address the health classification problem by leveraging multi-band imagery-derived vegetation indices. The proposed framework achieves an F1-score of 86.24% for tree detection and an overall accuracy of 97.52% for tree health assessment. Our study demonstrates the robustness of the proposed framework in accurately assessing orchard tree health from UAV images. Moreover, the proposed approach holds potential for application in various other plantation settings, enabling plant detection and health assessment using UAV imagery.

Keywords: unmanned aerial vehicle (UAV); DeepForest; YOLO; hard negative mining; vegetation indices (VIs); random forest



Citation: Jemaa, H.; Bouachir, W.; Leblon, B.; LaRocque, A.; Haddadi, A.; Bouguila, N. UAV-Based Computer Vision System for Orchard Apple Tree Detection and Health Assessment. *Remote Sens.* **2023**, *15*, 3558. <https://doi.org/10.3390/rs15143558>

Academic Editor: Thomas Alexandridis

Received: 8 June 2023

Revised: 28 June 2023

Accepted: 6 July 2023

Published: 15 July 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Tree diseases can have a significant impact on orchard quality and productivity, which is a major concern for the agricultural industry. Unhealthy or stressed orchard trees are more susceptible to pests, diseases, and environmental stressors, which can reduce the yield and quality of fruit and lead to financial losses for growers. Therefore, developing methods for surveying and monitoring tree health and production quality is essential for orchard management. This can help growers make informed decisions about practices such as irrigation, fertilization, and pest control, optimize orchard yield and quality, reduce input usage, and improve the long-term sustainability of orchard production systems.

Traditional methods of monitoring orchard tree health, such as manual inspection and visual examination, rely on human expertise to determine quantitative orchard tree parameters. These methods are labor-intensive, time-consuming, costly, and subject to errors. In recent years, remote sensing platforms such as satellites, airplanes, and unmanned aerial vehicles (UAVs) [1–4] have provided new tools that offer an alternative to traditional

methods. Deep neural networks (DNNs) [5] have also emerged as a powerful tool in the field of machine learning. The high spatial resolution provided by UAV images combined with computer vision algorithms have made tremendous advances in several domains such as forestry [6], agriculture [7], geology [8], surveillance [9], and traffic monitoring [10].

Motivated by the latest advances in machine-learning-based computer vision systems, this paper proposes a new framework to automatically detect orchard apple trees and assess their health from UAV multi-band imagery. The proposed framework adopts a two-stage approach. First, the tree localization problem is addressed using visual object detection. The second stage deals with tree health assessment through image patch classification.

1.1. Tree Detection

Over the years, both classical machine learning and deep learning methods have been extensively explored to address the tree detection problem.

Classical object detection methods often involve the utilization of handcrafted features and machine learning algorithms. Local binary pattern (LBP) [11], scale-invariant feature transform (SIFT) [12,13], and histogram of oriented gradients (HOG) [14,15] are the most frequently used handcrafted features in object detection. For example, the work in [16] presented a traditional method for walnut, mango, orange, and apple tree detection. It adopts the template matching image processing approach to very high resolution (VHR) Google Earth images acquired over a variety of orchard trees. The template is based on a geometrical optical model created from a series of parameters, such as illumination angles, maximum and ambient radiance, and tree size specifications. In [17,18], the authors detected palm trees on UAV RGB images by extracting a set of key points using the scale-invariant feature transform (SIFT). The key points are then analyzed with an extreme learning machine (ELM) classifier, which is a priori trained on a set of palm and no-palm tree keypoints. Similarly, ref. [19] employed a support vector machine (SVM) for image classification into vegetation and non-vegetation patches. Subsequently, the HOG feature extractor was applied on vegetation patches for feature extraction. These extracted features were then used to train a SVM to recognize palm tree images from background regions. The study in [20] proposed an object detection method using shape features for detecting and counting palm trees. The authors employed circular autocorrelation of the polar shape (CAPS) matrix representation as the shape feature and the linear SVM to standardize and reduce the dimensions of the feature. Finally, the study uses a local maximum detection algorithm based on the spatial distribution of standardized features to detect palm trees. The work in [7] presented a method to detect apple trees using multispectral UAV images. The authors identified trees using thresholding techniques applied on the Normalized Difference Vegetation Index (NDVI) and entropy images, as trees are chlorophyllous bodies that have high NDVI values and are heterogeneous with high entropy. The work in [21] proposed an automated approach to detect and count individual palm trees from UAV images. It is based on two processing steps: first, the authors employed the NDVI to perform the classification of image features as trees and non-trees. Then, palm trees were identified based on texture analysis using the Circular Hough Transform (CHT) and the morphological operators. In [22], the authors applied k-means to perform color-based clustering followed by a thresholding technique to segment out the green portion of the image. Then, trees were identified by applying an entropy filter and morphological operations on the segmented image.

On the other hand, numerous studies have investigated the use of deep-learning algorithms to detect trees in UAV RGB imagery. For instance, ref. [23] detected citrus and other crop trees from UAV images using a CNN algorithm applied to four spectral bands (i.e., green, red, near infrared and red edge). The initial detection was followed by a classification refinement procedure using superpixels derived from a Simple Linear Iterative Clustering (SLIC) algorithm and a thresholding technique to address the confusion between trees and weeds and deal with the difficulty in distinguishing small trees from large trees. In [24], the authors adopted a sliding window approach for oil palm tree

detection. A sliding window was integrated with a pre-trained AlexNet classifier to scan the input image and identify regions containing trees. The work in [25] exploited the use of state-of-the-art CNNs, including YOLO-v5 with its four sub-versions, YOLO-v3, YOLO-v4, and SSD300 in detecting date palm trees. Similarly, in [26], the authors explored the use of the YOLO-v5 with its subversions and DeepForest for the detection of orchard trees. In [27], three state-of-the-art object detection methods were evaluated for the detection of law-protected tree species: Faster Region-based Convolutional Neural Network (Faster R-CNN) [28], YOLOv3 [29], and RetinaNet [30]. Similarly, the work in [31] explored the use of Faster R-CNN, Single Shot Multi-Box Detector (SSD) [32], and R-FCN [33] architectures to detect seedlings.

Most of these works explored fine-tuning state-of-the-art object detectors for tree detection by taking an object detection model that is pre-trained on Benchmark datasets [34–36] and adapting it specifically for the task of detecting trees. However, applying these methods to UAV images has particular challenges [37] compared to conventional object detection tasks. For example, UAV images often have a large field of view with complex background regions, which can significantly disrupt detection accuracy. Furthermore, the objects of interest are often not uniformly distributed with respect to the background regions, creating an imbalance between positive and negative examples. Data imbalance can also be observed between easy and hard negative examples, since with UAV images, a large part of the background has regular patterns and can be easily analyzed for detection. We believe that applying deep learning detection algorithms directly in these situations is not an optimal choice [38], as they mostly assign the same weight to all training examples, so that during the training step easy examples may dominate the total loss and reduce training efficiency.

To mitigate this issue, hard negative mining (HNM) can be adopted for object detection. Various HNM approaches [37,39,40] involve iteratively bootstrapping a small set of negative examples, by selecting those that trigger a false positive alarm in the detector. For example, ref. [41] presented a training process of a state-of-the-art face detector by exploiting the idea of hard negative mining and iteratively updating the Faster R-CNN-based face detector with hard negatives harvested from a large set of background examples. Their method outperforms state-of-the-art detectors on the Face Detection Data Set and Benchmark (FDDB). Similarly, an improved version of faster R-CNN is proposed in [42], by using hard negative sample mining for object detection using PASCAL VOC dataset [36]. Likewise, ref. [43] used the bootstrapping of hard negatives to improve the performance of face detection on WIDER FACE dataset [44]. The authors pre-trained Faster R-CNN to mine hard negatives, before retraining the model. The work of [45] presented a cascaded Boosted Forest for pedestrian detection, which performs effective hard negative mining and sample reweighting, to classify the region proposals generated by RPN. The A-Fast-RCNN method, described in [46], adopts a different approach for generating hard negative samples, by using occlusion and spatial deformations through an adversarial process. The authors conducted their experiments on PASCAL VOC and MS-COCO datasets. Another approach to apply HNM using Single Shot multi-box Detector (SSD) is proposed in [47], where the authors use medium priors, anchor boxes with 20% to 50% overlap with ground truth boxes, to enhance object detector performance on the PASCAL VOC dataset. The proposed framework updates the loss function so that it considers the anchor boxes with partial and marginal overlap.

In our work, we propose a HNM approach for the tree detection stage, where the mined hard negative samples are included in the initial tree dataset to be considered during training. Then, we retrain the object detector using the true positive and false positive examples to enhance the discrimination power of the model. To the best of our knowledge, our work is the first to use a HNM approach for tree detection.

1.2. Tree Health Classification

Vegetation indices (VIs) have been introduced as indicators of vegetation status, as they provide information on the physiological and biochemical status of trees. These mathematical combinations of reflectance measurements are sensitive to different vegetation parameters, such as chlorophyll content, leaf area, and water stress. It has been shown through many studies [48–50] that by analyzing these indices, we can gain insights into the health and vitality of trees.

For example, the work in [51] presented a framework for orchard tree segmentation and health assessment. The proposed approach is applied to five different orchard tree species, namely plum, apricot, walnut, olive, and almond. Two vegetation indices, including visible atmospherically resistant index (VARI) and green leaf index (GLI) were used with the standard score (which is also known as z-score) for tree health assessment. The study in [52] proposed a process workflow for mapping and monitoring olive orchards at tree scale detail. Five VIs were investigated, including normalized difference vegetation index (NDVI), modified soil adjusted vegetation index 2 (MSAVI 2), normalized difference red edge Vegetation index (NDRE), modified chlorophyll absorption ratio index improved (MCARI2), and NDVI2. The authors applied statistical analyses to all calculated VIs. Similarly, ref. [53] presented an approach for Huanglongbing (HLB) disease detection on citrus trees. First, the trees were segmented using thresholding techniques applied on the normalized difference vegetation index (NDVI). Then, for each segmented tree, a total number of thirteen spectral features was computed, which include six spectral bands and seven vegetation indices. The indices studied were: NDVI, green normalized difference vegetation index (GNDVI), soil-adjusted vegetation index (SAVI), near infrared (NIR)—red(R), R/NIR, green (G)/R and NIR/R. A SVM classifier is then applied to distinguish between healthy and HLB-infected trees. The work in [54] presented a method for the identification of stress in olive trees. An SVM model was applied to VIs to classify each tree pixel into two categories: healthy and stressed. The work in [48] presented a method to monitor grapevine diseases affecting European vineyards. The authors explored the use of different features including spectral bands, vegetation indices and biophysical parameters. They conducted a statistical analysis for the selection of the best discriminating variables to separate between symptomatic vines including FD and GTD from asymptomatic vines (Case 1) and FD vines from GTD ones (Case 2).

In our work, we conducted a comprehensive investigation on the assessment of apple tree health using twelve different vegetation indices derived from multi-band UAV imagery. We also explored a set of machine learning classifiers to perform health classification based on vegetation indices.

The main contributions of this paper are summarized as follows:

- We propose a novel framework for automatic apple orchard tree detection and health assessment from UAV images. The proposed framework could be generalized for a wide range of other UAV applications that involve a detection/classification process.
- We adopt a hard negative mining approach for tree detection to improve the performance of the detection model.
- We formulate the tree health assessment problem as a supervised classification task based on vegetation indices calculated from multi-band images.
- We present an extensive experimental analysis covering various aspects of the proposed framework. Our analysis includes an ablation study demonstrating the importance of the HNM technique for tree detection, an exploration of several classification methods for health assessment, and a feature importance analysis within the context of health classification.

The rest of the paper is structured as follows. Section 2 provides a detailed description of the proposed framework. The experimental results are presented in Section 3 and discussed in Section 4, where we also suggest directions for future work. Finally, Section 5 concludes the paper.

2. Materials and Methods

2.1. Study Area

Images were captured during the summer of 2018 over two apple orchards in Souris, Prince Edward Island, Canada (Lat. 46.44633N, Long. 62.08151W), as shown in Figure 1. The surveyed orchard consists of 18 distinct types of apple trees, such as Cortland, Gala, Sunrise, Virginia Gold, Honey Gold, a mixed variety, Jona Gold, Russet, and Spygold. These trees vary in age, with some being young while others are older. The UAV images were taken using a MicaSense RedEdge3 multispectral camera (MicaSense Inc., Seattle, WA, USA). It has five sensors, including one sensor for each of the following spectral bands: B1 (blue) centered at 475 nm; B2 (green) centered at 560 nm; B3 (red) centered at 668 nm; B4 (RE) centered at 717 nm, and B5 (NIR) centered at 840 nm. The camera was mounted under a DJI Matrice 100 light Unmanned Aerial Vehicle (UAV). Its weight is slightly less than 2.0 kg. Both the camera and the UAV were connected to mission planner software from MicaSense to fly at 100 m above the ground with 70% overlap between adjacent images. Wind speed during image acquisition was less than 20 km/h and the weather condition was sunny-cloudy. The captured images were orthorectified and mosaicked together using Pix4D to cover the whole study area.

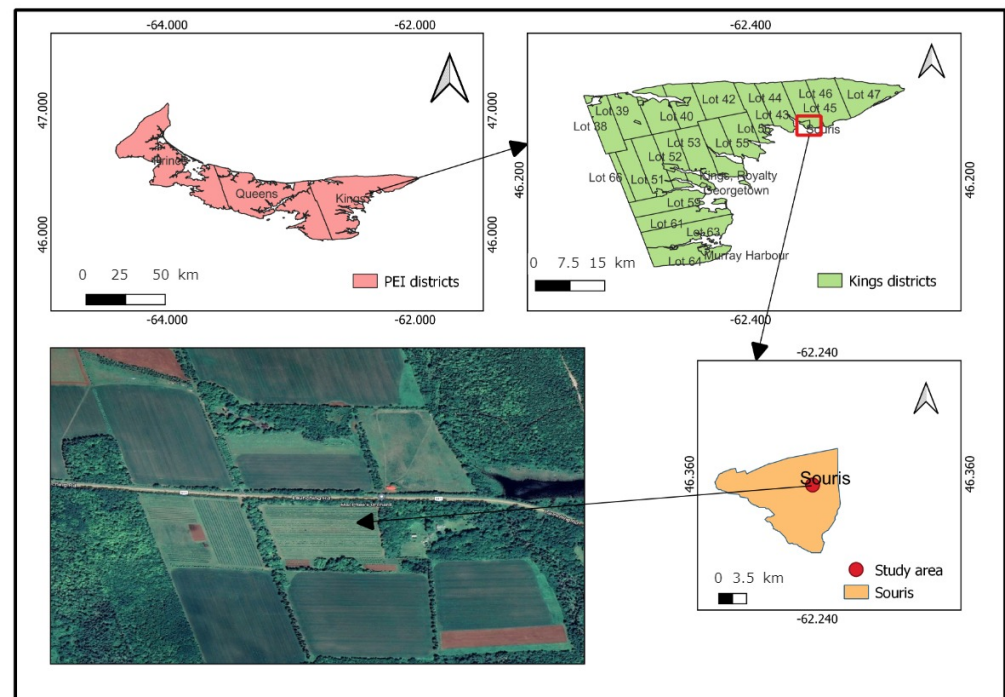


Figure 1. Location of the study area.

2.2. Dataset Construction

In order to prepare the data for object detection, the orthomosaic was subdivided into 515×512 -pixel patches using a regular grid. Then, we performed manual annotation for patches, which consists of localizing our objects of interest (apple trees) manually through bounding boxes using the open-source VGG Image Annotator (VIA) tool [55]. We used the field observations performed by specialists to validate our data annotation. These patches were split into three subsets: training, validation, and testing, using 3-fold cross-validation to ensure an impartial evaluation of object detection models.

To prepare the dataset for tree health classification, we utilized field inventory data. This involved gathering tree localization information through GPS, measuring tree parameters using specialized instruments, and recording all observations in a field notebook. By mapping these findings to the acquired images, two different tree health status were assigned: healthy and unhealthy to each individual tree. Finally, we performed data split-

ting using stratified cross-validation to divide the dataset into training, validation, and testing subsets.

Figure 2 illustrates two instances of trees captured at ground level: one depicting a healthy tree and the other representing an unhealthy tree. Stressed apple trees exhibit a reduction in chlorophyll content, red or brown discoloration on apple leaves, and infected flowers turning black, produced by a loss of photosynthetic metabolism due to bacterial disease or fungus [56,57].



Figure 2. Ground pictures of trees according to their health status. (a) Healthy tree. (b) Unhealthy tree. The unhealthy tree reveals stress symptoms that can be observed through its yellow and brown leaves, indicated by blue circles.

Our final dataset contained 2828 trees in total, where 2240 are healthy and 588 are unhealthy. Using stratified 10-fold cross-validation, the tree images were divided into training, validation, and testing sets, ensuring that the ratio of healthy to stressed trees remains consistent across subsets.

2.3. Proposed Framework

The objective of this research is to propose an automated framework for assessing tree health using UAV images. The proposed framework, as depicted in Figure 3, employs a two-stage approach to sequentially address two tasks: tree detection and tree health classification.

In the first stage, the input orthomosaic is subdivided into multiple patches of 512 by 512 pixels using a regular grid to fit the framework input. Then, each patch is subjected to a tree detection procedure, which aims to identify and locate trees within that patch. This is accomplished by outlining the detected trees using bounding boxes. Once the trees are identified through the tree detection module, the corresponding boxes containing the detected trees are extracted and cropped. These cropped boxes are then subjected to tree classification to determine their health status.

Due to potentially distracting complex backgrounds, tree recognition is a challenging task. As shown in Figure 4, some background objects (red rectangles) have a similar color, shape, and texture as the target tree objects (blue rectangles).

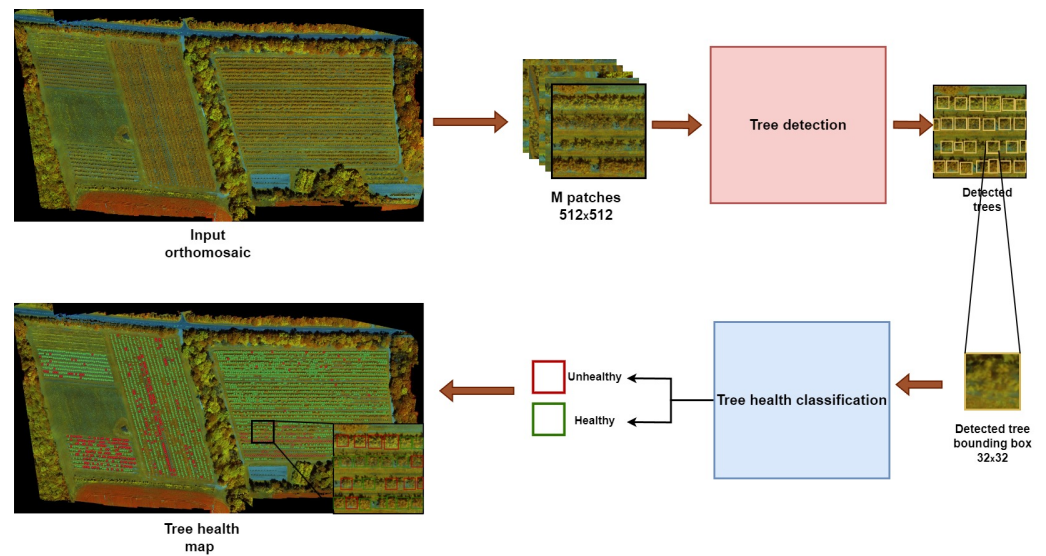


Figure 3. Overview of the proposed framework. In the health tree map, green rectangles correspond to healthy trees and red boxes correspond to stressed trees. The images are displayed in false color (near-infrared (NIR), red edge (RE), red (R)).

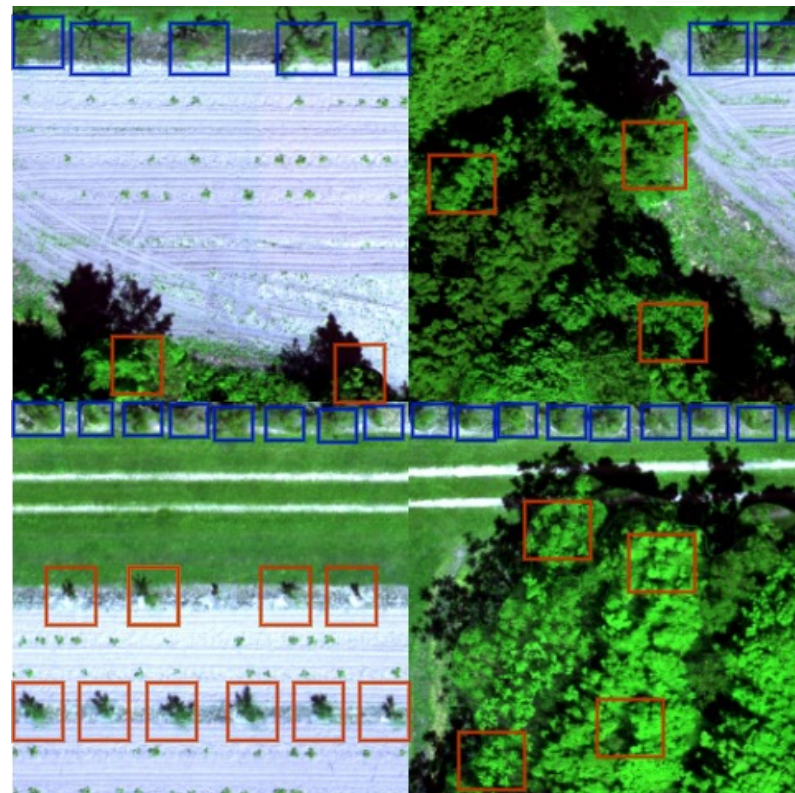


Figure 4. Detection results of a baseline object detector without the use of hard negative mining. Red rectangles are false detections (FP) and blue rectangles are correct detections (TP).

For tree health assessment, we investigated the use of vegetation indices as it has been shown that these spectral features are of great importance to identify the tree health status.

2.3.1. Tree Detection Stage

The proposed method for tree detection, depicted in Figure 5, starts by training a baseline object detector on UAV RGB images manually annotated for tree detection. The performance of the baseline model is then assessed to identify false detections, correspond-

ing to objects incorrectly detected as trees. In the object detection task, a detection is considered as false if its overlap with the ground truth object is lower than an overlap threshold. In our work, we select the samples that have no overlap with the target object and consider them as the hard negative samples. These hard negative examples are then used to create a new class, which is added to the training data. Afterward, we perform fine-tuning of the baseline tree detector using the updated training dataset that contains the two classes. Using false positive detections with no overlap with target objects as a new class is motivated by the fact that these samples represent a source of disturbance in the training data, which can lead to inaccurate detection. By doing this, the model can learn to distinguish between true positive (trees) and false positive (non-trees) examples. To address the imbalance problem between the target class and the hard negative class, we use the focal loss [30] as the objective function during fine-tuning of the object detector. This method helps to improve the accuracy of the tree detector by reducing the impact of the noise caused by hard negative samples in the training data.

In our work, we explored two state-of-the-art baseline models for object detection: YOLO [58] using its COCO pre-trained model and DeepForest prebuilt model [59] trained on The National Ecological Observatory Network (NEON [60]) crowns data set. The two baseline architectures are explained further in the Appendix A. The choice of YOLO-v5 and DeepForest as baseline models is motivated by their impressive performance, popularity, and suitability for the task. These models have achieved state-of-the-art accuracy and efficiency in object detection, making them strong candidates for comparison. Note that we finally adopted DeepForest as a baseline detection model in our framework. This choice is based on a performance analysis of the two architectures, as discussed further.

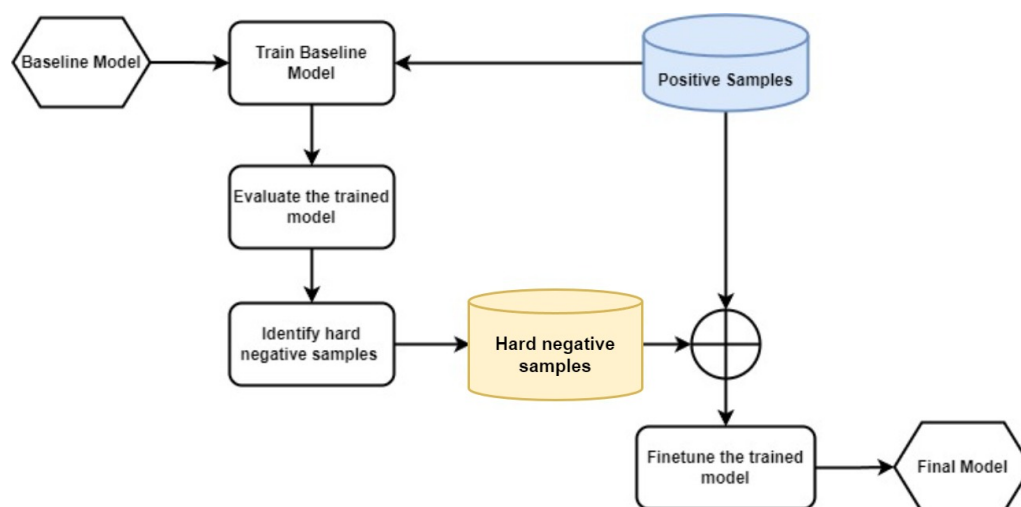


Figure 5. Tree detection stage using a hard negative mining strategy.

2.3.2. Tree Health Assessment

Figure 6 summarizes the proposed method for tree health mapping. It follows a machine learning pipeline. Each detected tree image from the first stage is extracted and cropped by selecting pixels within a rectangular area around the tree center using a 32-pixel box size.

First, we compute vegetation indices from the five-band raw reflectance tree images. Among the large number of vegetation indices that are typically used as indicators of vegetation status [61], there is a subset of VIs that are commonly applied for health assessment [62–64]. Because of their high sensitivity to changes in moisture content, pigment indices, and vegetation health, red, green, blue, red edge, and near-infrared bands are frequently used to evaluate the status of vegetation [65]. Using these bands, a total of 12 vegetation indices (Table 1) were calculated based on their ability to reveal stress. The difference vegetation index (DVI) and the normalized difference vegetation index (NDVI)

were estimated based on the evidence of correlation with tree stress [53,66]. The green normalized difference vegetation index (GNDVI) and normalized difference red-edge (NDRE) were also estimated, as both have demonstrated high sensitivity to variations in chlorophyll concentrations [65,67,68]. As an ideal vegetation index should be most sensitive to vegetation dynamics and not sensitive to changes in soil background or variable illumination or atmospheric effects, The intensity-normalization method [69] was introduced to minimize errors due to these changes. The normalized Green (NG), normalized Red (NR), and normalized NIR (NNIR) were calculated using this method and included with features.

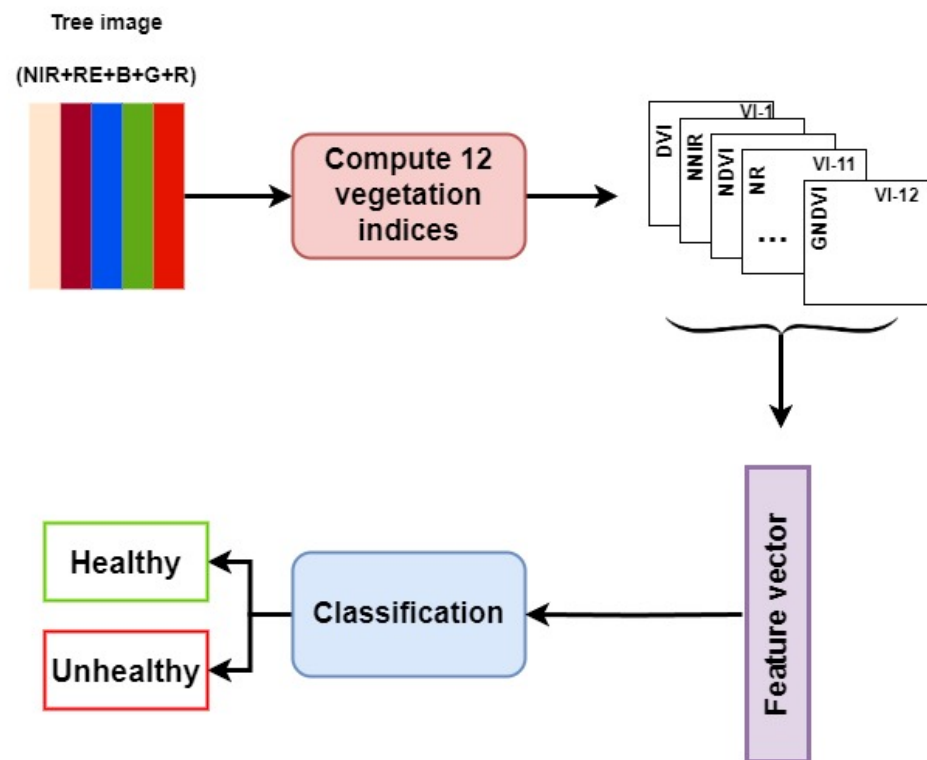


Figure 6. Tree health assessment using a ML classifier based on a selected set of vegetation indices. The feature vector is composed of the mean of each vegetation index map generated from the tree image patch.

Table 1. Vegetation indices derived from UAV band reflectance.

Vegetation Index	Equation	Reference
Difference Vegetation Index	$DVI = \text{Near-infrared (NIR)} - \text{Red}$	[70]
Generalized Difference Vegetation Index	$GDVI = \text{NIR} - \text{Green}$	[71]
Green Normalized Difference Vegetation Index	$GNDVI = (\text{NIR} - \text{Green}) / (\text{NIR} + \text{Green})$	[72]
Green-Red Vegetation Index	$GRVI = \text{NIR} / \text{Green}$	[71]
Normalized Difference Aquatic Vegetation Index	$NDAVI = (\text{NIR} - \text{Blue}) / (\text{NIR} + \text{Blue})$	[73]
Normalized Difference Vegetation Index	$NDVI = (\text{NIR} - \text{Red}) / (\text{NIR} + \text{Red})$	[70]
Normalized Difference Red-Edge	$NDRE = (\text{NIR} - \text{RedEdge}) / (\text{NIR} + \text{RedEdge})$	[74]
Normalized Green	$NG = \text{Green} / (\text{NIR} + \text{Red} + \text{Green})$	[69]
Normalized Red	$NR = \text{Red} / (\text{NIR} + \text{Red} + \text{Green})$	[69]
Normalized NIR	$NNIR = \text{NIR} / (\text{NIR} + \text{Red} + \text{Green})$	[69]

Table 1. Cont.

Vegetation Index	Equation	Reference
Red simple ratio Vegetation Index	$RVI = NIR/Red$	[75]
Water Adjusted Vegetation Index	$WAVI = (1.5*(NIR - Blue))/((NIR + Blue) + 0.5)$	[73]

Each detected tree image is represented by a feature vector that contains the average value of each vegetation index map. This feature vector comprises 12 values, where each value represents the average of a specific vegetation index map. To address class unbalance, we applied the synthetic minority oversampling technique (SMOTE) to oversample the examples in the minority class.

Finally, we applied a set of classifiers to perform the classification of the health status of apple trees. These classifiers were chosen based on their suitability for handling multivariate data and their recognized performance in similar studies. In our work, we explored the use of support vector machines (SVM) [76], decision tree [77], random forests (RF) [78], k-nearest neighbors (KNN) [79], and light gradient boosting machine (LightGBM) [80], which are explained further in Appendix B.

2.4. Implementation

All experiments were conducted on a PC with Intel Core i7-7700 CPU, NVIDIA GeForce GTX1080 GPU, and 64 GB of RAM. The operating system used by the PC was Windows 10. For the tree detection stage, our approach involved using two benchmark models. We applied transfer learning to the YOLO model (with 300 epochs and a batch size of 16) and fine-tuned the DeepForest model (with 100 epochs and a batch size of 16). After mining hard negatives and updating our dataset, we further fine-tuned the detection model using the updated dataset. We experimented with freezing the entire backbone and varying numbers of layers. YOLO model (with 4 frozen backbone layers) yielded the best results. In the case of the DeepForest model, we performed a full retraining process using the updated dataset. For the tree health classification, we explored the use of various supervised classifiers. To handle the issue of class imbalance, we employed the SMOTE technique. A grid search strategy was used to identify the best parameters for each classifier. With respect to our chosen model, Random Forest, we selected the Gini index as the splitting criterion.

2.5. Evaluation Metrics

To evaluate the performance of our framework for **tree detection**, we use the following metrics.

- Precision P_d (Equation (1)) is the percentage of correct detections among all the detected trees.

$$P_d = \frac{TP_d}{TP_d + FP_d} \quad (1)$$

- Recall R_d (Equation (2)) is the percentage of correctly detected trees over the total number of trees in the ground truth.

$$R_d = \frac{TP_d}{TP_d + FN_d} \quad (2)$$

- $F1\text{-score}_d$ (Equation (3)) is the harmonic average of precision and recall.

$$F1\text{-score}_d = 2 * \frac{P_d * R_d}{P_d + R_d} \quad (3)$$

In Equations (1)–(3), the subscript d denotes detection, TP_d is the number of true positives (i.e., correctly detected trees), FP_d is the number of false positives (i.e., regions incorrectly detected as trees), and FN_d denotes the number of false negatives (i.e., the number of missed trees). On a test image, a detection is considered as correct if the overlap between the detected tree and the tree in the ground truth was greater than 50%. The overlap between the detection and the ground truth is computed using the Intersection Over Union (IOU) metric (Equation (4)).

$$IOU = \frac{Area(B_1 \cap B_2)}{Area(B_1 \cup B_2)}, \quad (4)$$

where B_1 is the ground truth bounding box and B_2 is the predicted bounding box.

To evaluate the performance of our framework for tree **health classification**, we use the following metrics.

- Precision P_c (Equation (5)) is defined as the ratio of correct classifications for a given class to the total number of classifications made for that class.

$$P_c = \frac{TP_c}{TP_c + FP_c} \quad (5)$$

- Recall R_c (Equation (6)) is defined as the ratio of correct classifications for a given class to the total number of instances that actually belong to that class.

$$R_c = \frac{TP_c}{TP_c + FN_c} \quad (6)$$

- $F1\text{-score}_c$ is the harmonic average of P_c and R_c of a given class.
- Accuracy (Equation (7)) is defined as the ratio of the correct classifications to the total number of tree instances classified.

$$Accuracy = \frac{\text{Number of correct classifications}}{\text{Total number of trees classified}} \quad (7)$$

In Equations (5)–(7), the subscript c denotes classification, TP_c is the number of correctly classified instances of a given class, FP_c is the number of instances incorrectly classified as belonging to a given class, and FN_c is the number of instances incorrectly classified as not belonging to a given class.

3. Results

3.1. Tree Detection

Table 2 shows the detailed and overall cross-validation results of the proposed tree detection approach using a hard negative mining approach for each baseline model. We can notice that the proposed detection method using the two baselines is stable across all folds. This demonstrates the robustness of our model, which is able to perform well and consistently on different partitions. We can also see that the DeepForest model is slightly better than YOLO, achieving a higher F1-score average of 86.24%.

DeepForest's superior performance could be explained by the fact that it is specifically designed for tree detection in aerial imagery. It offers several kernel-specific advantages over YOLO for tree detection. DeepForest is trained on a large dataset that includes UAV images of different tree species, ages, and environmental conditions. This specialized training allows it to leverage domain-specific knowledge and detect trees with high accuracy and specificity, even in challenging scenarios such as partial occlusions or low contrast with the background.

Table 2. Detailed and overall cross-validation results of the tree detection stage using hard negative mining approach in terms of precision, recall, and F1-score. Values in bold font correspond to the best-achieved results.

Fold	Baseline 1: DeepForest			Baseline 2: YOLO		
	P_d (%)	R_d (%)	F1-Score _d (%)	P_d (%)	R_d (%)	F1-Score _d (%)
Fold 1	82.25	87.24	84.67	83.89	87.27	85.15
Fold 2	87.57	88.06	87.82	79.12	92.35	84.91
Fold 3	87.87	84.73	86.27	83.09	87.45	84.38
Average	85.85	86.67	86.24	82.01	88.99	84.81

We conducted an ablation study to evaluate the importance of mining hard examples for tree detection. We reported the results of training baseline object detectors without hard negative samples by comparison to our HNM-based approach. Table 3 reports the overall 3-fold cross-validation results using two baseline models: DeepForest and YOLO-v5 [81]. We train YOLO-v5 using the COCO pre-trained model and DeepForest using its prebuilt model. The reported results show that both YOLO and DeepForest benefited from hard negative mining.

Table 3. Ablation study result of the detection step: overall 3-fold cross-validation results of two baseline models applied with and without HNM. Values in bold font correspond to the best results.

Baseline Model	P_d (%)	R_d (%)	F1-Score _d (%)
<i>Without HNM</i>			
Baseline 1: DeepForest	84.82	86.18	85.46
Baseline 2: YOLO	79.40	88.05	82.64
<i>With HNM</i>			
Baseline 1: DeepForest	85.85	86.67	86.24
Baseline 2: YOLO	82.01	88.99	84.81

YOLO fine-tuned with the mined hard negatives achieved an F1-score of 84.81%, outperforming the YOLO baseline by 2.17%, which suggests that the detector learns to eliminate a number of false detections. Using the DeepForest model, the inclusion of hard negatives in training improves the performance compared to the baseline, with an improvement of 0.78% based on the F1-score. Overall, our results demonstrate the importance of HNM in tree detection to improve the detection ability in both baseline models.

3.2. Health Classification

The tree health assessment stage is formulated as a patch image classification problem. We applied different classifiers including SVM, decision tree, and Random Forest using twelve spectral features computed from five-band imagery (Red, green, blue, near-infrared, and red edge). Table 4 presents the performance of each classifier for each class in terms of precision, recall, F1-score, and overall accuracy.

We can notice that all the explored classifiers performed well in the two classes. Specifically, they achieved higher results for the healthy class. Based on the overall accuracy, RandomForest [78] is the best classifier, achieving 97.52%. Random forests classifier adopts an ensemble approach by combining multiple decision trees to improve the accuracy and robustness of predictions. It randomly selects subsets of features and samples from the dataset to create a set of decision trees. Each tree independently makes predictions based on the selected features. The final prediction is obtained by combining the predictions of all the trees in the forest.

Table 4. Overall cross-validation results of different ML classifiers for tree health classification using vegetation indices in terms of precision, recall, F1-score, and overall accuracy. Values in bold font correspond to the best-achieved accuracy.

Health Status	P_c (%)	R_c (%)	F1-Score $_c$ (%)	Accuracy (%)
<i>Random Forest Classifier (RF)</i>				
Healthy	98.8	98.8	98.8	97.52
Unhealthy	95.5	95.5	95.5	
<i>Light Gradient Boosting Machine (LightGBM)</i>				
Healthy	99	98.8	98.9	97.47
Unhealthy	95.6	96.1	95.8	
<i>K-nearest neighbors (KNN)</i>				
Healthy	98.6	98.1	98.4	97.07
Unhealthy	92.9	95	93.9	
<i>Support Vector Machine (SVM)</i>				
Healthy	99.4	95.8	97.6	96.31
Unhealthy	86.2	97.8	91.6	
<i>Decision Tree Classifier (DT)</i>				
Healthy	98.4	97.9	98.1	95.91
Unhealthy	92.3	93.9	93.1	

These results show the effectiveness of random forest in accurately classifying tree health status based on the selected vegetation indices.

In Table 5, we present the feature importance ranking generated by the Random Forest classifier. It allows us to assess the contribution of each feature to the overall accuracy of the model. The feature importance is determined using a Gini importance metric [82], which measures the frequency with which a particular feature is utilized for data splitting across all trees in the forest. This metric is weighted by the number of samples in each node.

Table 5. Feature importance using Random Forest.

Ranking	Feature	Contribution
1	NR	34%
2	NNIR	25%
3	RVI	16%
4	NDAVI	15%
5	DVI	10%

We can conclude that the Normalized Red (NR) and Normalized Near-Infrared (NNIR) features are the most important features used by RandomForest to distinguish between healthy and stressed trees.

The Normalized Red (NR) index measures the reflectance of red light in vegetation. It is often used to estimate the amount of chlorophyll in plant leaves, which is an indicator of plant health. Chlorophyll is responsible for photosynthesis, and healthy trees typically have higher chlorophyll content than unhealthy trees. Therefore, the contribution of NR suggests that chlorophyll content is a key factor in determining tree health.

On the other hand, the Normalized Near-Infrared (NNIR) index measures the reflectance of near-infrared light in vegetation. It is often used to estimate the amount of vegetation biomass, which is an indicator of tree productivity. Healthy trees typically have a higher biomass than unhealthy trees. Therefore, the importance of NNIR suggests that biomass is also a key factor in determining tree health.

Taken together, the importance of NR and NNIR indicates that tree health status is determined by a combination of factors related to chlorophyll content and biomass.

4. Discussion

Over the years, the use of unmanned aerial vehicles (UAVs) in precision agriculture has experienced significant growth. Initially used for crop monitoring, UAV applications have expanded to include pest and disease control, treatment, and even providing real-time decision support. Our study specifically focuses on mapping individual tree crowns and evaluating their health using high-resolution UAV imagery and computer vision techniques. Our proposed method utilizes hard negative mining strategy to perform individual tree detection. To assess the health of the detected trees, vegetation indices derived from multi-band UAV images were employed.

4.1. Tree Detection

Our proposed method for individual tree detection based on the HNM technique demonstrated its efficiency compared to the use of a baseline object detector. The inclusion of hard negative samples during model training improves the model's ability to distinguish between the target object (apple trees in our case) and other background objects. We believe that the application of the hard negative mining (HNM) learning strategy could be promising on UAV images as this kind of image displays the challenge of a complex background that may distract the visual object detection as shown in Figure 4. To the best of our knowledge, this is the first work to apply the HNM strategy for tree detection. In comparison to existing works in the literature, our proposed tree detection method achieves interesting results that are comparable or better than other studies. Our F1-score of 86.24% is comparable to the F1-score of [83] (86%), who uses a local maximum algorithm to detect pine trees on UAV-derived canopy height model imagery. Similarly, ref. [20] obtained an F1-score of 75.75% using a SVM based on shape features to detect palm trees on UAV RGB images, which is also slightly lower than our achieved F1-score.

Overall, our results demonstrate the effectiveness of hard negative mining approach for tree detection and highlight the importance of using specialized models like DeepForest for this task.

4.2. Trees Health Assessment

In comparison to previous studies carried out to address tree health assessment, our classification accuracy of 98.06% was higher than the one obtained by [84] (93.17%) using an SVM classifier to map the degree of damage caused by forest pests on UAV-based hyperspectral images. It was also higher than the one obtained by [85], who mapped tree health status using multispectral imagery (95%) or hyperspectral imagery (91%). Our method also outperforms statistical methods reviewed in [86] using generalized linear models (GLM), maximum entropy (ME), and random forests (RF), achieving an overall accuracy of 87.3%, 93.9%, and 97.1%, respectively, to identify areas affected by bark beetles on satellite imagery. Generally, these results showed that our approach utilizing supervised classification via Random Forests provides similar or better levels of accuracy to comparable studies. Furthermore, the conducted feature importance analysis underscores the contribution on NR and NNIR indices in assessing apple trees' health. This highlights the value and efficiency of vegetation indices derived from remote sensing imagery in autonomously assessing vegetation health, eliminating the need for manual human intervention.

In the context of precision agriculture, these findings align with numerous studies that have leveraged Unmanned Aerial Vehicles (UAVs) for efficient and precise applications such as monitoring water stress [87,88], production volume estimation [89], and assessment of resource efficiencies [90]. This reinforces the paradigm shift towards technology-driven agricultural practices, promoting enhanced productivity and sustainable resource use.

Our future work aims to investigate the use of other bands such as red edge and near-infrared for tree detection, as they may contain relevant information for accurate detection. Moreover, hyperspectral imaging technology can also be explored for tree detection. It can provide both spatial geometric information, such as the tree crown shape, size, and

texture, as well as the relationship between adjacent tree crowns, and precise spectral information. Forestry and agriculture have benefited from the use of this technology [91] for tree identification, species classification [92], tree health monitoring [93], and biodiversity assessment [93].

5. Conclusions

In this paper, an effective two-stage framework is proposed to automate orchard apple tree health assessment from UAV images. The first stage addresses the tree detection problem using a hard negative mining approach to improve tree detection performance. The second stage deals with tree health classification, where we use Random forest based on twelve selected vegetation indices. The proposed framework has the potential to reduce errors associated with manual field inventory, and eliminate the need for time-consuming and expensive field surveys. Furthermore, our framework produces maps showcasing stressed trees across orchards. Utilizing a tree health map for an orchard provides multiple advantages for growers. It enables the identification of potential issues at an early stage, precise intervention, strategic planning for harvesting, and enhanced crop management. This map can be used as a guide for further investigation by growers. Thus, orchard managers can benefit from the early detection of stress to perform targeted plantation management strategies. By leveraging this valuable information, growers can enhance productivity, minimize risks, and promote the long-term health and success of their orchard.

Author Contributions: Conceptualization, H.J., W.B., N.B. and B.L.; methodology, H.J., W.B., N.B. and B.L.; software, H.J.; validation, H.J.; formal analysis, H.J.; resources, W.B., B.L., A.L., N.B. and A.H.; data curation, H.J., A.L. and A.H.; writing—original draft preparation, H.J.; writing—review and editing, W.B., B.L., N.B. and A.L.; supervision, W.B., N.B. and B.L.; funding acquisition, W.B. and B.L. All authors have read and agreed to the published version of the manuscript.

Funding: This study was funded by an NSERC-CRD grant and an NSERC Discovery grant awarded to Brigitte Leblon. This work was also supported by the NSERC Discovery grant number RGPIN-2020-04937 awarded to Wassim Bouachir.

Data Availability Statement: Not applicable.

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

Appendix A. Brief Description of the Explored Object Detectors Used as Baseline Models in the Hard Negative Mining Approach to Address Tree Detection, That Is, YOLO and DeepForest

DeepForest [59] is a deep-learning model developed to detect individual trees on high-resolution RGB imagery. It is mainly based on the Retinanet model [30], a one-stage object detector that enables the focal loss function to address the excessive foreground-background class imbalance between RetinaNet and state-of-the-art two-stage detectors like Faster R-CNN with FPN while operating at higher speeds. RetinaNet is a network architecture based on ResNet as a backbone [94] that generates a rich, multiscale convolutional Feature Pyramid Network (FPN) [95] that is connected to two subnetworks: one for classifying anchor boxes and another one for regressing object boxes. There are several barriers to applying deep learning to ecological applications including insufficient technical expertise, a lack of a large amount of training data, and the need for significant computational resources. A key advantage of DeepForest's neural network is that we can retrain the prebuilt model to learn new tree features and image backgrounds while leveraging information from the existing model weights based on data from a diverse set of forests. The DeepForest prebuilt model was trained on data from the National Ecological Observatory Network (NEON) using a semi-supervised approach. The model was pre-trained on data from 22 NEON sites using an unsupervised LiDAR-based algorithm to generate millions of moderate-quality annotations for model pretraining. The pre-trained model was then retrained based on over 10,000 hand annotations of airborne RGB imagery from six sites.

You Only Look Once (YOLO) is a one-stage object detector. Its network architecture is made up of three primary parts: the backbone, the neck, and the head. The YOLO-V5 model backbone is based on the Cross Stage Partial Network (CSP-Net). It aims to extract high-level features while maintaining high accuracy and shortening the model processing time. This is accomplished by splitting the base layer's feature map into two sections and then merging them using a suggested cross-stage hierarchy. The fundamental idea is to separate the gradient flow in order to make it propagate over several network paths. Furthermore, CSPNet can significantly minimize the amount of computation required and increase both the speed and accuracy of inference. It deals with three important problems: Strengthening the learning ability of a CNN, removing computational bottlenecks, and reducing memory costs. The model neck is used to collect feature maps from various stages to generate feature pyramids. At this level, the Path Aggregation Network (PANet) [96] and Spatial Pooling Pyramid (SPP) [97] are adopted for parameter aggregation from different backbone levels for different detector levels, instead of FPN used in YOLO-v3 [29]. Finally, for the head, the YOLO-v3 anchor-based head architecture is adopted for the used YOLO version. Within each portion of the network, YOLO-V5 has numerous key components, including Focus, CBL (Convolution, Batch Normalization, and Leaky- ReLU), CSP (Cross-Stage Partial Connections), and SPP (Spatial Pyramid Pooling). The Focus module divides the input image into four parallel slices, which are then utilized to construct feature maps with the CBL module. The CBL module is a basic feature extraction module that employs a convolution operation, batch normalization, and a leaky-ReLU activation function. The CSP module is a CSPNetbased module that is used to improve the model's learning capability. The SPP module is a module that allows the mixing and pooling of spatial elements. It concatenates to its initial features after downsampling the input features through three parallel max-pooling layers. YOLO-v5 implies some new data augmentation techniques such as Mosaic and SAT (Sel Adversarial Training).

Appendix B. Brief Description of the Explored Machine Learning Classifiers for Tree Health Classification

KNN [79] is a machine learning algorithm used for both classification and regression tasks. Its basic idea is to classify or predict the label of a given input data point by finding the K nearest neighbors in the training dataset and taking a majority vote among their labels. To determine the K nearest neighbors, KNN uses a distance metric such as Euclidean distance, Manhattan distance, or cosine similarity to measure the similarity between the input data point and each data point in the training dataset. The algorithm then selects the K data points with the smallest distance to the input data point. KNN is a non-parametric algorithm, which means it does not make any assumptions about the underlying distribution of the data. It is also known as a lazy learner, as it simply memorizes the training dataset and does not learn a model or parameters. One advantage of KNN is its simplicity and ease of implementation, especially for small datasets. However, it can be computationally expensive and may not perform well on high-dimensional or sparse datasets. Additionally, the choice of K and the distance metric can have a significant impact on the performance of the algorithm.

LightGBM [80] is a gradient-boosting framework that uses tree-based learning algorithms for solving supervised learning problems. It uses a technique called histogram-based gradient boosting, which discretizes continuous features into discrete bins to speed up the training process. It also implements features such as bagging, feature sub-sampling, and regularization to prevent overfitting and improve model generalization. One of the key advantages of LightGBM is its scalability, making it ideal for large datasets with millions of samples and thousands of features. It also has high accuracy and is often used for a wide range of applications, including image classification, text classification, and recommendation systems. Overall, LightGBM is a powerful machine-learning framework that provides efficient and accurate solutions for a variety of supervised learning problems.

SVM [76] is a supervised learning method that separates data into classes by finding the optimal hyperplane that maximally separates the classes. Its basic idea is to find the optimal hyperplane that maximizes the margin between the classes. The margin is the distance between the hyperplane and the closest data points of each class. SVM aims to find the hyperplane that maximizes this margin, which helps to minimize the generalization error and improves the model's ability to classify new data.

The DT [77] classifier is a machine learning algorithm used for classification tasks. It works by partitioning the input data into smaller subsets based on the values of the input features, and recursively splitting the subsets until a pure subset is obtained, or until a predefined stopping criterion is met. It uses a criterion, such as Gini impurity or entropy, to measure the homogeneity of the data at each node and determine the best feature to split the data. It recursively splits the data based on the feature that maximally reduces the impurity of the data, until a stopping criterion is met.

References

- Kim, J.; Kim, S.; Ju, C.; Son, H.I. Unmanned aerial vehicles in agriculture: A review of perspective of platform, control, and applications. *IEEE Access* **2019**, *7*, 105100–105115. [\[CrossRef\]](#)
- Barbedo, J.G.A. A review on the use of unmanned aerial vehicles and imaging sensors for monitoring and assessing plant stresses. *Drones* **2019**, *3*, 40. [\[CrossRef\]](#)
- Costa, F.G.; Ueyama, J.; Braun, T.; Pessin, G.; Osório, F.S.; Vargas, P.A. The use of unmanned aerial vehicles and wireless sensor network in agricultural applications. In Proceedings of the 2012 IEEE International Geoscience and Remote Sensing Symposium, Munich, Germany, 22–27 July 2012; pp. 5045–5048.
- Urbahs, A.; Jonaite, I. Features of the use of unmanned aerial vehicles for agriculture applications. *Aviation* **2013**, *17*, 170–175. [\[CrossRef\]](#)
- LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [\[CrossRef\]](#)
- Bouachir, W.; Ihou, K.E.; Gueziri, H.E.; Bouguila, N.; Bélanger, N. Computer Vision System for Automatic Counting of Planting Microsites Using UAV Imagery. *IEEE Access* **2019**, *7*, 82491–82500. [\[CrossRef\]](#)
- Haddadi, A.; Leblon, B.; Patterson, G. Detecting and Counting Orchard Trees on Unmanned Aerial Vehicle (UAV)-Based Images Using Entropy and Ndvi Features. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2020**, *43*, 1211–1215. [\[CrossRef\]](#)
- Zhang, Y.; Wang, G.; Li, M.; Han, S. Automated classification analysis of geological structures based on images data and deep learning model. *Appl. Sci.* **2018**, *8*, 2493. [\[CrossRef\]](#)
- Geng, L.; Zhang, Y.; Wang, P.; Wang, J.J.; Fuh, J.Y.; Teo, S. UAV surveillance mission planning with gimbaled sensors. In Proceedings of the 11th IEEE International Conference on Control & Automation (ICCA), Taichung, Taiwan, 21 November 2014; pp. 320–325.
- Kanistras, K.; Martins, G.; Rutherford, M.J.; Valavanis, K.P. A survey of unmanned aerial vehicles (UAVs) for traffic monitoring. In Proceedings of the 2013 International Conference on Unmanned Aircraft Systems (ICUAS), Atlanta, GA, USA, 28–31 May 2013; pp. 221–234.
- Ojala, T.; Pietikainen, M.; Maenpää, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 971–987. [\[CrossRef\]](#)
- Sedaghat, A.; Mokhtarzade, M.; Ebadi, H. Uniform robust scale-invariant feature matching for optical remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2011**, *49*, 4516–4527. [\[CrossRef\]](#)
- Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [\[CrossRef\]](#)
- Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the CVPR, San Diego, CA, USA, 20–26 June 2005; pp. 886–893.
- Shao, W.; Yang, W.; Liu, G.; Liu, J. Car detection from high-resolution aerial imagery using multiple features. In Proceedings of the 2012 IEEE International Geoscience and Remote Sensing Symposium, Munich, Germany, 22–27 July 2012; pp. 4379–4382.
- Maillard, P.; Gomes, M.F. Detection and counting of orchard trees from vhr images using a geometrical-optical model and marked template matching. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *3*, 75. [\[CrossRef\]](#)
- Malek, S.; Bazi, Y.; Alajlan, N.; AlHichri, H.; Melgani, F. Efficient framework for palm tree detection in UAV images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 4692–4703. [\[CrossRef\]](#)
- Bazi, Y.; Malek, S.; Alajlan, N.A.; Alhichri, H.S. An automatic approach for palm tree counting in UAV images. In Proceedings of the 2014 IEEE Geoscience and Remote Sensing Symposium, Quebec City, QC, Canada, 13–18 July 2014; pp. 537–540.
- Wang, Y.; Zhu, X.; Wu, B. Automatic detection of individual oil palm trees from UAV images using HOG features and an SVM classifier. *Int. J. Remote Sens.* **2019**, *40*, 7356–7370. [\[CrossRef\]](#)
- Manandhar, A.; Hoegner, L.; Stilla, U. Palm tree detection using circular autocorrelation of polar shape matrix. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *3*, 465. [\[CrossRef\]](#)
- Mansoori, S.A.; Kunhu, A.; Ahmad, H.A. Automatic palm trees detection from multispectral UAV data using normalized difference vegetation index and circular Hough transform. *Remote Sens.* **2018**, *10*, 1792, 11–19.

22. Hassaan, O.; Nasir, A.K.; Roth, H.; Khan, M.F. Precision forestry: Trees counting in urban areas using visible imagery based on an unmanned aerial vehicle. *IFAC-PapersOnLine* **2016**, *49*, 16–21. [[CrossRef](#)]
23. Csillik, O.; Cherbini, J.; Johnson, R.; Lyons, A.; Kelly, M. Identification of citrus trees from unmanned aerial vehicle imagery using convolutional neural networks. *Drones* **2018**, *2*, 39. [[CrossRef](#)]
24. Li, W.; Fu, H.; Yu, L. Deep convolutional neural network based large-scale oil palm tree detection for high-resolution remote sensing images. In Proceedings of the 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), New York, NY, USA, 11–13 December 2017; pp. 846–849.
25. Jintasuttisak, T.; Edirisinghe, E.; Elbattay, A. Deep neural network based date palm tree detection in drone imagery. *Comput. Electron. Agric.* **2022**, *192*, 106560. [[CrossRef](#)]
26. Jemaa, H.; Bouachir, W.; Leblon, B.; Bouguila, N. Computer vision system for detecting orchard trees from UAV images. *ISPRS-Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2022**, *43*, 661–668. [[CrossRef](#)]
27. 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](#)]
28. 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*, 1137–1149. [[CrossRef](#)] [[PubMed](#)]
29. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
30. Lin, T.Y.; Goyal, P.; Girshick, R.B.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *42*, 318–327. [[CrossRef](#)]
31. Fromm, M.; Schubert, M.; Castilla, G.; Linke, J.; McDermid, G. Automated detection of conifer seedlings in drone imagery using convolutional neural networks. *Remote Sens.* **2019**, *11*, 2585. [[CrossRef](#)]
32. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.E.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the European Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 21–37.
33. Dai, J.; Li, Y.; He, K.; Sun, J. R-fcn: Object detection via region-based fully convolutional networks. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 379–387.
34. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
35. Lin, T.Y.; Maire, M.; Belongie, S.J.; 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.
36. Hoiem, D.; Divvala, S.K.; Hays, J.H. Pascal VOC 2008 challenge. *World Lit. Today* **2009**, *24*.
37. Zhang, L.; Wang, Y.; Huo, Y. Object detection in high-resolution remote sensing images based on a hard-example-mining network. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 8768–8780. [[CrossRef](#)]
38. Xia, G.S.; Bai, X.; Ding, J.; Zhu, Z.; Belongie, S.J.; Luo, J.; Datcu, M.; Pelillo, M.; Zhang, L. DOTA: A Large-Scale Dataset for Object Detection in Aerial Images. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3974–3983.
39. Jin, S.; RoyChowdhury, A.; Jiang, H.; Singh, A.; Prasad, A.; Chakraborty, D.; Learned-Miller, E.G. Unsupervised Hard Example Mining from Videos for Improved Object Detection. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 307–324.
40. Shrivastava, A.; Gupta, A.K.; Girshick, R.B. Training Region-Based Object Detectors with Online Hard Example Mining. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 761–769.
41. Wan, S.; Chen, Z.; Tao, Z.; Zhang, B.; kat Wong, K. Bootstrapping Face Detection with Hard Negative Examples. *arXiv* **2016**, arXiv:1608.02236.
42. Liu, Y. An Improved Faster R-CNN for Object Detection. In Proceedings of the 2018 11th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 8–9 December 2018; Volume 2, pp. 119–123.
43. Sun, X.; Wu, P.; Hoi, S.C. Face detection using deep learning: An improved faster RCNN approach. *Neurocomputing* **2018**, *299*, 42–50. [[CrossRef](#)]
44. Yang, S.; Luo, P.; Loy, C.C.; Tang, X. WIDER FACE: A Face Detection Benchmark. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 5525–5533.
45. Zhang, L.; Lin, L.; Liang, X.; He, K. Is Faster R-CNN Doing Well for Pedestrian Detection? *arXiv* **2016**, arXiv:1607.07032.
46. Wang, X.; Shrivastava, A.; Gupta, A.K. A-Fast-RCNN: Hard Positive Generation via Adversary for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 3039–3048.
47. Ravi, N.; El-Sharkawy, M. Improved Single Shot Detector with Enhanced Hard Negative Mining Approach. In Proceedings of the 2022 International Conference on Advanced Computer Science and Information Systems (ICACSIS), Depok, Indonesia, 1–3 October 2022; pp. 25–30.
48. Albetis, J.; Jacquin, A.; Goulard, M.; Poilvé, H.; Rousseau, J.; Clenet, H.; Dedieu, G.; Duthoit, S. On the potentiality of UAV multispectral imagery to detect Flavescence dorée and Grapevine Trunk Diseases. *Remote Sens.* **2018**, *11*, 23. [[CrossRef](#)]

49. Vélez, S.; Ariza-Sentís, M.; Valente, J. Mapping the spatial variability of Botrytis bunch rot risk in vineyards using UAV multispectral imagery. *Eur. J. Agron.* **2023**, *142*, 126691. [[CrossRef](#)]
50. 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](#)]
51. Šandric, I.; Irimia, R.; Petropoulos, G.P.; Anand, A.; Srivastava, P.K.; Pleșoianu, A.; Faraslis, I.; Stateras, D.; Kalivas, D. Tree's detection & health's assessment from ultra-high resolution UAV imagery and deep learning. *Geocarto Int.* **2022**, *37*, 10459–10479.
52. Solano, F.; Di Fazio, S.; Modica, G. A methodology based on GEOBIA and WorldView-3 imagery to derive vegetation indices at tree crown detail in olive orchards. *Int. J. Appl. Earth Obs. Geoinf.* **2019**, *83*, 101912. [[CrossRef](#)]
53. Garcia-Ruiz, F.; Sankaran, S.; Maja, J.M.; Lee, W.S.; Rasmussen, J.; Ehsani, R. Comparison of two aerial imaging platforms for identification of Huanglongbing-infected citrus trees. *Comput. Electron. Agric.* **2013**, *91*, 106–115. [[CrossRef](#)]
54. Navrozidis, I.; Haugomard, A.; Kasampalis, D.; Alexandridis, T.; Castel, F.; Moshou, D.; Ovakoglou, G.; Pantazi, X.E.; Tamouridou, A.A.; Lagopodi, A.L.; et al. Assessing Olive Trees Health Using Vegetation Indices and Mundi Web Services for Sentinel-2 Images. In Proceedings of the Hellenic Association on Information and Communication Technologies in Agriculture, Food & Environment, Thessaloniki, Greece, 24–27 September 2020; pp. 130–136.
55. 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.
56. Zarco-Tejada, P.J.; Miller, J.R.; Mohammed, G.; Noland, T.L.; Sampson, P. Vegetation stress detection through chlorophyll a + b estimation and fluorescence effects on hyperspectral imagery. *J. Environ. Qual.* **2002**, *31*, 1433–1441. [[CrossRef](#)]
57. Barry, K.M.; Stone, C.; Mohammed, C. Crown-scale evaluation of spectral indices for defoliated and discoloured eucalypts. *Int. J. Remote Sens.* **2008**, *29*, 47–69. [[CrossRef](#)]
58. Redmon, J.; Divvala, S.K.; Girshick, R.B.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
59. Weinstein, B.G.; Marconi, S.; Aubry-Kientz, M.; Vincent, G.; Senyondo, H.; White, E.P. DeepForest: A Python package for RGB deep learning tree crown delineation. *Methods Ecol. Evol.* **2020**, *11*, 1743–1751. [[CrossRef](#)]
60. Weinstein, B.G.; Marconi, S.; Bohlman, S.A.; Zare, A.; Singh, A.; Graves, S.J.; White, E.P. A remote sensing derived data set of 100 million individual tree crowns for the National Ecological Observatory Network. *eLife* **2021**, *10*, e62922. [[CrossRef](#)]
61. Kobayashi, N.; Tani, H.; Wang, X.; Sonobe, R. Crop classification using spectral indices derived from Sentinel-2A imagery. *J. Inf. Telecommun.* **2020**, *4*, 67–90. [[CrossRef](#)]
62. Dash, J.P.; Watt, M.S.; Pearce, G.D.; Heaphy, M.; Dungey, H.S. Assessing very high resolution UAV imagery for monitoring forest health during a simulated disease outbreak. *ISPRS J. Photogramm. Remote Sens.* **2017**, *131*, 1–14. [[CrossRef](#)]
63. Cogato, A.; Pagay, V.; Marinello, F.; Meggio, F.; Grace, P.; De Antoni Migliorati, M. Assessing the feasibility of using sentinel-2 imagery to quantify the impact of heatwaves on irrigated vineyards. *Remote Sens.* **2019**, *11*, 2869. [[CrossRef](#)]
64. Hawryło, P.; Bednarz, B.; Weżyk, P.; Szostak, M. Estimating defoliation of Scots pine stands using machine learning methods and vegetation indices of Sentinel-2. *Eur. J. Remote Sens.* **2018**, *51*, 194–204. [[CrossRef](#)]
65. Oumar, Z.; Mutanga, O. Using WorldView-2 bands and indices to predict bronze bug (*Thaumastocoris peregrinus*) damage in plantation forests. *Int. J. Remote Sens.* **2013**, *34*, 2236–2249. [[CrossRef](#)]
66. Verbesselt, J.; Robinson, A.; Stone, C.; Culvenor, D. Forecasting tree mortality using change metrics derived from MODIS satellite data. *For. Ecol. Manag.* **2009**, *258*, 1166–1173. [[CrossRef](#)]
67. Datt, B. Remote sensing of chlorophyll a, chlorophyll b, chlorophyll a + b, and total carotenoid content in eucalyptus leaves. *Remote Sens. Environ.* **1998**, *66*, 111–121. [[CrossRef](#)]
68. Deng, X.; Guo, S.; Sun, L.; Chen, J. Identification of short-rotation eucalyptus plantation at large scale using multi-satellite imageries and cloud computing platform. *Remote Sens.* **2020**, *12*, 2153. [[CrossRef](#)]
69. Bajwa, S.G.; Tian, L. Multispectral CIR image calibration for cloud shadow and soil background influence using intensity normalization. *Appl. Eng. Agric.* **2002**, *18*, 627. [[CrossRef](#)]
70. Tucker, C.J. Red and photographic infrared linear combinations for monitoring vegetation. *Remote Sens. Environ.* **1979**, *8*, 127–150. [[CrossRef](#)]
71. Sripada, R.P.; Heiniger, R.W.; White, J.G.; Meijer, A.D. Aerial color infrared photography for determining early in-season nitrogen requirements in corn. *Agron. J.* **2006**, *98*, 968–977. [[CrossRef](#)]
72. Buschmann, C.; Nagel, E. In vivo spectroscopy and internal optics of leaves as basis for remote sensing of vegetation. *Int. J. Remote Sens.* **1993**, *14*, 711–722. [[CrossRef](#)]
73. Villa, P.; Mousivand, A.; Bresciani, M. Aquatic vegetation indices assessment through radiative transfer modeling and linear mixture simulation. *Int. J. Appl. Earth Obs. Geoinf.* **2014**, *30*, 113–127. [[CrossRef](#)]
74. Barnes, E.; Clarke, T.; Richards, S.; Colaizzi, P.; Haberland, J.; Kostrzewski, M.; Waller, P.; Choi, C.; Riley, E.; Thompson, T.; et al. Coincident detection of crop water stress, nitrogen status and canopy density using ground based multispectral data. In Proceedings of the Fifth International Conference on Precision Agriculture, Bloomington, MN, USA, 16–19 July 2000; Volume 1619, p. 6.
75. Rouse, J.W.; Haas, R.H.; Schell, J.A.; Deering, D.W. Monitoring vegetation systems in the Great Plains with ERTS. *NASA Spec. Publ.* **1974**, *351*, 309.

76. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [[CrossRef](#)]
77. Loh, W.Y. Classification and regression trees. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2011**, *1*, 14–23. [[CrossRef](#)]
78. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]
79. Cover, T.; Hart, P. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **1967**, *13*, 21–27. [[CrossRef](#)]
80. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.Y. Lightgbm: A highly efficient gradient boosting decision tree. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 3149–3157.
81. Puliti, S.; Astrup, R. Automatic detection of snow breakage at single tree level using YOLOv5 applied to UAV imagery. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *112*, 102946. [[CrossRef](#)]
82. Nembrini, S.; König, I.R.; Wright, M.N. The revival of the Gini importance? *Bioinformatics* **2018**, *34*, 3711–3718. [[CrossRef](#)] [[PubMed](#)]
83. Mohan, M.; Silva, C.A.; Klauber, C.; Jat, P.; Catts, G.; Cardil, A.; Hudak, A.T.; Dia, M. Individual tree detection from unmanned aerial vehicle (UAV) derived canopy height model in an open canopy mixed conifer forest. *Forests* **2017**, *8*, 340. [[CrossRef](#)]
84. Zhang, N.; Wang, Y.; Zhang, X. Extraction of tree crowns damaged by *Dendrolimus tabulaeformis* Tsai et Liu via spectral-spatial classification using UAV-based hyperspectral images. *Plant Methods* **2020**, *16*, 1–19. [[CrossRef](#)]
85. Iordache, M.D.; Mantas, V.; Baltazar, E.; Pauly, K.; Lewycky, N. A machine learning approach to detecting pine wilt disease using airborne spectral imagery. *Remote Sens.* **2020**, *12*, 2280. [[CrossRef](#)]
86. Ortiz, S.M.; Breidenbach, J.; Kändler, G. Early detection of bark beetle green attack using TerraSAR-X and RapidEye data. *Remote Sens.* **2013**, *5*, 1912–1931. [[CrossRef](#)]
87. Gago, J.; Douthe, C.; Coopman, R.E.; Gallego, P.P.; Ribas-Carbo, M.; Flexas, J.; Escalona, J.; Medrano, H. UAVs challenge to assess water stress for sustainable agriculture. *Agric. Water Manag.* **2015**, *153*, 9–19. [[CrossRef](#)]
88. Berni, J.; Zarco-Tejada, P.; Sepulcre-Cantó, G.; Fereres, E.; Villalobos, F. Mapping canopy conductance and CWSI in olive orchards using high resolution thermal remote sensing imagery. *Remote Sens. Environ.* **2009**, *113*, 2380–2388. [[CrossRef](#)]
89. Moriondo, M.; Maselli, F.; Bindi, M. A simple model of regional wheat yield based on NDVI data. *Eur. J. Agron.* **2007**, *26*, 266–274. [[CrossRef](#)]
90. Yang, M.; Hassan, M.A.; Xu, K.; Zheng, C.; Rasheed, A.; Zhang, Y.; Jin, X.; Xia, X.; Xiao, Y.; He, Z. Assessment of water and nitrogen use efficiencies through UAV-based multispectral phenotyping in winter wheat. *Front. Plant Sci.* **2020**, *11*, 927. [[CrossRef](#)]
91. Adão, T.; Hruška, J.; Pádua, L.; Bessa, J.; Peres, E.; Morais, R.; Sousa, J.J. Hyperspectral imaging: A review on UAV-based sensors, data processing and applications for agriculture and forestry. *Remote Sens.* **2017**, *9*, 1110. [[CrossRef](#)]
92. Michez, A.; Piégay, H.; Lisein, J.; Claessens, H.; Lejeune, P. Classification of riparian forest species and health condition using multi-temporal and hyperspatial imagery from unmanned aerial system. *Environ. Monit. Assess.* **2016**, *188*, 1–19. [[CrossRef](#)] [[PubMed](#)]
93. Smigaj, M.; Gaulton, R.; Barr, S.L.; Suárez, J.C. UAV-borne Thermal Imaging for Forest Health Monitoring: Detection of Disease-Induced Canopy Temperature Increase. *ISPRS Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2015**, *XL-3/W3*, 349–354. [[CrossRef](#)]
94. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
95. Lin, T.Y.; Dollár, P.; Girshick, R.B.; He, K.; Hariharan, B.; Belongie, S.J. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944.
96. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
97. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [[CrossRef](#)] [[PubMed](#)]

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.