Orchard Apple Tree Health Assessment using UAV Imagery-Based Computer Vision System

Hela Jemaa

A Thesis

in

The Department

of

Concordia Institute for Information Systems Engineering (CIISE)

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Applied Science (Quality Systems Engineering) at

Concordia University

Montréal, Québec, Canada

June 2023

© Hela Jemaa, 2023

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Hela Jemaa**

Entitled: **Orchard Apple Tree Health Assessment using UAV Imagery-Based Computer Vision System**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Quality Systems Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

______________________ Chair and Examiner
Dr. Abdessamad Ben Hamza

______________________ Examiner
Dr. Manar Amayri

______________________ Supervisor
Dr. Wassim Bouachir

______________________ Supervisor
Dr. Brigitte Leblon

______________________ Supervisor
Dr. Nizar Bouguila

Approved by ______________________
Dr. Abdessamad Ben Hamza, Chair
Department of Concordia Institute for Information Systems Engineering (CIISE)

__________ 2023

____
Mourad Debbabi, Dean
Faculty of Engineering and Computer Science

Abstract

Orchard Apple Tree Health Assessment using UAV Imagery-Based Computer Vision System

Hela Jemaa

Accurate and efficient orchard tree inventories play a crucial role in obtaining up-to-date information for effective treatments and crop insurance purposes. Surveying orchard trees, including counting, locating, and assessing their health status, is vital for predicting production volumes and facilitating orchard management. However, traditional manual inventories are labor-intensive, expensive, and prone to errors. Motivated by the recent advances in UAV imagery and computer vision methods, we propose a new framework for individual tree detection and health assessment. The proposed approach follows a two-stage process. First, we build a tree detection model based on a hard negative mining strategy using RGB UAV images. In the second stage, we address the health classification problem using two methods. We present a classical machine learning approach by exploring the use of multi-band imagery-derived vegetation indices. We also propose a new convolutional autoencoder-based architecture mainly designed to extract the relevant features for tree health classification.

The performed experiments demonstrate the robustness of the proposed framework for orchard tree health assessment from UAV images. In particular, our framework achieves an F1-score of 86.24% for tree detection and an overall accuracy of 98.06% for tree health assessment. Moreover, our work could be generalized for a wide range of UAV applications involving a detection/classification process.

Acknowledgments

I would like to express my sincere gratitude to everyone who has supported me throughout my thesis journey. First, I would like to thank my supervisors, Prof. Nizar Bouguila, Prof. Wassim Bouachir, and Prof. Brigitte Leblon, for their guidance, advice, and patience. Their expertise and insights have been essential in shaping my research and improving the quality of my work. I am also thankful to Prof. Armand LaRocque for his assistance with data collection, analysis, and proofreading. Without their help, this thesis would not have been possible. Additionally, I would like to express my heartfelt appreciation to my family and friends for their encouragement, understanding, and unwavering support throughout this challenging period. Their love and faith in me have been a constant source of inspiration and motivation. Finally, I would like to acknowledge the contributions of all the participants who generously shared their time and experience with me. Your insights and perspectives have enriched my research and made it more meaningful.

This thesis was partially funded by an NSERC CRD grant awarded to Prof. Brigitte Leblon and sponsored by A&L Canada Labs Inc. A special thanks to Dr. Ata Haddadi from A&L Canada Labs Inc. and the field crew made of Odysseas Vlachopoulos, Anthony Bastarache, Nafsika Papageorgiou, Timothy Roberts, and the UAV pilot Brennan Gaudet.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context and research problem

Orchard tree inventory has been an essential step in obtaining up-to-date information for effective tree treatments and crop insurance purposes. Therefore, surveying orchard trees, including counting their numbers and determining their locations, pattern, and health status is important for predicting production volumes and for the purpose of plantation management.

Tree diseases have long been a source of concern for the agricultural industry due to their considerable and negative impact on orchard quality and productivity. Unhealthy or stressed orchard trees are more susceptible to pests, diseases, and environmental stressors, which can reduce the yield and quality of fruit, leading 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 more informed decisions about practices such as irrigation, fertilization, and pest control. This can help optimize orchard yield and quality while minimizing the use of inputs, reducing the risk of orchard loss, and improving 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 such as tree position, total height, crown diameter, leaf area index [1], and chlorophyll content [2]). 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) have provided new tools that offer an alternative to traditional methods. Deep neural networks (DNNs) [3], on the other hand, have emerged as a powerful tool in the field of computer vision. The high spatial resolution provided by UAV images combined with computer vision algorithms have made tremendous advances in several domains such as forestry [4], agriculture [5], geology [6], surveillance [7], and traffic monitoring [8].

Motivated by the latest advances in computer vision systems, we propose a new framework to automatically detect orchard apple trees and assess their health from UAV multi-band imagery.

1.2 Method overview

The objective of this work is to develop an automated framework for tree health assessment using UAV images. The proposed framework adopts a divide-and-conquer strategy to solve two sub-problems:

1. Tree localization problem using hard negative mining strategy based on deep learning detection models,

2. Tree health assessment through image patch classification using two different methods:

    - Traditional machine learning method by exploiting the use of vegetation indices.

    - Deep learning-based method using a new discriminant semi-supervised autoencoder.

In the field of **object detection**, extensive studies have been conducted on aerial imagery applications [9, 10, 11, 12, 13, 14]. Most of the proposed methods have adopted approaches based on fine-tuning networks that have been pre-trained on large-scale image datasets such as ImageNet [15] and MS-COCO [16] for detection in the UAV domain. However, applying these methods to UAV images has particular challenges 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 [17], 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.

In order to improve the robustness of the model against complex backgrounds, we suggest a dedicated detection procedure employing a hard negative mining strategy [18]. This approach enables the model to focus on hard negative examples during training, thereby reducing detection errors resulting from complex backgrounds.

Following the detection stage, we formulate the health status assessment of each detected tree as a **classification** task of tree image patches. With visual classification, a key challenge is to find the relevant feature representation that best describes the data in the feature space. In our work, we explored the use of two types of features: Vegetation index-derived features and deep image features by using autoencoders.

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 [19, 20, 21, 22, 23] that by analyzing these indices, we can gain insights into the health and vitality of trees.

Traditional autoencoders [24, 25, 26], on the other hand, have been studied and used for feature extraction tasks. However, because standard autoencoders generally do not use label information, the extracted representation may be limited in handling discriminative tasks. To address the feature relevance problem, we instead propose a semi-supervised discriminant convolutional autoencoder, where the encoder part is trained on both labeled and unlabeled data to learn a compact and relevant representation of input data. This approach allows the classification network to exploit both labeled and unlabeled data to learn a more robust feature representation for classification.

1.3 Contributions

The main contributions of our work are summarized as follows:

1. **Contributions to the precision agriculture field:** This study focuses on a crucial task in precision agriculture. We propose a novel framework for automatic tree detection and health assessment from UAV images, according to a divide-and-conquer approach. The proposed framework could be generalized for a wide range of other UAV applications involving a detection/classification process.

2. **Contributions to the research community:** This study stimulates the interest of the scientific community and contributes to advancing knowledge in the field of object detection and classification.

   - We adopt a hard negative mining approach for tree detection to address the problem of negative hard-easy examples imbalance.
   - We present two methods to address the tree health assessment problem. The first method is a classical ML approach based on VIs, and the second method is a deep learning-based approach using a semi-supervised convolutional autoencoder, taking advantage of both labeled and unlabeled data, to ensure the relevance of feature representation for the health classification task.

   In this direction, our scientific contributions have been documented in the following papers:

   - Hela Jemaa, Wassim Bouachir, Brigitte Leblon, and Nizar Bouguila. Computer vision system for detecting orchard trees from UAV images. ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 43:661–668, 2022, **published**.
   - Hela Jemaa, Wassim Bouachir, Brigitte Leblon, Armand LaRocque, Ata Haddadi, and Nizar Bouguila. Tree health assessment from UAV images: Improving object detection and classification using hard negative mining and semi-supervised autoencoder. In 2023 20th Conference on Robots and Vision (CRV). IEEE, 2023, **accepted**.

- Hela Jemaa, Wassim Bouachir, Brigitte Leblon, Armand LaRocque, Ata Haddadi, and Nizar Bouguila. UAV-based computer vision system for orchard apple tree detection and health assessment. Remote Sensing, 2023, **submitted**.

1.4 Thesis outline

The rest of the thesis is structured as follows. Chapter 2 introduces background concepts and related works on orchard tree health assessment, with a focus on UAV images. In addition, we discuss the scientific literature describing the approaches used in our work to address our problem. Chapter 3 provides a detailed description of the proposed framework. The experimental results are presented and discussed in Chapter 4. Finally, Chapter 5 concludes the thesis and suggests directions for future work.

Chapter 2

Literature Review

Tree health assessment has been a concern for growers for centuries, and over time, a range of methods have been developed for this purpose. Initially, visual inspection [27] was the most common method used by arborists to identify signs of damage or disease. Later, tapping, microscopy, and chemical analysis [28] were developed as additional methods to assist in the detection of tree health problems. More recently, remote sensing technologies have been introduced, enabling the assessment of tree health from distance. This includes the use of aerial photography, satellite imagery, and drones equipped with specialized sensors to detect changes in vegetation or temperature variations that may indicate tree stress or disease.

In the current era, with rapid advancements in technology, numerous methods are available for tree health assessment. In this chapter, a literature review is conducted. It reviews the related works on tree detection and health classification, with a focus on UAV images. In addition, we discuss the scientific literature related to the adopted techniques, such as the hard negative mining learning startegy and autoencoders approach.

2.1 Tree detection

2.1.1 Traditional approaches

Classical object detection methods often involve the utilization of handcrafted features and machine learning algorithms. Local binary pattern (LBP) [29], scale-invariant feature transform (SIFT)

[30, 31], and histogram of oriented gradients (HOG) [32, 33] are the most frequently used hand-crafted features in object detection. For example, the work in [34] 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 [35, 36], 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, [37] employed 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 an SVM to recognize palm tree images from background regions. The study in [38] 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 local maximum detection algorithm based on the spatial distribution of standardized features to detect palm trees. The work in [5] presented a method to detect apple trees using multispectral UAV images. The authors identified trees using thresholding techniques applied on Normalized Difference Vegetation Index (NDVI) and entropy images, as trees are chlorophyllous bodies that have high NDVI values and are heterogeneous with high entropy. Similarly, [39] presented a detection strategy that finds the regions that are more likely to contain trees. The method is mainly based on a thresholding technique applied on RGB images. In [40], the authors described the algorithm that was used to detect treetops using normalized digital surface model (nDSM) data. A sliding window was used to process small parts of the nDSM. At each window position, they apply thresholding techniques to only consider the pixels with higher intensities compared to a set threshold. Then, a refinement procedure is applied to eliminate those that are too close to each other. The work in [41] proposed a tree crown delineation algorithm using RGB images. It goes through three main stages: Preprocessing step for contrast enhancement, crown segmentation based on wavelet transformation

and morphological operations, and boundary detection. The work in [42] 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 [43], 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.

Overall, the traditional object detection methods manually extract low-level features. However, these features are not competent to perform the task of object detection in images with complex backgrounds and noise. Furthermore, it is a difficult task to manually design a robust and efficient feature for object detection; a great deal of particular knowledge is needed, and tremendous parameters need to be tuned carefully.

2.1.2 Deep-Learning approaches

Lately, deep learning (DL) methods have emerged as they surpassed classical Machine Learning (ML) algorithms in a large spectrum of computer vision applications. Deep learning methods have demonstrated their ability to extract robust semantic information from massive datasets, allowing them to manage many tough conditions such as scale change, rotation, and appearance variation, thanks to their deep and sophisticated architectures. Convolutional Neural Networks (CNN) are the most popular deep-learning models that have the capability to extract millions of high-level features that can be used effectively for object detection.

Several studies explored various deep-learning algorithms to detect trees over UAV imagery. For instance, [44] 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 generated 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 [45], the authors adopted a sliding window approach for oil palm tree detection. A sliding window is integrated with a pre-trained AlexNet classifier to scan the input image and identify regions that contain trees. Most of the explored methods involved fine-tuning pre-trained object detection networks that were trained on large-scale image datasets such as ImageNet [15] and MSCOCO [16]. For example, The work in [46] exploit 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. In [14], the authors explored the use of the Yolo version-5 with its subversions and DeepForest for the detection of orchard apple trees. In [47], 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) [48], YOLOv3 [49], and RetinaNet. Similarly, the work in [50] explored the use of Faster R-CNN, Single Shot Multi-Box Detector (SSD) [51], and R-FCN [52] architectures to detect seedlings.

Other works perform tree detection as a segmentation task. As an illustration, in [53], the authors developed a method to detect Amazonian palm trees. They incorporated ResNet-18 in the DeepLabv3+ architecture for semantic image segmentation to obtain score maps for each tree species. Then, they applied morphological operations at the score maps to separate overlapped trees. In the work of [54], the authors explored the use of Mask Region-based Convolutional Neural Network (Mask R-CNN) using Blue, Green, Red, Red Edge, NIR, and canopy height model (CHM) bands separately and RGB combination. The explored model with the multi-band combination produced higher accuracy than the model with a single-band image, and the RGB band combination achieved the highest accuracy. Similarly, the work in [55] addressed the tree detection task using the Mask R-CNN algorithm. Six different band combinations and derivations of the UAV imagery were used to detect tree crown and height respectively (Multi band-DSM, RGB-DSM, NDVI-DSM, Multi band-CHM, RGB-CHM, and NDVI-CHM combination). The Mask R-CNN model with the NDVI-CHM combination achieved superior performance.

This analysis of the various methodologies reveals that CNN-based techniques are becoming more prevalent in tree detection.

2.2 Tree health classification

2.2.1 Traditional approches

The work in [56] introduces a spectral-spatial classification framework to automatically extract tree crowns damaged by Dendrolimus tabulaeformis. The proposed framework combines a support vector machine (SVM) with an edge-preserving flter (EPF). The SVM pixel-wise spectral classifier is applied to the hyperspectral image to produce a probability map. Then, the EPF is applied to extract the spatial features using RGB images. The work in [22] studied pine wild disease using a Random Forest classifier applied to both multispectral and hyperspectral imagery. In [41], the authors propose an approach for the assessment of the vital status of forest stands using multispectral and RGB images. They employ an ensemble algorithm known as error-correcting output codes (ECOC) combined with SVM to carry out pixel-based classification of images by spectral features.

Vegetation indices are also explored in different works for tree health assessment. For example, the study in [57] proposes a framework for detecting and quantifying Eucalyptus Longhorned Borers (ELB) damages in eucalyptus stands. Treetops are calculated using the local maxima filter of a sliding window algorithm. Afterward, large-scale mean-shift segmentation is performed to extract the crowns. Five selected vegetation indices, including the difference vegetation index (DVI), normalized difference vegetation index (NDVI), green normalized vegetation index (GNDVI), normalized difference red-edge (NDRE), and soil adjusted vegetation index (SAVI), are calculated to perform classification using random forest. Similarly, the work in [19] uses a modified NDVI to discriminate between five classes of infestation by the oak splendor beetle through object-based imagery analysis (OBIA) classification. NDVI was also explored in combination with RGB, near-infrared (NIR), and red-edge bands in [20] to investigate ark beetle damage at the tree level using the K-nearest neighbor (KNN) classifier. Likewise, NDVI and red-edge NDVI were used in [21] to study the discoloration classes of Pinus radiata through an OBIA classification approach with a random forest algorithm. The work in [23] detects the defoliation of pine trees affected by pine processionary and distinguishes pine species at a pixel level using four spectral indices and the NIR band. The authors estimate the threshold values using histogram analysis. The work in [58] presents a method for the identification of stress in olive trees. It uses a support vector machine (SVM) model applied to VIs to classify each tree pixel into one of two categories: healthy and stressed. In [21], the authors conducted a multi-temporal analysis of disease outbreaks in mature Pinus radiata D. Don trees. They explored the use of random forest with four spectral indices, including NDVI, GNDVI, NDRE, and non linear index (NLI).

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

2.2.2 Deep learning approaches

The tree health classification problem is addressed by some works using deep learning approach. For example, The work in [39] developed a new CNN architecture that predicts damage stages of fir trees infected by the Bark Beetle using RGB images. The classification model is applied to candidate regions chosen by a detection strategy. In [59], the authors explored the use of Fatser R-CNN combined with high-resolution RGB imagery for automatic detection and health classification of oil palm trees. In [40], the authors evaluated a series of deep learning classifiers for individual sick fir tree (Abies mariesii) identification using RGB images. They explored the use of Alexnet [60], Squeezenet [61], VGG [62], Resnet [63], and Densenet [64]. Similarly, [65] explored the potential use of VGG-16 to map tree health status using two different channel combinations, including RGB and IRG (Infra-red+red+green). The work in [66] designed a novel framework, named Soybean Network (SNet) for assessment of damage conditions in soybean seeds. The proposed network is mainly composed of deep separable convolution [67] instead of standard convolution layers. The study compared the proposed SNet with six state-of-the-art models, including three classical recognition models: Inception [68], ResNet50 [63], and ResNeXt50 [69]; three lightweight models: MobileNet [70], ShuffleNet [71], and EfficientNet [72].

2.3 Hard negative mining learning strategy

Object detectors often face the issue of data imbalance, where training set is distinguished by a large imbalance between the number of annotated objects and the number of background examples, image regions not belonging to any object class of interest.

To mitigate this issue, hard negative mining (HNM) can be adopted for object detection. This technique originally known as bootstrapping, now commonly referred to as hard negative mining, has been first introduced in [73]. The authors presented bootstrapping as a method for training face detection models. Their main concept was to gradually expand the set of background examples by selecting those instances that triggered false alarms in the detector. This approach resulted in an iterative training algorithm that alternates between updating the detection model based on the current set of examples and utilizing the updated model to identify new false positives for inclusion in the bootstrapped training set. Typically, the process starts with a training set comprising all object examples along with a small, random selection of background examples. Recently, Various HNM approaches have been adopted. For example, In [74], the authors introduced an online hard example mining (OHEM) procedure based on a training process that uses a data sampling strategy favoring high-loss region proposals. This technique, originally applied to the Fast R-CNN detector, yielded significant gains on the PASCAL and MS-COCO benchmarks. [75] involve iteratively bootstrapping a small set of negative examples, by selecting those that trigger a false positive alarm in the detector. The work in [18] 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), which is the de facto standard for evaluating face detection algorithms. Similarly, an improved version of faster R-CNN is proposed in [76], by using hard negative sample mining for object detection. Likewise, [77] used the bootstrapping of hard negatives to improve the performance of CNN-based detectors. The authors pre-trained Faster R-CNN to mine hard negatives, before retraining the model. The work in [78] presented a cascaded Boosted Forest, which performs effective hard negative mining and sample reweighting, to classify the region proposals generated by RPN. In [79], the authors proposed a deep learning-based object detection framework based on a hard-example-mining network (HEMN) to make the detection network focus on hard examples during the training phase by changing the input data distribution of each stage. The A-Fast-RCNN method, described in [80], adopts a different approach for generating hard negative samples, by using occlusion and spatial deformations through an adversarial process.

Another approach to apply HNM using Single Shot multi-box Detector (SSD) is proposed in [81], where the authors use medium priors, anchor boxes with 20% to 50% overlap with ground truth boxes, to enhance object detector performance. 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 apply the HNM approch specifically for tree detection.

2.4 Semi-supervised autoencoder

Autoencoders and their extensions [25, 26, 82] have been important in various fields such as computer vision, and natural language processing (NLP). A standard Autoencoder is an unsupervised model made up of an encoder network that maps the input data to a latent space representation, and a decoder network mapping the latent space representation back to the original input data. The goal is to learn a compact representation capturing the most important features, by minimizing the reconstruction loss. However, representations extracted by conventional autoencoders may not be useful for discriminative tasks [83], as label information is not used by unsupervised autoencoders.

There are some novel semi-supervised autoencoders that are able to learn from both labeled and unlabeled data [84, 85, 86]. The basic idea is to use the autoencoder to learn an efficient representation of the input data, and then use the learned representation for training on a specific task using labeled data. Semi-supervised autoencoders have been applied in various domains, including computer vision and NLP. For example, the work in [84] introduces a semi-supervised version of the Variational Autoencoder (VAE) model for image classification. The authors use a VAE to learn a latent representation of the input images, and then use a classifier to predict the label of the image based on the latent representation. The model is trained on both labeled and unlabeled data, with the objective of maximizing the log-likelihood of the labeled data and the marginal likelihood of the unlabeled data. The authors demonstrate that their model outperforms existing semi-supervised methods on the MNIST dataset. Similarly, a dual-objective framework is presented in [86] for feature extraction in fault diagnosis. The work in [85] proposes a semi-supervised variant of the Generative Adversarial Network (GAN) model for image classification. A GAN is used to learn a generator network and a discriminator network. The network is used to generate images and the discriminator network is used to predict the label of the image. The model is trained on both labeled and unlabeled data, with the objective of minimizing the cross-entropy loss of the labeled data and the Wasserstein distance of the unlabeled data.

For our tree health classification stage, we adopt a semi-supervised autoencoder using both labeled and unlabeled data to provide relevant features that best determine the tree damage status. Our proposed autoencoder-based solution uses a dual-objective framework, where the autoencoder and the classifier are jointly trained to optimize both reconstruction and classification objectives.

Chapter 3

Materials and methods

3.1 Study area and materials

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 3.1.

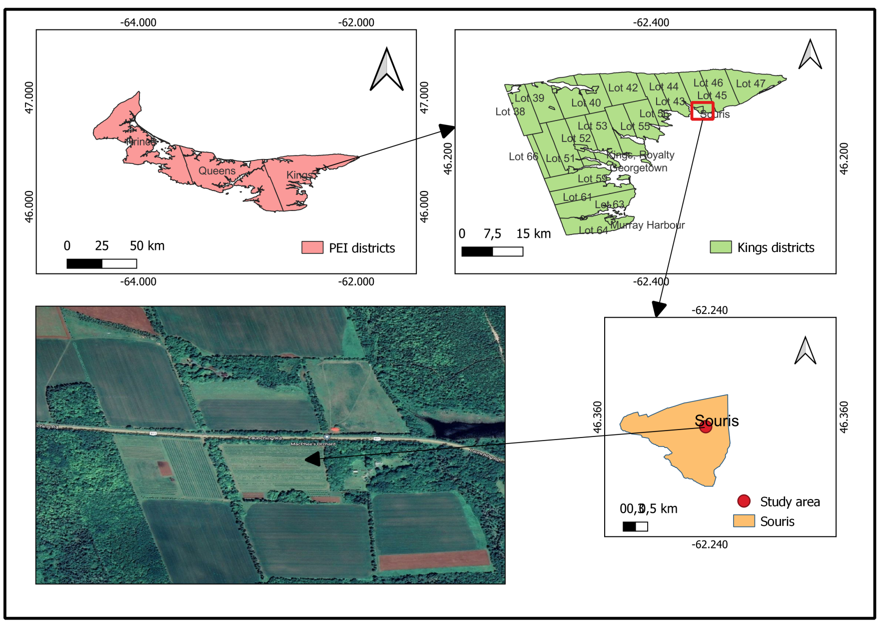

Figure 3.1: Location of the study area.

The UAV images were taken using a MicaSense multispectral camera (MicaSense Inc., Seattle, U.S.A.). 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 UAV that has been developed by A&L Canada. Its weight is slightly less than 2.0 kg. Both the camera and the UAV were connected to mission planner software 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.

In general, UAV images have high data volume given that they can cover dozens of hectares. Due to the huge volume, complexity, and computational cost, we adopted a patch-based approach to split the orchard orthomosaic and create non-overlapped patches with regular pixel sizes of 512×512. The patch size is chosen to fit our framework input.

In order to prepare the data for object detection, we performed manual annotation for patches, which consists in localizing our objects of interest (apple trees) manually through bounding boxes using the open-source VGG Image Annotator (VIA) tool [87]. Then, the annotated patches are split into three subsets: training, validation, and testing, using 3-fold cross-validation to assure an impartial evaluation of object detection models.

Using fieldwork observations, trees were examined to estimate their health status. Two different tree health status were assigned: healthy and unhealthy. Figure 3.2 illustrates two instances of trees captured at ground level: one depicting a healthy tree and the other representing an unhealthy tree.

Figure 3.2: 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 2,828 trees in total, where 2,240 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.

3.2 Framework overview

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 3.3, follows a two-stage approach to sequentially address two tasks: tree detection and tree health classification.

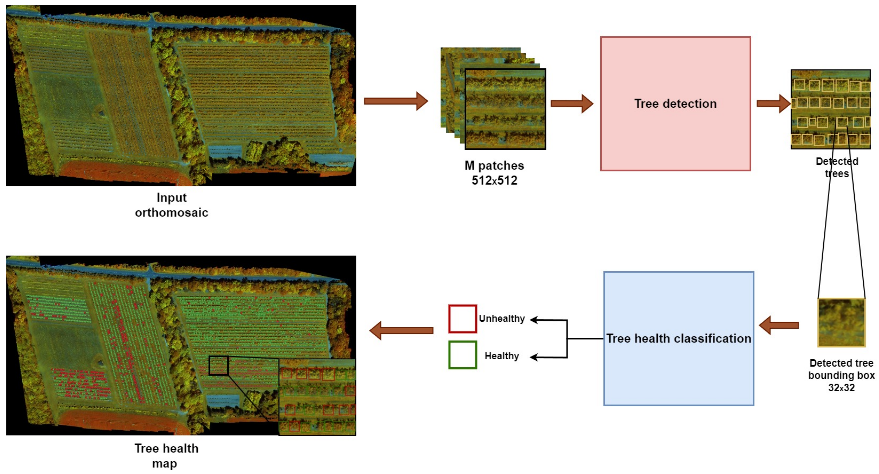

Figure 3.3: Overview of the proposed framework. In the health tree map, green rectangles correspond to healthy trees and red boxes correspond to stressed trees. The images are displayed in false color (near-infrared (NIR), red edge (RE), red (R)).

In the first stage, the input orthomosaic is subdivided into multiple patches of 512 by 512 pixels each. 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.

By combining the tree localization results from the tree detection module with the corresponding health status obtained through tree classification, a health map can be generated. It provides valuable information to orchard growers, enabling them to implement targeted management strategies for their orchards based on the health status of individual trees.

3.3 Tree detection

The task of tree detection is challenging due to the presence of complex backgrounds that may distract object detectors. As depicted in Figure 3.4, the color, shape, and texture of some objects belonging to the background (red rectangles) are visually similar to the target tree objects (blue rectangles), leading to false positives. To overcome this challenge, we adopted a hard negative mining approach to improve the detector's ability to discriminate between trees and background regions. By iteratively introducing false detections as hard negative examples during training, the detector can learn to better differentiate between the objects of interest and background regions.

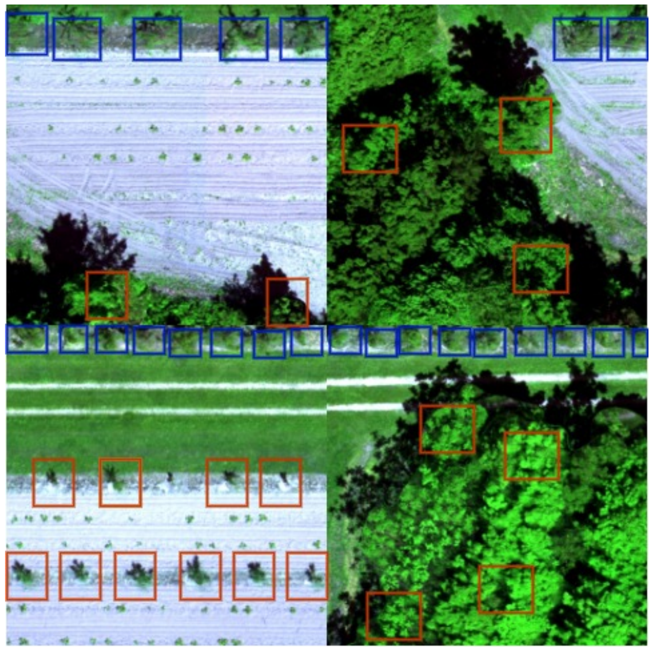

Figure 3.4: Detection results of a baseline object detector without the use of hard negative mining. Red rectangles are false detections (FP) and blue rectangles are correct detections (TP).

The proposed method for tree detection, depicted in Figure 3.5, 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, corresponding to objects incorrectly detected as trees. In the object detection task, a detection is considered 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. The harvested hard negatives are then used to introduce 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 that have 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 [88] 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. The process of mining hard negatives and fine-tuning is performed iteratively, by continually refining the training set, which gradually improves the tree detection accuracy.

In our work, we adopt two state-of-the-art object detector baselines: YOLO [89] using its COCO pre-trained model and DeepForest prebuilt model [90] that is trained on The National Ecological Observatory Network (NEON [91]) crowns data set. The adopted baseline architectures are explained further in the Appendix A.1. 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.

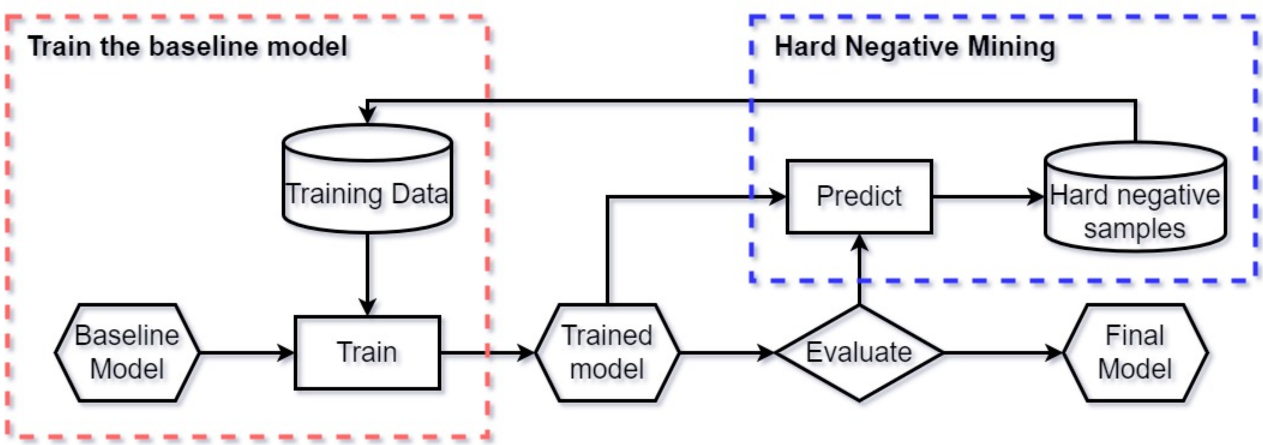

Figure 3.5: Tree Detection using hard negative mining approach.

The steps of the proposed tree detection algorithm are outlined in the pseudo-code shown in Algorithm 1.

Algorithm 1 Hard negative mining algorithm for tree detection

Require: Manually annotated tree dataset

Ensure: Improved tree detection model

```
Train a baseline detector using the annotated tree dataset
Perform qualitative and quantitative evaluation
while Performance is unsatisfactory do
    Identify hard negative samples
    Update the training dataset by adding the identified hard negatives
    Fine-tune model using the focal loss and the updated training dataset
    Evaluate
end while
```

3.4 Tree health assessment

The tree health assessment stage is formulated as a patch image classification task. In our work, we propose two methods. The first method consists of a machine learning-based model using multi-band vegetation index-derived features. In the second method, we adopted a new semi-supervised autoencoder-based approach using RGB UAV images.

3.4.1 Machine learning-based approach using vegetation indices (VIs)

Figure 3.6 summarizes the proposed method for tree health mapping. It follows a machine learning pipeline. First, we compute vegetation indices from the five-band raw reflectance tree images. Among the high number of vegetation indices that are typically used as indicators of vegetation status [92], there is a subset of VIs that are generally applied for health assessment [21, 93, 94]. 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 [95, 96]. 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 employed to evaluate the status of vegetation [97].

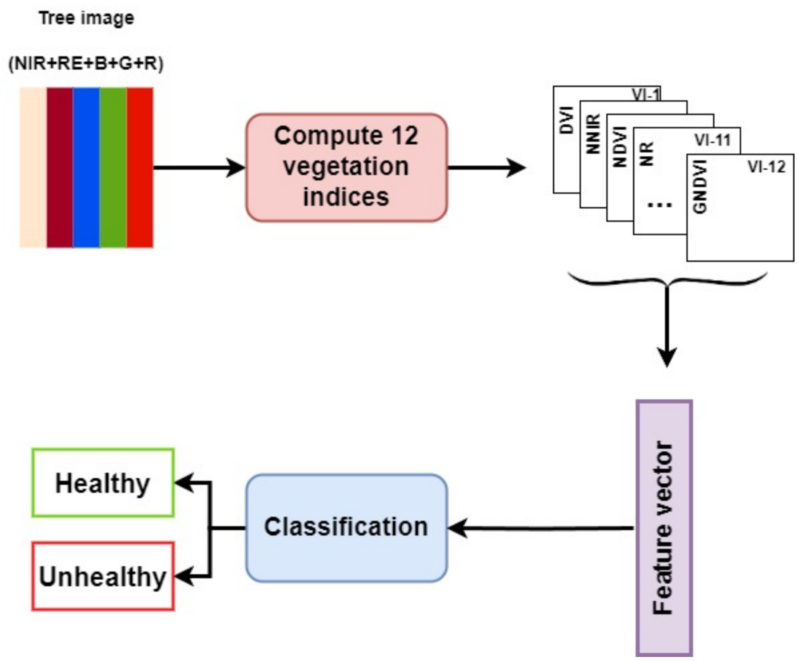

Figure 3.6: Tree health assessment using ML classifier based on a selected set of vegetation indices. The feature vector is composed of the mean of each vegetation index map generated from the tree image patch.

Using these band reflectances, a total of 12 vegetation indices (Table 3.1) 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 [98, 99]. 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 [97, 100, 101]. 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 [102] 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.

Table 3.1: Vegetation indices derived from UAV band reflectance.

Vegetation Index	**Equation**	**References**
Difference Vegetation Index	DVI = Near-infrared (NIR)-Red	[103]
Generalized Difference Vegetation Index	GDVI = NIR – Green	[104]
Green Normalized Difference Vegetation Index	GNDVI = (NIR – Green) / (NIR + Green)	[105]
Green-Red Vegetation Index	GRVI = NIR / Green	[104]
Normalized Difference Aquatic Vegetation Index	NDAVI = (NIR – Blue) / (NIR + Blue)	[106]
Normalized Difference Vegetation Index	NDVI = (NIR – Red) / (NIR + Red)	[103]
Normalized Difference Red-Edge	NDRE = (NIR – RedEdge) / (NIR + RedEdge)	[107]
Normalized Green	NG = Green / (NIR + Red + Green)	[102]
Normalized Red	NR = Red / (NIR + Red + Green)	[102]
Normalized NIR	NNIR = NIR / (NIR + Red + Green)	[102]
Red simple ratio Vegetation Index	RVI = NIR / Red	[108]
Water Adjusted Vegetation Index	WAVI = (1.5*(NIR–Blue))/((NIR+Blue)+0.5)	[106]

Subsequently, 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) [109], decision tree [110], random forests (RF) [111], k-nearest neighbors (KNN) [112], and light gradient boosting machine (LightGBM) [113] that are explained further in Appendix A.2.

3.4.2 Deep learning-based approach using discriminant semi-supervised autoencoder (DSSA)

The second method for tree health classification uses a semi-supervised autoencoder. The proposed network architecture, shown in Figure 3.7, is similar to that of a traditional autoencoder, except for the training process and the loss function.

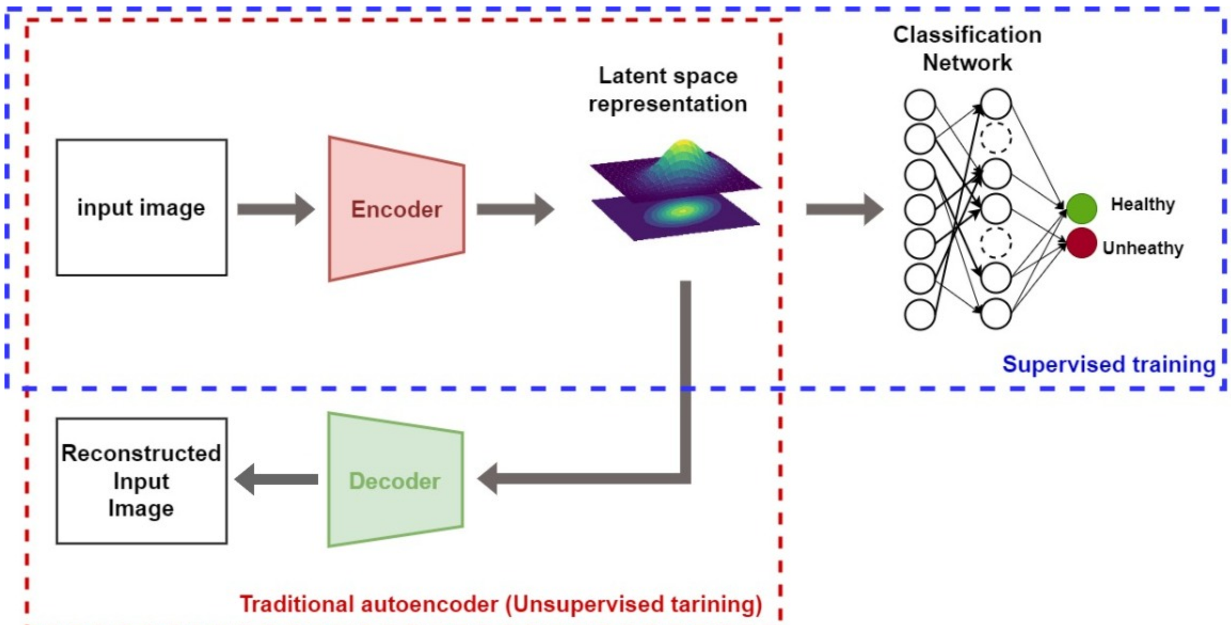

Figure 3.7: Semi-supervised autoencoder for tree health assessment. The red box illustrates the unsupervised learning using a traditional autoencoder, and the blue box represents the supervised training for tree health classification.

The training process is made up of two types of training:

- **Unsupervised training**: During this step, the architecture and the loss function of our semi-supervised autoencoder are the same as that of the traditional autoencoder. Its architecture consists of an encoder that maps the input data into a lower-dimensional latent space representation and a decoder that maps the encoding back into the original input space.

- **Supervised training**: In the supervised training, the architecture of the encoder is the same as the traditional autoencoder with an altered head designed for binary classification. The optimal values of parameters obtained through the unsupervised training process are set as the initial values of the encoder during the supervised training process. Thanks to the supervised training, a semi-supervised autoencoder makes full use of label information to provide a more appropriate representation than traditional autoencoders.

The loss function (Eq. 1) is a combination of reconstruction loss and binary cross-entropy loss.

$$L_{total} = L_{recons}(x, \hat{x}) + L_{binary}(y, \hat{y}), \tag{1}$$

where $L_{recons}(x, \hat{x}) = ||x - \hat{x}||_2^2$ is the reconstruction loss, x is the input tree image, and $\hat{x}$ is the reconstructed image from the autoencoder. $L_{binary}(y, \hat{y}) = -y\log(\hat{y}) - (1-y)\log(1-\hat{y})$ is the binary cross entropy, where y is the true binary label, and $\hat{y}$ is the predicted probability of the positive class.

3.5 Evaluation metrics

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

- Precision P_{d} (Eq. 2) is the percentage of correct detections among all the detected trees.

$$P_d = \frac{\mathrm{TP}_d}{\mathrm{TP}_d + \mathrm{FP}_d} \tag{2}$$

- Recall R_{d} (Eq. 3) is the percentage of correctly detected trees over the total number of trees in the ground truth.

$$R_d = \frac{\mathrm{TP}_d}{\mathrm{TP}_d + \mathrm{FN}_d} \tag{3}$$

- $F1$-score$_d$ (Eq. 4) is the harmonic average of precision and recall.

$$F1\text{-score}_d = 2 * \frac{\mathrm{P}_d * \mathrm{R}_d}{\mathrm{P}_d + \mathrm{R}_d} \tag{4}$$

In equations 2, 3, and 4, 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, as shown in Eq. 5.

$$IOU = \frac{Area(B_1 \cap B_2)}{Area(B_1 \cup B_2)} \tag{5}$$

where $B1$ = The ground truth bounding box

$B2$ = The predicted bounding box

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

- Precision P_c (Eq. 6) is defined as the ratio of correct classifications for a given class to the total number of classifications made for that class.

$$P_c = \frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FP}_c} \tag{6}$$

- Recall R_c (Eq. 7) is defined as the ratio of correct classifications for a given class to the total number of instances that actually belong to that class.

$$R_c = \frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FN}_c} \tag{7}$$

- $F1$-score$_c$ is the harmonic average of P_c and R_c of a given class.

- Accuracy (Eq. 8) is defined as the ratio of the correct classifications to the total number of tree instances classified.

$$Accuracy = \frac{\text{Number of correct classifications}}{\text{Total number of trees classified}} \tag{8}$$

In equations 6, 7, and 8, 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.

Chapter 4

Results

4.1 Tree detection

Table 4.1 shows the detailed and overall cross-validation results of the proposed tree detection 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 outperforms YOLO, achieving a higher F1-score average of 86.24%.

Table 4.1: Detailed and overall cross-validation results of the tree detection stage in terms of precision, recall, and F1-score. Values in bold font correspond to the best-achieved results.

	Baseline 1: DeepForest			**Baseline 2: YOLO**		
Fold	$P_{\mathrm{d}}(\%)$	$R_{\mathrm{d}}(\%)$	$F1\text{-score}_d(\%)$	$P_{\mathrm{d}}(\%)$	$R_{\mathrm{d}}(\%)$	$F1\text{-score}_d(\%)$
Fold1	82.25	87.24	84.67	**83.89**	**87.27**	**85.15**
Fold2	**87.57**	88.06	**87.82**	79.12	**92.35**	84.91
Fold3	**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

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. By contrast, YOLO may struggle to differentiate between trees and other objects that share similar visual features.

In comparison to existing works in the literature, our proposed tree detection method achieves interesting results that are comparable or better to other studies. Our F1-score of 86.24% is comparable to the F1-score of [114] (86%) who uses a local maximum algorithm to detect pine tree on UAV-derived canopy height model imagery. Similarly, [38] obtained an F1-score of 75.75% using an SVM based on shape features to detect palm tree 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. Further, in the ablation study section, we will conduct experiments to demonstrate the importance of our HNM strategy in improving detection results compared to standard detection methods.

4.2 Tree health classification

4.2.1 Tree health classification results using VIs

Table 4.2 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 are performing well in the two classes. Specifically, they achieved higher results for the class healthy. Based on overall accuracy, RandomForest 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 4.2: Overall cross-validation results of different classical machine learning 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(%)
		Random Forest Classifier		
Healthy	98.80	98.80	98.80	**97.52**
Unhealthy	95.50	95.50	95.50	
		Light Gradient Boosting Machine (LightGBM)		
Healthy	99.00	98.8	98.90	97.47
Unhealthy	95.60	96.10	95.80	
		K-nearest neighbors (KNN)		
Healthy	98.60	98.10	98.40	97.07
Unhealthy	92.90	95.00	93.90	
		Support Vector Machine (SVM)		
Healthy	99.40	95.80	97.60	96.31
Unhealthy	86.20	97.80	91.60	
		Decision Tree Classifier		
Healthy	98.40	97.90	98.10	95.91
Unhealthy	92.30	93.90	93.10	

These results show the effectiveness of random forest in accurately classifying tree health status based on the selected vegetation indices. In Table 4.3, we present the feature importance generated by the Random Forest classifier. It allows us to assess the contribution of each feature to the overall accuracy of the model using Gini importance metric [115].

Table 4.3: Feature importance using Random Forest.

Ranking	Feature name	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 infra-red (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.

4.2.2 Tree health classification results using DSSA

We addressed the problem of tree health classification using a discriminative semi-supervised autoencoder and compared its performance with other widely used deep learning-based classifiers, including ResNet [63], VGG [62], and DenseNet [64]. The explored classifier architectures are explained further in Appendix A.3. We trained these models on the same training set, using standard supervised learning. They were pretrained on the ImageNet dataset and fine-tuned on our tree dataset. We evaluated the performance of all classifiers on the test set, using the defined classification metrics.

Table 4.4: Overall cross-validation results of our model (DSSA) compared to other deep learning-based approaches for health assessment in terms of precision, recall, F1-score, and accuracy. Values in bold font correspond to the best-achieved accuracy.

Health Status	P_c(%)	R_c(%)	$F1$-score$_c$(%)	Accuracy(%)
		ResNet-101		
Healthy	96.52	93.68	95.07	90.85
Unhealthy	30.65	43.75	35.46	
		DenseNet-121		
Healthy	96.78	92.56	94.61	90.07
Unhealthy	28.46	48.75	35.74	
		VGG-16		
Healthy	96.02	93.05	94.41	89.75
Unhealthy	25.26	35.00	25.58	
		Discriminant semi-supervised autoencoder (DSSA)		
Healthy	98.05	99.00	98.25	**98.06**
Unhealthy	95.78	96.45	96.03	

Our results in Table 4.4 showed that the proposed semi-supervised autoencoder approach outperformed the other classifiers in all metrics, achieving an accuracy of 98.06%. The ResNet, VGG, and DenseNet classifiers achieved lower accuracy, precision, recall, and F1-score, with the best-performing model being ResNet with an accuracy of 90.85%. From the table, we can see also that the compared classifiers achieved unsatisfactory results for the unhealthy class compared to the healthy class.

The outperformance of the proposed semi-supervised autoencoder over VGG, ResNet, and DenseNet for tree health classification can be attributed to several factors. One of the key advantages of the semi-supervised approach is its ability to leverage both labeled and unlabeled data to learn robust and discriminative features. This is achieved by enforcing a reconstruction loss preserving the model input structure, while also learning informative features for the classification task. However, the compared models rely solely on labeled data and use more complex and computationally intensive architectures that may struggle to generalize to rare classes, leading to a low performance for the class unhealthy.

Overall, our results demonstrate the effectiveness of the proposed discriminant semi-supervised autoencoder approach for tree health classification. Further, in the ablation study section, we will conduct an experiment to demonstrate the importance of the autoencoder module in our DSSA classification method.

4.2.3 Comparaison between the two proposed classification methods

Table 4.5 illustrates a comparison between the two explored methods to perform tree health classification.

Table 4.5: Comparison between VI-based classification method and DSSA. Values in bold font correspond to the best-achieved accuracy.

Health Status	P_c(%)	R_c(%)	$F1$-score$_c$(%)	Accuracy(%)
		Random forest using VIs		
Healthy	98.8	98.8	98.8	97.52
Unhealthy	95.5	95.5	95.5	
		DL approach using DSSA		
Healthy	98.05	99.00	98.25	**98.06**
Unhealthy	95.78	96.45	96.03	

The results of the two methods used for tree health classification are quite promising, demonstrating their effectiveness in accurately assessing the health of trees. The first method employed a Random Forest algorithm, leveraging the utilization of vegetation indices. This approach achieved an impressive overall accuracy of 97.52%. The second method employed a convolutional discriminant semi-supervised autoencoder, which outperformed the Random Forest method with an even higher overall accuracy of 98.07%. These results indicate that both methods are highly reliable and capable of distinguishing between different classes of tree health with remarkable precision.

The high accuracy achieved by both methods highlights their ability to effectively capture and utilize relevant features for tree health classification. The Random Forest method's reliance on vegetation indices suggests that these indices provide valuable information that significantly contributes to accurate classification. On the other hand, the superior performance of the convolutional semi-supervised autoencoder indicates the power of deep learning techniques in extracting intricate patterns and representations from the input data.

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 [56] (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 [22] who mapped tree health status using multispectral imagery (95%) or hyperspectral imagery (91%). Our method also outperforms statistical methods reviewed in [116] 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.

It is worth noting that both methods exhibited excellent performance for both classes, suggesting their robustness in correctly identifying both healthy and unhealthy trees, providing similar or better levels of accuracy to comparable studies.

4.3 Ablation study

4.3.1 Importance of the HNM technique for tree detection

In order to evaluate the performance of the hard negative mining strategy for tree detection, we reported the results of training a baseline object detector without hard negative samples by comparison to our HNM-based approach. Table 4.6 reports the overall cross-validation results using the two baseline models: DeepForest and YOLO-v5. The reported results show that both YOLO and DeepForest benefited from hard negative mining, with significant improvements in F1-score. However, DeepForest consistently outperformed YOLO in all experiments, achieving higher results. YOLO fine-tuned with the mined hard negatives achieved an F1-score of 84.81%, outperforming the YOLO baseline by 2.17%, which means 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.

Table 4.6: Ablation study results 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	$P_{\text{d}}(\%)$	$R_{\text{d}}(\%)$	$F1\text{-score}_d(\%)$
	Without HNM		
Baseline 1: DeepForest	84.82	86.18	85.46
Baseline 2: YOLO	79.40	88.05	82.64
	With HNM		
Baseline 1: DeepForest	**85.85**	86.67	**86.24**
Baseline 2: YOLO	82.01	**88.99**	84.81

Overall, our results demonstrate the importance of HNM in tree detection to improve the detection ability in both baseline models.

4.3.2 Importance of using the autoencoder for tree health assessment

We conducted an ablation study to investigate the contribution of unsupervised training using the autoencoder component of the proposed approach. In Table 4.7, we report the results of our classifier without and with the use of the autoencoder. We can see that the unsupervised pretraining of the autoencoder is a key component of the proposed approach, contributing to its superior performance. Training the autoencoder using unlabeled data, followed by supervised training on labeled data for the specific task of tree classification resulted in a 2.71% improvement in overall accuracy. The improvement can also be seen through the other metrics, precision, recall, and F1-score.

Table 4.7: Ablation Study of the classification step. Values in bold font correspond to the best results

Health Status	P_{c}(%)	R_{c}(%)	$F1$-score$_c$(%)	Accuracy(%)
Our classification model without the autoencoder				
Healthy	97.57	96.56	97.06	95.35
Unhealthy	87.49	90.69	88.97	
Our classification model with the autoencoder				
Healthy	98.05	99.00	98.25	**98.06**
Unhealthy	95.78	96.45	96.03	

Overall, these results underscore the effectiveness of our approach to extracting relevant features.

Chapter 5

Conclusions

In this thesis, an effective framework is proposed to address the problem of 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 explored the use of Random forest based on twelve selected vegetation indices and proposed a new discriminant semi-supervised autoencoder. The conducted experiments showcase the resilience of the suggested framework designed for evaluating the health of orchard trees using UAV images. Specifically, our framework attains an F1-score of 86.24% for accurately detecting trees and achieves an impressive overall accuracy of 98.06% for assessing tree health. Furthermore, the versatility of our approach enables its applicability in diverse UAV applications that involve the process of detection and classification.

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 [117] for tree identification, species classification [118], tree health monitoring [119], and biodiversity assessment [119]. Further research might also focus on implementing periodical flights at different attack stages to provide a multitemporal analysis of tree health assessment tasks.

Appendix A

A.1 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 [90] is a deep-learning model developed to detect individual trees on high-resolution RGB imagery. It is mainly based on the Retinanet model [88], a one-stage object detector that enables the focal loss function to address the excessive foreground-background class imbalance. RetinaNet is a network architecture based on ResNet as a backbone [63] that generates a rich, multiscale convolutional Feature Pyramid Network (FPN) [120] that is connected to two subnetworks: one for classifying anchor boxes and another one for regressing object boxes. 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 prebuild 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) [89] 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 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) [121] and Spatial Pooling Pyramid (SPP) [122] are adopted for parameter aggregation from different backbone levels for different detector levels, instead of FPN used in YOLO-v3 [49]. 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).

A.2 Brief description of the explored machine learning classifiers for tree health classification using VIs, that is, K-Nearest Neighbors (KNN), Light Gradient Boosting Machine (LightGBM), Support Vector machine (SVM), and decision tree (DT).

KNN [112] 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.

LightGBM [113] 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. 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 [109] 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.

DT [110] 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.

A.3 Brief description of the explored deep learning classifiers compared to the proposed DSSA, that is, visual geometry group (VGG), residual network (ResNet), and dense network (DenseNet).

VGG is a popular convolutional neural network (CNN) classification architecture. Its key idea is to use of a stack of convolutional layers with small filter sizes (3x3) and a large number of channels, followed by max-pooling layers for downsampling. This design choice allows the network to learn more complex features while keeping the number of parameters manageable. The VGG architecture is known for its simplicity. The most common versions of VGG are VGG-16 and VGG-19, which differ in the number of weight layers. VGG16 has 16 weight layers, including 13 convolutional layers and 3 fully connected layers, while VGG19 has 19 weight layers, including 16 convolutional layers and 3 fully connected layers.

ResNet is a deep convolutional neural network architecture. It addresses the problem of vanishing gradients in very deep networks by introducing skip connections, or ”residual connections,” that allow the network to learn residual functions instead of directly learning the desired underlying mappings. The core idea behind ResNet is the concept of residual learning. In traditional neural networks, each layer learns a set of features that transform the input data. However, as the network becomes deeper, it becomes increasingly difficult to optimize the network’s performance due to the vanishing gradient problem, where gradients diminish as they propagate through many layers. ResNet tackles this problem by introducing residual blocks, which enable the network to directly learn the difference between the input and the desired output.

DenseNet is a convolutional neural network architecture. It addresses the vanishing gradient problem and improves information flow by introducing dense connections between layers, allowing each layer to receive direct inputs from all preceding layers. The main idea behind DenseNet is to encourage feature reuse throughout the network by connecting each layer to every other layer in a feed-forward fashion. This dense connectivity pattern enables direct information flow from the early layers to the later layers, facilitating gradient propagation and enhancing feature propagation.

Bibliography

[1] Zhijie Liu, Pengju Guo, Heng Liu, Pan Fan, Pengzong Zeng, Xiangyang Liu, Ce Feng, Wang Wang, and Fuzeng Yang. Gradient boosting estimation of the leaf area index of apple orchards in uav remote sensing. *Remote Sensing*, 13(16):3263, 2021.

[2] F Talebzadeh and C Valeo. Evaluating the effects of environmental stress on leaf chlorophyll content as an index for tree health. In *IOP Conference Series: Earth and Environmental Science*, volume 1006, page 012007. IOP Publishing, 2022.

[3] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[4] Wassim Bouachir, Koffi Eddy Ihou, Houssem-Eddine Gueziri, Nizar Bouguila, and Nicolas Bélanger. Computer vision system for automatic counting of planting microsites using uav imagery. *IEEE Access*, 7:82491–82500, 2019.

[5] A Haddadi, B Leblon, and G Patterson. Detecting and counting orchard trees on unmanned aerial vehicle (uav)-based images using entropy and ndvi features. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 43:1211–1215, 2020.

[6] Ye Zhang, Gang Wang, Mingchao Li, and Shuai Han. Automated classification analysis of geological structures based on images data and deep learning model. *Applied Sciences*, 8(12):2493, 2018.

[7] L Geng, YF Zhang, PF Wang, J Jay Wang, Jerry YH Fuh, and SH Teo. Uav surveillance mission planning with gimbaled sensors. In *11th IEEE International Conference on Control & Automation (ICCA)*, pages 320–325. IEEE, **2014**.

[8] Konstantinos Kanistras, Goncalo Martins, Matthew J Rutherford, and Kimon P Valavanis. A survey of unmanned aerial vehicles (uavs) for traffic monitoring. In *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 221–234. IEEE, **2013**.

[9] Yang Long, Yiping Gong, Zhifeng Xiao, and Qing Liu. Accurate object localization in remote sensing images based on convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 55(5):2486–2498, 2017.

[10] Guoli Wang, Xinchao Wang, Bin Fan, and Chunhong Pan. Feature extraction by rotation-invariant matrix representation for object detection in aerial image. *IEEE Geoscience and Remote Sensing Letters*, 14(6):851–855, 2017.

[11] Thomas Moranduzzo and Farid Melgani. Detecting cars in uav images with a catalog-based approach. *IEEE Transactions on Geoscience and Remote Sensing*, 52(10):6356–6367, 2014.

[12] Gong Cheng, Junwei Han, Peicheng Zhou, and Dong Xu. Learning rotation-invariant and fisher discriminative convolutional neural networks for object detection. *IEEE Transactions on Image Processing*, 28(1):265–278, 2018.

[13] Yudong Lin, Hongjie He, Zhongke Yin, and Fan Chen. Rotation-invariant object detection in remote sensing images based on radial-gradient angle. *IEEE Geoscience and Remote Sensing Letters*, 12(4):746–750, 2014.

[14] Hela Jemaa, Wassim Bouachir, Brigitte Leblon, and Nizar Bouguila. Computer vision system for detecting orchard trees from uav images. *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 43:661–668, 2022.

[15] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, **2009**.

[16] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755, **2014**.

[17] Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge J. Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Dota: A large-scale dataset for object detection in aerial images. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3974–3983, 2017.

[18] Shaohua Wan, Zhijun Chen, Zhang Tao, Bo Zhang, and Kong kat Wong. Bootstrapping face detection with hard negative examples. *ArXiv*, 2016.

[19] Jan Rudolf Karl Lehmann, Felix Nieberding, Torsten Prinz, and Christian Knoth. Analysis of unmanned aerial system-based cir images in forestry—a new perspective to monitor pest infestation levels. *Forests*, 6(3):594–612, 2015.

[20] Roope Näsi, Eija Honkavaara, Päivi Lyytikäinen-Saarenmaa, Minna Blomqvist, Paula Litkey, Teemu Hakala, Niko Viljanen, Tuula Kantola, Topi Tanhuanpää, and Markus Holopainen. Using uav-based photogrammetry and hyperspectral imaging for mapping bark beetle damage at tree-level. *Remote Sensing*, 7(11):15467–15493, 2015.

[21] Jonathan P Dash, Michael S Watt, Grant D Pearse, Marie Heaphy, and Heidi S Dungey. Assessing very high resolution uav imagery for monitoring forest health during a simulated disease outbreak. *ISPRS Journal of Photogrammetry and Remote Sensing*, 131:1–14, 2017.

[22] Marian-Daniel Iordache, Vasco Mantas, Elsa Baltazar, Klaas Pauly, and Nicolas Lewyckyj. A machine learning approach to detecting pine wilt disease using airborne spectral imagery. *Remote Sensing*, 12(14):2280, 2020.

[23] Kaori Otsu, Magda Pla, Andrea Duane, Adrián Cardil, and Lluís Brotons. Estimating the threshold of detection on tree crown defoliation using vegetation indices from uas multispectral imagery. *Drones*, 3(4):80, 2019.

[24] Qinxue Meng, Daniel Catchpoole, David Skillicom, and Paul J Kennedy. Relational autoencoder for feature extraction. In *2017 International joint conference on neural networks (IJCNN)*, pages 364–371. IEEE, 2017.

[25] Kunjin Chen, Jun Hu, and Jinliang He. A framework for automatically extracting overvoltage features based on sparse autoencoder. *IEEE Transactions on Smart Grid*, 9(2):594–604, 2016.

[26] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

[27] Wayne A Sinclair, Howard H Lyon, et al. *Diseases of trees and shrubs.* Comstock Publishing Associates, 2005.

[28] Sindhuja Sankaran, Ashish Mishra, Reza Ehsani, and Cristina Davis. A review of advanced techniques for detecting plant diseases. *Computers and Electronics in Agriculture*, 72(1):1–13, 2010.

[29] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[30] Amin Sedaghat, Mehdi Mokhtarzade, and Hamid Ebadi. Uniform robust scale-invariant feature matching for optical remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 49(11):4516–4527, 2011.

[31] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[32] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. *Proceedings of CVPR, 2005*, pages 886–893, 2005.

[33] Wen Shao, Wen Yang, Gang Liu, and Jie Liu. Car detection from high-resolution aerial imagery using multiple features. *2012 IEEE International Geoscience and Remote Sensing Symposium*, pages 4379–4382, 2012.

[34] Philippe Maillard and Marília F Gomes. Detection and counting of orchard trees from vhr images using a geometrical-optical model and marked template matching. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3:75, 2016.

[35] Salim Malek, Yakoub Bazi, Naif Alajlan, Haikel AlHichri, and Farid Melgani. Efficient framework for palm tree detection in uav images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(12):4692–4703, 2014.

[36] Yakoub Bazi, Salim Malek, Naif A. Alajlan, and Haikel Salem Alhichri. An automatic approach for palm tree counting in uav images. *2014 IEEE Geoscience and Remote Sensing Symposium*, pages 537–540, 2014.

[37] Yiran Wang, Xiaolin Zhu, and Bo Wu. Automatic detection of individual oil palm trees from uav images using hog features and an svm classifier. *International Journal of Remote Sensing*, 40(19):7356–7370, 2019.

[38] Abhishek Manandhar, Ludwig Hoegner, and Uwe Stilla. Palm tree detection using circular autocorrelation of polar shape matrix. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3:465, 2016.

[39] Anastasiia Safonova, Siham Tabik, Domingo Alcaraz-Segura, Alexey Rubtsov, Yuriy Maglinets, and Francisco Herrera. Detection of fir trees (abies sibirica) damaged by the bark beetle in unmanned aerial vehicle images with deep learning. *Remote Sensing*, 11(6):643, 2019.

[40] Ha Trang Nguyen, Maximo Larry Lopez Caceres, Koma Moritake, Sarah Kentsch, Hase Shu, and Yago Diez. Individual sick fir tree (abies mariesii) identification in insect infested forests by means of uav images and deep learning. *Remote Sensing*, 13(2):260, 2021.

[41] Anastasiia Safonova, Yousif Hamad, Egor Dmitriev, Georgi Georgiev, Vladislav Trenkin, Margarita Georgieva, Stelian Dimitrov, and Martin Iliev. Individual tree crown delineation for the species classification and assessment of vital status of forest stands from uav images. *Drones*, 5(3):77, 2021.

[42] Saeed Al Mansoori, Alavi Kunhu, and Hussain Al Ahmad. Automatic palm trees detection from multispectral uav data using normalized difference vegetation index and circular hough transform. In *Remote Sensing*, volume 10792, pages 11–19, **2018**.

[43] Omair Hassaan, Ahmad Kamal Nasir, Hubert Roth, and M Fakhir Khan. Precision forestry: trees counting in urban areas using visible imagery based on an unmanned aerial vehicle. *IFAC-PapersOnLine*, 49(16):16–21, 2016.

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

[45] Weijia Li, Haohuan Fu, and Le Yu. Deep convolutional neural network based large-scale oil palm tree detection for high-resolution remote sensing images. *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 846–849, 2017.

[46] Thani Jintasuttisak, Eran Edirisinghe, and Ali Elbattay. Deep neural network based date palm tree detection in drone imagery. *Computers and Electronics in Agriculture*, 192:106560, 2022.

[47] Anderson Aparecido dos Santos, Jose Marcato Junior, Márcio Santos Araújo, David Robledo Di Martini, Everton Castelão Tetila, Henrique Lopes Siqueira, Camila Aoki, Anette Eltner, Edson Takashi Matsubara, Hemerson Pistori, et al. Assessment of cnn-based methods for individual tree detection on images captured by rgb cameras attached to uavs. *Sensors*, 19(16):3595, 2019.

[48] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28, 2015.

[49] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *ArXiv*, 2018.

[50] Michael Fromm, Matthias Schubert, Guillermo Castilla, Julia Linke, and Greg McDermid. Automated detection of conifer seedlings in drone imagery using convolutional neural networks. *Remote Sensing*, 11(21):2585, 2019.

[51] W. Liu, Dragomir Anguelov, D. Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*, pages 21–37, **2015**.

[52] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. *Advances in Neural Information Processing Systems*, 29, 2016.

[53] Matheus Pinheiro Ferreira, Danilo Roberti Alves de Almeida, Daniel de Almeida Papa, Juliano Baldez Silva Minervino, Hudson Franklin Pessoa Veras, Arthur Formighieri, Caio Alexandre Nascimento Santos, Marcio Aurélio Dantas Ferreira, Evandro Orfano Figueiredo, and Evandro José Linhares Ferreira. Individual tree detection and species classification of amazonian palms using uav images and deep learning. *Forest Ecology and Management*, 475:118397, 2020.

[54] Kunyong Yu, Zhenbang Hao, Christopher J Post, Elena A Mikhailova, Lili Lin, Gejin Zhao, Shangfeng Tian, and Jian Liu. Comparison of classical methods and mask r-cnn for automatic tree detection and mapping using uav imagery. *Remote Sensing*, 14(2):295, 2022.

[55] Zhenbang Hao, Lili Lin, Christopher J Post, Elena A Mikhailova, Minghui Li, Yan Chen, Kunyong Yu, and Jian Liu. Automated tree-crown and height detection in a young forest plantation using mask region-based convolutional neural network (mask r-cnn). *ISPRS Journal of Photogrammetry and Remote Sensing*, 178:112–123, 2021.

[56] Ning Zhang, Yueting Wang, and Xiaoli Zhang. Extraction of tree crowns damaged by dendrolimus tabulaeformis tsai et liu via spectral-spatial classification using uav-based hyperspectral images. *Plant Methods*, 16:1–19, 2020.

[57] André Duarte, Luis Acevedo-Muñoz, Catarina I Gonçalves, Luís Mota, Alexandre Sarmento, Margarida Silva, Sérgio Fabres, Nuno Borralho, and Carlos Valente. Detection of longhorned

borer attack and assessment in eucalyptus plantations using uav imagery. *Remote Sensing*, 12(19):3153, 2020.

[58] Ioannis Navrozidis, Anne Haugommard, Dimitrios Kasampalis, Thomas Alexandridis, Fabien Castel, Dimitrios Moshou, Georgios Ovakoglou, Xanthoula Eirini Pantazi, Afroditi Alexandra Tamouridou, Anastasia L Lagopodi, et al. Assessing olive trees health using vegetation indices and mundi web services for sentinel-2 images. In *Hellenic Association on Information and Communication Technologies in Agriculture, Food Environment*, pages 130–136, **2020**.

[59] Kanitta Yarak, Apichon Witayangkurn, Kunnaree Kritiyutanont, Chomchanok Arunplod, and Ryosuke Shibasaki. Oil palm tree detection and health classification on high-resolution imagery using deep learning. *Agriculture*, 11(2):183, 2021.

[60] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[61] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv*, 2016.

[62] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[63] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2015.

[64] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2016.

[65] Jean-Daniel Sylvain, Guillaume Drolet, and Nicolas Brown. Mapping dead forest cover using a deep convolutional neural network and digital aerial photography. *ISPRS Journal of Photogrammetry and Remote Sensing*, 156:14–26, 2019.

[66] Ziliang Huang, Rujing Wang, Ying Cao, Shijian Zheng, Yue Teng, Fenmei Wang, Liusan Wang, and Jianming Du. Deep learning based soybean seed classification. *Computers and Electronics in Agriculture*, 202:107393, 2022.

[67] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1251–1258, 2017.

[68] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[69] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1492–1500, 2017.

[70] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv*, 2017.

[71] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6848–6856, 2018.

[72] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

[73] Kah-Kay Sung. Learning and example selection for object and pattern detection. 1996.

[74] Abhinav Shrivastava, Abhinav Kumar Gupta, and Ross B. Girshick. Training region-based object detectors with online hard example mining. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 761–769, 2016.

[75] SouYoung Jin, Aruni RoyChowdhury, Huaizu Jiang, Ashish Singh, Aditya Prasad, Deep Chakraborty, and Erik G. Learned-Miller. Unsupervised hard example mining from videos for improved object detection. In *European Conference on Computer Vision*, pages 307–324, **2018**.

[76] Yu Liu. An improved faster r-cnn for object detection. *2018 11th International Symposium on Computational Intelligence and Design (ISCID)*, 02:119–123, 2018.

[77] Xudong Sun, Pengcheng Wu, and Steven CH Hoi. Face detection using deep learning: An improved faster rcnn approach. *Neurocomputing*, 299:42–50, 2018.

[78] Liliang Zhang, Liang Lin, Xiaodan Liang, and Kaiming He. Is faster r-cnn doing well for pedestrian detection? *ArXiv*, pages 443–457, 2016.

[79] Lei Zhang, Yuehuan Wang, and Yang Huo. Object detection in high-resolution remote sensing images based on a hard-example-mining network. *IEEE Transactions on Geoscience and Remote Sensing*, 59(10):8768–8780, 2020.

[80] Xiaolong Wang, Abhinav Shrivastava, and Abhinav Gupta. A-fast-rcnn: Hard positive generation via adversary for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[81] Niranjan Ravi and Mohamed El-Sharkawy. Improved single shot detector with enhanced hard negative mining approach. In *2022 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, pages 25–30. IEEE, 2022.

[82] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th international conference on international conference on machine learning*, pages 833–840, 2011.

[83] Jasper Snoek, Ryan Prescott Adams, and Hugo Larochelle. Nonparametric guidance of autoencoder representations using label information. *Journal of Machine Learning Research*, 2012.

[84] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. *Advances in neural information processing systems*, 27, 2014.

[85] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

[86] Xiaoyi Luo, Xianmin Li, Ziyang Wang, and Jun Liang. Discriminant autoencoder for feature extraction in fault diagnosis. *Chemometrics and Intelligent Laboratory Systems*, 192:103814, 2019.

[87] Abhishek Dutta and Andrew Zisserman. The via annotation software for images, audio and video. *Proceedings of the 27th ACM International Conference on Multimedia*, pages 2276–2279, 2019.

[88] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42:318–327, 2017.

[89] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2015.

[90] Ben G Weinstein, Sergio Marconi, Mélaine Aubry-Kientz, Gregoire Vincent, Henry Senyondo, and Ethan P White. Deepforest: A python package for rgb deep learning tree crown delineation. *Methods in Ecology and Evolution*, 11(12):1743–1751, 2020.

[91] Ben G Weinstein, Sergio Marconi, Stephanie A Bohlman, Alina Zare, Aditya Singh, Sarah J Graves, and Ethan P White. A remote sensing derived data set of 100 million individual tree crowns for the national ecological observatory network. *Elife*, 10:e62922, 2021.

[92] Nobuyuki Kobayashi, Hiroshi Tani, Xiufeng Wang, and Rei Sonobe. Crop classification using spectral indices derived from sentinel-2a imagery. *Journal of Information and Telecommunication*, 4(1):67–90, 2020.

[93] Alessia Cogato, Vinay Pagay, Francesco Marinello, Franco Meggio, Peter Grace, and Massimiliano De Antoni Migliorati. Assessing the feasibility of using sentinel-2 imagery to quantify the impact of heatwaves on irrigated vineyards. *Remote Sensing*, 11(23):2869, 2019.

[94] Paweł Hawryło, Bartłomiej Bednarz, Piotr Wężyk, and Marta Szostak. Estimating defoliation of scots pine stands using machine learning methods and vegetation indices of sentinel-2. *European Journal of Remote Sensing*, 51(1):194–204, 2018.

[95] Pablo J Zarco-Tejada, John R Miller, GH Mohammed, Thomas L Noland, and PH Sampson. Vegetation stress detection through chlorophyll a+ b estimation and fluorescence effects on hyperspectral imagery. *Journal of Environmental Quality*, 31(5):1433–1441, 2002.

[96] Karen M Barry, C Stone, and CL Mohammed. Crown-scale evaluation of spectral indices for defoliated and discoloured eucalypts. *International Journal of Remote Sensing*, 29(1):47–69, 2008.

[97] Zakariyyaa Oumar and Onisimo Mutanga. Using worldview-2 bands and indices to predict bronze bug (thaumastocoris peregrinus) damage in plantation forests. *International Journal of Remote Sensing*, 34(6):2236–2249, 2013.

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

[99] Jan Verbesselt, Andrew Robinson, Christine Stone, and Darius Culvenor. Forecasting tree mortality using change metrics derived from modis satellite data. *Forest Ecology and Management*, 258(7):1166–1173, 2009.

[100] Bisun Datt. Remote sensing of chlorophyll a, chlorophyll b, chlorophyll a+ b, and total carotenoid content in eucalyptus leaves. *Remote Sensing of Environment*, 66(2):111–121, 1998.

[101] Xinping Deng, Shanxin Guo, Luyi Sun, and Jinsong Chen. Identification of short-rotation eucalyptus plantation at large scale using multi-satellite imageries and cloud computing platform. *Remote Sensing*, 12(13):2153, 2020.

[102] Sreekala G Bajwa and Lei Tian. Multispectral cir image calibration for cloud shadow and soil background influence using intensity normalization. *Applied Engineering in Agriculture*, 18(5):627, 2002.

[103] Compton J Tucker. Red and photographic infrared linear combinations for monitoring vegetation. *Remote Sensing of Environment*, 8(2):127–150, 1979.

[104] Ravi P Sripada, Ronnie W Heiniger, Jeffrey G White, and Alan D Meijer. Aerial color infrared photography for determining early in-season nitrogen requirements in corn. *Agronomy Journal*, 98(4):968–977, 2006.

[105] C Buschmann and E Nagel. In vivo spectroscopy and internal optics of leaves as basis for remote sensing of vegetation. *International Journal of Remote Sensing*, 14(4):711–722, 1993.

[106] Paolo Villa, Alijafar Mousivand, and Mariano Bresciani. Aquatic vegetation indices assessment through radiative transfer modeling and linear mixture simulation. *International Journal of Applied Earth Observation and Geoinformation*, 30:113–127, 2014.

[107] EM Barnes, TR Clarke, SE Richards, PD Colaizzi, J Haberland, M Kostrzewski, P Waller, C Choi, E Riley, T Thompson, et al. Coincident detection of crop water stress, nitrogen status and canopy density using ground based multispectral data. In *Fifth International Conference on Precision Agriculture, Bloomington, MN, USA*, volume 1619, page 6, **2000**.

[108] John Wilson Rouse, Rüdiger H Haas, John A Schell, Donald W Deering, et al. Monitoring vegetation systems in the great plains with erts. *NASA Special Publications*, 351(1):309, 1974.

[109] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.

[110] Wei-Yin Loh. Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1):14–23, 2011.

[111] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[112] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

[113] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, 30, 2017.

[114] Midhun Mohan, Carlos Alberto Silva, Carine Klauberg, Prahlad Jat, Glenn Catts, Adrián Cardil, Andrew Thomas Hudak, and Mahendra Dia. Individual tree detection from unmanned aerial vehicle (uav) derived canopy height model in an open canopy mixed conifer forest. *Forests*, 8(9):340, 2017.

[115] Stefano Nembrini, Inke R König, and Marvin N Wright. The revival of the gini importance? *Bioinformatics*, 34(21):3711–3718, 2018.

[116] Sonia M Ortiz, Johannes Breidenbach, and Gerald Kändler. Early detection of bark beetle green attack using terrasar-x and rapideye data. *Remote Sensing*, 5(4):1912–1931, 2013.

[117] Telmo Adão, Jonáš Hruška, Luís Pádua, José Bessa, Emanuel Peres, Raul Morais, and Joaquim Joao Sousa. Hyperspectral imaging: A review on uav-based sensors, data processing and applications for agriculture and forestry. *Remote Sensing*, 9(11):1110, 2017.

[118] Adrien Michez, Hervé Piégay, Jonathan Lisein, Hugues Claessens, and Philippe Lejeune. Classification of riparian forest species and health condition using multi-temporal and hyperspatial imagery from unmanned aerial system. *Environmental Monitoring and Assessment*, 188:1–19, 2016.

[119] Magdalena Smigaj, Rachel Gaulton, Stuart L. Barr, and Juan C. Suárez. Uav-borne thermal imaging for forest health monitoring: Detection of disease-induced canopy temperature increase. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pages 349–354, 2015.

[120] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, 2016.

[121] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8759–8768, 2018.

[122] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1904–1916, 2015.