



Article

A Deep Learning-Based Generalized System for Detecting Pine Wilt Disease Using RGB-Based UAV Images

Jie You ¹ , Ruirui Zhang ² and Joonwhoan Lee ^{1,*}

¹ Department of Computer Engineering, Jeonbuk National University, Jeonju-si 54896, Korea; youjie80@gmail.com

² School of Computer Science and Engineering, Cangzhou Normal University, Cangzhou 061001, China; czwxz@caztc.edu.cn

* Correspondence: chlee@jbnu.ac.kr

Abstract: Pine wilt is a devastating disease that typically kills affected pine trees within a few months. In this paper, we confront the problem of detecting pine wilt disease. In the image samples that have been used for pine wilt disease detection, there is high ambiguity due to poor image resolution and the presence of “disease-like” objects. We therefore created a new dataset using large-sized orthophotographs collected from 32 cities, 167 regions, and 6121 pine wilt disease hotspots in South Korea. In our system, pine wilt disease was detected in two stages: in the first stage, the disease and hard negative samples were collected using a convolutional neural network. Because the diseased areas varied in size and color, and as the disease manifests differently from the early stage to the late stage, hard negative samples were further categorized into six different classes to simplify the complexity of the dataset. Then, in the second stage, we used an object detection model to localize the disease and “disease-like” hard negative samples. We used several image augmentation methods to boost system performance and avoid overfitting. The test process was divided into two phases: a patch-based test and a real-world test. During the patch-based test, we used the test-time augmentation method to obtain the average prediction of our system across multiple augmented samples of data, and the prediction results showed a mean average precision of 89.44% in five-fold cross validation, thus representing an increase of around 5% over the alternative system. In the real-world test, we collected 10 orthophotographs in various resolutions and areas, and our system successfully detected 711 out of 730 potential disease spots.

Keywords: pine wilt disease dataset; GIS application visualization; test-time augmentation; object detection; hard negative mining



Citation: You, J.; Zhang, R.; Lee, J. A Deep Learning-Based Generalized System for Detecting Pine Wilt Disease Using RGB-Based UAV Images. *Remote Sens.* **2022**, *14*, 150. <https://doi.org/10.3390/rs14010150>

Academic Editor: Karem Chokmani

Received: 19 December 2021

Accepted: 23 December 2021

Published: 30 December 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Pinewood nematode (*Bursaphelenchus xylophilus*) is a microscopic worm-like creature that causes pine wilt disease (PWD), which poses a serious threat to pine forests, as infected trees die within a few months [1]. Pinewood nematodes can quickly pass from sick to healthy trees via biologic vectors or human activities. The disease is responsible for substantial environmental and economic losses in the pine forests of Europe, the Americas, and Asia [2].

Remote sensing technology is very powerful and widely used to monitor criminal activity or geographical changes, forecast weather, scan using airborne lasers, and plan urban developments. Unmanned Aerial Vehicles (UAVs) are capable of capturing high-quality aerial photographs or videos through various high-precision sensors and automated GPS (the global positioning system) navigation. Researchers [3–6] have recently used UAVs for tree species classification. In early 2005, ref. [7] successfully utilized remote-piloted vehicles to collect viable spores of *Gibberella Zeae* (anamorph *Fusarium graminearum*) and evaluate the impact of their transport. Some studies [8–10] have focused on plant disease identification based on spectral and texture features captured by aerial images.

Despite these achievements, it is still difficult to accurately detect PWD in high accuracy using UAV images for the following reasons: (1) PWD data collection is time-consuming and costly. Data must be collected from August through September. As PWD-infected trees typically start to die and appear red in late August, it is best to collect images after August, once these symptoms have appeared. However, after October, broad-leaved trees (such as maple trees) change color and show a similar appearance to PWD-infected trees. (2) It is difficult to obtain high-quality orthophotographs, because they require the careful selection of proper settings in terms of image resolution, shooting perspectives, exposure time, and weather. Typically, captured patch images can be overlapped, and orthophotograph can be created based on these overlapped patch images. However, such orthophotographs often suffer from poor image registration due to the elevation differences in forest areas. (3) PWD symptoms vary between stages. In the early stage, PWD-infected tree has a similar appearance to a healthy tree. In the late stage, infected trees show visual symptoms with features that appear close to those of yellow land, bare branches, or maple trees. Color-based algorithms typically show poor detection performance for this issue. (4) Annotation of these data are a challenge and time-consuming task; mis-annotation often occurs due to the poor image resolution and the background similarity.

A common method of locating PWD is based on the handcrafted features of texture and color; specifically, identifying their corresponding relationship to find infected trees [11–13]. Their results have been based on a limited number of samples (Table 1), and it is more desirable to analyze PWD using a large number of data samples. Deep learning technology has a powerful ability to process complicated GIS (geographic information system) data. While the encoder of a deep convolutional network automatically extracts inherent feature information from a given input data X , the decoder tries to approximate the desired outputs Y as closely as possible to solve complex classification and regression problems. Previous studies have used deep learning methods [14] to detect interesting objects. In this paper, we propose a deep learning-based PWD detection system which is verified to be effective and can be generalized for various object detection models using RGB-based images.

To summarize, our major contributions include:

- We collected and annotated a large dataset for PWD detection using orthophotographs taken from different areas in South Korea. The dataset has 6121 PWD hotspots in total. The obtained PWD-infected trees have arbitrary sizes and resolutions, and they show various symptoms during the different stages of infection.
- In our work, large number of easy negative samples from healthy area cause unbalanced positive and negative samples, which prevents model from effective learning. Besides, some “disease-like” objects (hard negative samples)—such as maple trees are hard to be correctly recognized. We overcome those difficulties using hard negative example mining [15,16]. The hard negative samples are selected in trained network, and merged with genuine PWD infected object to retrain the network. For simplification, we accumulate all the “disease-like” objects and categorize them into six negative categories (“white branch (wb)”, “white green (wg)”, “yellow land”, “maple”, “oak”, and something “yellow”) to be learned along with the positive PWD objects. This simple method improves the discriminatory power of the networks and easily exclude negative objects from the final detection result.
- Drone captured data has varying image resolutions as well as differing levels of illumination and sharpness. To achieve successful detection, our network was learned with augmented training data considering the diverse imaging conditions.
- The test-time augmentation method was used to make robust predictions. To our knowledge, this method has never been applied to PWD detection problems.
- We deployed the proposed system for real-world scenarios and tested it on several orthophotographs. We found that the proposed system successfully detected 711 out of 730 PWD-infected trees. The predicted model is made freely available as an open-source module for further field investigation.

Table 1. Information about the data sets in various approaches. Symbol “-” denote the missing data in reference paper.

Data Source	GSD (cm/pixel)	The Number of Bands	Collected Cities	PWD Infected Trees
WV/2 + WV/3 [11]	45	8	2	311 + 344
Jian [13]	8.97	3	1	340
MS + HS [17]	5	5	5	44 + 40
Anbi [18]	-	3	2	59
Qingkou [19]	4	4	3	42
Huangshan + Wuhan + Yantai [20]	-	5	3	1706
Taishan [21]	-	3	3	3670
Ours+Real environment test	3.2~10.2	3	32	6121 + 730

2. Related Work

2.1. Pine Wilt Disease Classification

The damage to pine forests caused by PWD is a serious social issue, and several attempts have been made to track the disease using orthophotographs. This method of PWD detection is challenging because it is only possible to collect the data within a limited time. An infected pine tree shows its disease symptoms through changes to the colors of its pine needles; the pine needles gradually change in appearance from solid green to yellow and then brown, and the dead tree color finally turns to ash grey.

Reference [22] analyzed the spatial distribution pattern of damaged trees while introducing the Classification and Regression Trees (CART) model. Another study [23] effectively extracted ecological information to predict risk rates of infected trees using a self-organizing map (SOM) and random forest models.

In later studies, spectral sensors were used to capture hyper-spectral images for PWD analysis [11,12,17,24]. Spectral sensors can provide different surface reflections of PWD that can help evaluate the regions of interest better than the naked eye. For example, near-infrared (NIR) is a subset of the infrared band which covers the wavelength range from 780 to 1400 nm. The fusion of the NIR and the red band can reflect the changes in photosynthesis [25]. However, multi-spectral cameras are costly and more unstable than general RGB cameras, as their image quality is affected by various environmental conditions.

As PWD becomes a more serious problem worldwide, machine learning technology is increasingly being used for its automatic detection. The most common method for the early diagnosis of the disease depends on UAV images. In this method [18], a high-quality orthophotograph is generally cropped into small pieces, and a conventional supervised classifier is applied to recognize the disease location in a limited region. In another study, ref. [26] introduced a method using simple classifiers like multi-layer perceptron (MLP) and Support Vector Machine (SVM) to distinguish the regions of healthy or PWD-infected trees.

2.2. Deep Learning-Based PWD Detection

Several deep neural architectures have been proposed in recent years, and these have achieved significant breakthroughs in diverse problem domains. Deep learning-based object detection is a challenging multi-task problem which involves assigning a class label to an object of interest and then learning to locate the object's position. There have been two main streams of research related to the object detection model. The first is the region of interest (ROI) driven method, where the DNN (deep neural network) filters out the irrelevant background and then leaves the rest for refined classification. The typical example is Faster R-CNN [27], which consists of an encoder (CNN layer), Region Proposal Network (RPN), ROI pooling, and a classifier. The encoder network extracts various feature maps by convolution mapping of the input sample into latent space. The RPN first generates thousands of candidate bounding boxes from the feature map, then a simple classifier filters out the negative bounding boxes (i.e., background) while retaining the more probable positive bounding boxes (i.e., foreground). ROI pooling [28] collects feature maps

in the positive bounding box and pools them into the same size. Finally, the bounding box regression block captures the precise location, and the classifier blocks divide the ROIs into specific categories. These two-stage object detection methods are relatively slow for real-time problems. That is why another types of single-stage object detection methods have been proposed. The well-known single-stage architectures include the YOLO series [29–32] and the SSD series [33,34]. In the single-stage method, there is no RPN, and the DNN has the ability to simultaneously localize and classify the target object. Therefore, using a single-stage detector network significantly reduces network complexity and speeds up the inference process. However, it still fails to provide more accurate position and classification information than the two-stage ROI driven method.

Some researchers have used a deep learning-based object detection for PWD. Reference [13] presented a two-stage object detection method that uses UAV remote sensing images to locate PWD-infected trees. Reference [19] compared the performance of single and two-stage object detection methods for PWD. In addition, Reference [20] presented a dataset with multi-band images and built a spatial-context-attention network (SCANet) with an expanded receptive field to better utilize context information. Further, Reference [21] proposed a faster disease filtering method that employs a lightweight one-stage detection model that discards a large number of irrelevant images before classifying the rest.

The previous studies devoted to PWD detection have suffered from a lack of data to train the DNN. By contrast, the DNN in our PWD detection method was trained on a large and previously-unused dataset, and our method is therefore more generalized and robust. A large number of “disease-like” objects push our object detector to build clear boundaries to distinguish them. Compared to the existing methods, our proposed method is more precise and rigorous in a real-world scenario.

3. Methodology

3.1. System Overview

The main challenges associated with PWD detection are a lack of data for training the DNN in various stages of infection, and the existence of background objects that are similar to the object of interest. We collected a large dataset consisting of observations from various districts in South Korea. The suppression of ambiguous background objects is particularly important in this domain [21]. The hard negative samples for PWD were further divided into six categories according to their appearance and texture information. The problem-solving strategy was found to be beneficial for guiding an object detector, particularly in instances where the object of interest was highly correlated with other background objects.

As shown in Figure 1, we first augmented the training samples with suitable augmentation methods and then trained our first DNN, which was designed to distinguish the background and PWD objects. The detector was robust at detecting most background regions, including healthy trees, land, buildings, lakes, etc., but PWD-infected trees confused with “disease-like” areas (FP) such as yellow land or maple trees. In this phase, it is easy to collect a lot of “disease-like” areas, which is referred to as the mining of hard negative samples [35]. We further categorized those ambiguous FP samples into six distinct categories (Figure 2). The disease and “disease-like” samples were passed through the DNN to perform a fine-level object detection. The UAV images in our system included disease regions that varied in size from 12×8 to 360×300 pixels. Accordingly, we applied a feature pyramid network (FPN) to capture arbitrary scale features based on both bottom-up and top-down connections. ResNet was selected as a backbone network, and the features in each residual block were processed for pyramidal representation. The bottom-up pathway produced the feature map hierarchy and the top-down pathway fused higher resolution features through upsampling the spatially coarse ones. This combined bottom-up and top-down process helps generate semantically stronger features. For the feature maps in each hierarchical stage, we appended a 3×3 convolution layer to reduce the aliasing effect of upsampling. Then, the set of merged features of each FPN stage was finally used for pre-

dictions. In the inference stage, we cropped the 800×800 image patch from the large sized orthophotograph (with “*.tif” extension). We assembled the results of augmented inference images and fused them by weighted boxes fusion algorithm to enhance the location and classification accuracies. A more detailed description of each module is provided later.

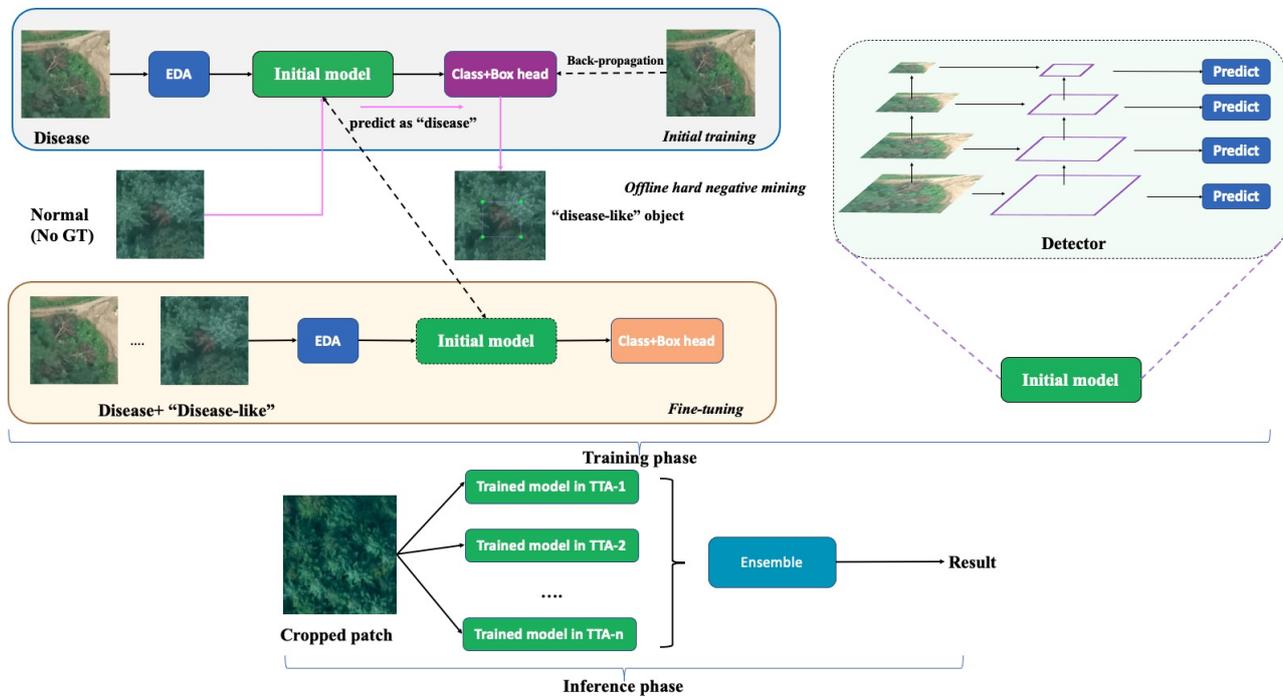


Figure 1. Pipeline overview.

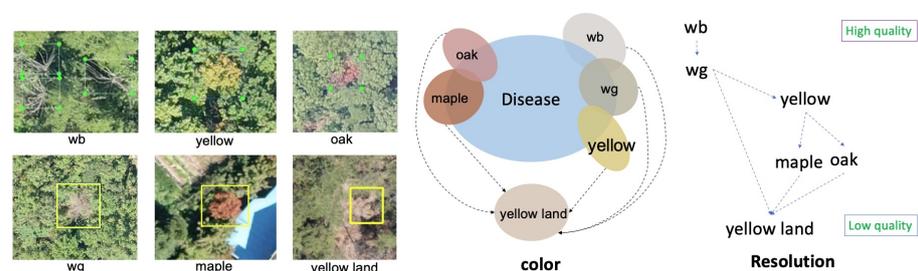


Figure 2. Left figure: the object of hard negative samples. Middle figure: the appearance relationship among “disease-like” objects and ground truth disease. Right figure: the reasons that the network predicted the different types of false negative samples. In high-quality image, branches and leaves are clear and easily identifiable. However, in low-resolution image, the real PWD-infected trees appear similar to the maple tree and the yellow land.

3.2. Efficient Data Augmentation (EDA)

Data augmentation is a well-known strategy for significantly increasing the amount of data available for training models without having to collect new data. It acts as a regularizer to reduce bias and generalize the system capability. Real-world UAV images are highly sensitive and collecting them is time-consuming. The image quality of UAV images is dependent on several environmental factors, including reflected light, contrast effects, and camera shake. Meanwhile, detection accuracy is affected by natural weather phenomena such as clouds or thick haze. We used several advanced augmentation methods to improve the performance for locating the PWD in different imaging situations.

Geometric transformation: We first cropped 800×800 patches from large “*.tif” images (more than 6×10^8 pixels) and applied random horizontal and vertical flip, rotation ($0 \sim 90$ degree), and resizing ($0.9 \sim 2.0$ times zooming) to augment the images and strengthen the model’s ability to handle the various resolutions/shooting angles.

Color space augmentation: The outward appearance of PWD varies based on the stage of the disease; diseased trees are grayish-green in color in the early stage, with the needles turning brown, and eventually ash grey. In the middle stage of the disease, the color of the diseased leaf (brown) resembles the color of a maple leaf. We carried out random gamma, brightness, and contrast adjustments as well as PCA color augmentation [36] to generate synthetic images from real ones. The trained model with the augmented data including these synthetic images was less sensitive to color and focused more on texture discriminative features.

Noise injection: Poor-quality photography devices mounted on UAV often have unavoidable shot noise from unwanted electrical fluctuation that occurs when taking the pictures. We simulated this condition by adding random Gaussian noise (mean of 0~0.3, std set to 1) to real images. Moreover, the device can be affected by radio interference, and a damaged image may randomly lose information, where this missing data appears as irregular black blocks. To address this problem, we augmented the training data by adopting the robust regularization technique [37] (cutout) to randomly remove various regions of input.

Other augmentations: Clouds may block sunlight and create a dark region (shadow) in an image. We added a mask to the original image to change the brightness of the local area of the image so as to simulate shadows. Furthermore, haze is more commonly seen in mountainous areas is another special natural phenomenon that reduces visibility on PWD-infected trees. Changing the opacity of image can generate synthetic haze. We augmented our images to maintain the performance of the network under poor weather conditions.

3.3. The Hard Negative Mining Algorithm

Hard negative mining (HNM) is a bootstrapping method that has been widely used in the classification field and which improves network performance by focusing on hard training samples. We improved the algorithm and applied it to the PWD detection problem. The modified method proceeds according to the following steps: (a) The object detection network is trained on the supervised training dataset. (b) The trained neural network is used to predict the unseen samples (no PWD samples) that are not included in the training samples. (c) The network predicts the objects of interest including “disease-like” objects which can be relabeled into several categories. (d) The “disease-like” objects are merged with genuine PWD-infected objects, and the neural network is retrained on the new dataset. The workflow of selecting “disease-like” objects is shown Figure 3. Initially, the large-sized orthophotograph (*.tif) is divided into small pieces, where the image without GT samples is kept and fed into the trained detector. Then, the model automatically filters out the easily recognizable objects (confidence score < 0.7), and the expert manually relabels the remaining hard objects according to the texture information.

The network with only three categories (background, disease, and hard negative samples) did not perform well due to the ambiguous classification decision boundaries. We manually annotated the hard negative samples into different categories. Figure 2 shows the division of “disease-like” objects and their relationships with the disease in terms of resolution and color. White branch (wb) denotes a dead tree that has a radial umbrella shape. The white-green (wg), yellow, and maple trees have a homogeneous color similar to PWD in its early and middle stages. The oak category indicates the presence of oak tree disease symptoms that resemble PWD-infected trees in UAV images. We also categorize yellow land into a separate category because this can appear similar to PWD, especially in low-resolution images.

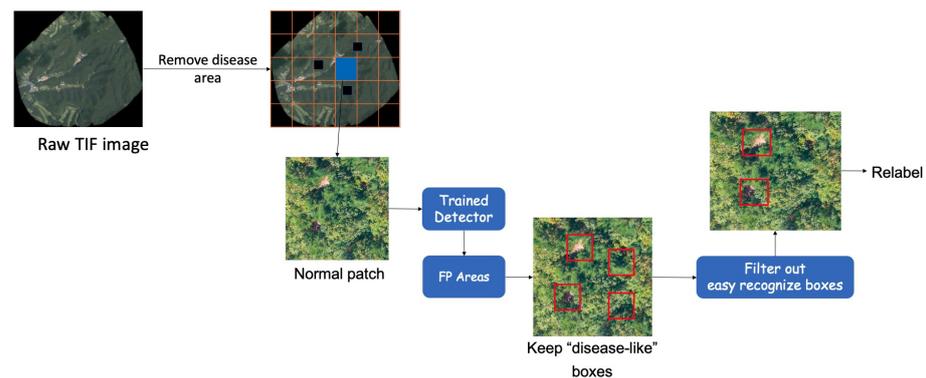


Figure 3. Extraction of “disease-like” objects.

3.4. Test Time Augmentation

Data augmentation is a common technique for increasing the size of a training dataset and reducing the chances of overfitting. Meanwhile, test time augmentation (TTA) is a data augmentation method applied during test time to improve the prediction capability of a neural network. As shown in the pipeline in Figure 4, we created multiple augmented copies from a sample image and then made predictions for the original and synthetic samples. The prediction results show the bounding box coordinates with corresponding confidence scores from different augmented samples as well as their original test images. TTA operation is an ensemble method in which multiple augmented samples of a test image are evaluated based on a trained model. The final decision is made by weighted boxes fusion algorithm. While pursuing a balance between computation complexity and performance, we pick three augmentation methods (horizontal flip, vertical flip, and 90-degree rotation) to evaluate the improvement achieved by the TTA method.

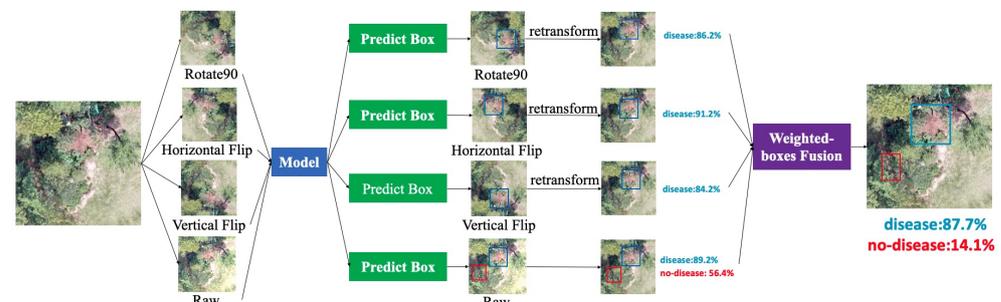


Figure 4. Test time augmentation applied in trained model. The bounding box prediction results are fused through weighted boxes fusion to highlight the important region and suppress the false detection.

3.5. Weighted Boxes Fusion Algorithm

Weighted boxes fusion (WBF) [38] is a key step while efficiently merging the predict position and the confidence score in TTA. Some common bounding box fusion algorithms such as NMS and soft-NMS [39] also work well for selecting bounding boxes by removing low-threshold overlapping boxes. However, they fail to consider the importance of different predicted bounding boxes. The WBF method will not discard any bounding boxes; instead, it uses the classification confidence scores of each predicted box to produce a combined predicted rectangle with high quality. The detailed WBF algorithm is described in Table 2, and the notations used in the table are summarized as follows:

- **IoU**: Intersection over union.
- **BBox**: Bounding box.
- **N**: Augmentation methods (i.e., horizontal flip, vertical flip).
- **B**: Empty list to store predicted boxes from N augmentation methods.
- **C**: Confidence score of predicted boxes.

- **THR**: Mini bounding box overlap threshold.
- **L**: Empty list to save cluster predicted boxes in one position “**pos**”, “**pos**” in one position, and **IoU** of two predicted BBox > **THR** (largely overlap).
- **F**: Empty list to store fused L boxes in different “**pos**”.
- **T**: Total number of predicted bounding boxes.

Table 2. Weighted boxes fusion for merging BBox in test time augmentation method.

Step 1: Initialize the list **L**, **B**, **F**, inference **N** methods and store predicted BBox into **B**.

Step 2: Iterate BBox in **B**, find matching BBox in **F**, if **IoU** (F_i, B_i) > **THR**.

– **Step 2.1:** If no matching BBox, dequeue the BBox in **B** and add it into **L**, **F**.

– **Step 2.2:** If the match BBox is found, dequeue the BBox in **B** and add it into **L**. This BBox represents the matching box in “**pos**” position in **F**.

Step 3: In each “**pos**” in **F**, recalculate the all the BBox confidence score and coordinate by formula Equations (1)–(5), weighted sum of the coordinates of the boxes, in here, weight is the confidence score for each BBox. The weight of high confidence score large than small one.

$$C_i = \frac{\sum_{i=1}^T C_i}{T} \quad (1)$$

$$x_{min} = \frac{\sum_{i=1}^T C_i * x_{min}}{\sum_{i=1}^T C_i} \quad (2)$$

$$x_{max} = \frac{\sum_{i=1}^T C_i * x_{max}}{\sum_{i=1}^T C_i} \quad (3)$$

$$y_{min} = \frac{\sum_{i=1}^T C_i * y_{min}}{\sum_{i=1}^T C_i} \quad (4)$$

$$y_{max} = \frac{\sum_{i=1}^T C_i * y_{max}}{\sum_{i=1}^T C_i} \quad (5)$$

Step 4: Readjust the confidence score as $C = C * \frac{T}{N}$, If there are few BBox in “**pos**” in the same position, the detected region is less like GT categories.

4. Experiment

4.1. Pine Wilt Disease-Infected Image Acquisition

Data acquisition is a primary task in training a robust DNN for classification or regression tasks. For this step, we collected diverse PWD data samples representing various stages of infection, as shown in Figure 5. Pine wilt can kill a pine tree between 40 days to a few months after infection, and an infected tree shows different symptoms according to its stage of infection. In the early stage, the needles will remain green, but the accumulation of terpenes in xylem tissue results in cavitation, which interrupts water flux in the pine trees. In the second stage, the tree can no longer move water upward, thus causing it to wilt and the needles to turn yellow. The pinewood nematodes then grow in number, and all needles turn yellow-brown or reddish-brown. The disease progresses uniformly branch by branch. After the whole tree is fully dead, the tree needles still remain in place without falling, and show white bare color.

In this work, we obtained UAV images using drone technology. Between August and September, we took orthophotographs with UAVs in the disease-prone region. We tried to capture high-resolution images to best observe the changes in color caused by the disease. We captured high-resolution images from a low altitude using a CMOS camera (Sony Rx R12). The terrain level changes frequently within a hilly region, which affects the resolution quality of an orthophotograph. The images were sequentially taken while overlapping the area. We used drone mapping software (Pixel4Dmapper) to recorrect the GIS coordinates and ensure that the orthophotograph GSD was between 3.2 cm/pixel and 10.2 cm/pixel, where all overlapped patches composed a large orthophotograph (“*.tif”). Furthermore, each large orthophotograph had an ESRI (Environmental Systems Research Institute) format output with geographic coordinates that specify trees that have been deemed to be potentially infected with PWD by experts. During the training, we converted those geographic coordinates to a bounding box annotation. Then, we conducted a large

orthophotograph crop to small 800×800 patches and sent the data to train the network with GT. After the data cleaning, we obtain a total of 4836 images with 6121 PWD damaged tree points as well as 265,694 normal patches (river, roof, field, etc. no PWD images) used to extract the “disease-like” objects. We used five-fold cross-validation to evaluate our proposed system, where each fold included balanced samples of various resolutions. In the real-world scenarios test, we captured another 10 real-world orthophotographs and compared the results with the expert-labeled ground truth points (730 PWD-infected trees).

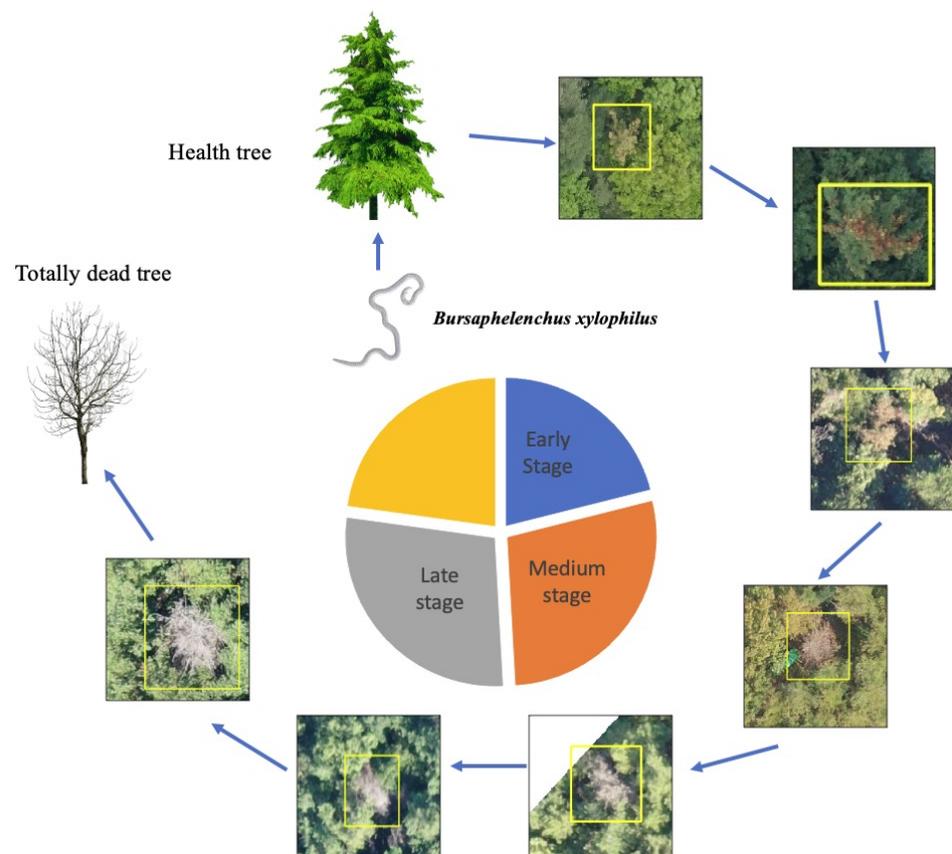


Figure 5. Symptoms of each stage of PWD infection.

4.2. Training Strategy

Our code was written in Python and the network models were implemented with PyTorch. A workstation with multiple TITAN XP GPUs and parallel processing was used to speed up the training. The COCO pre-trained model [40] was used for transfer learning, and the network was warmed up with a 1.0×10^{-6} learning rate for the first epoch to reduce the primacy effect. Then, we set a 0.001 learning rate and gradually reduced it by 50% every 50 epochs. In the experiment, the batch size was set to 8, the optimizer was SGD with 0.9 momentum for 200 training epochs. The K-means algorithm [30] was used to ensure the anchor scales and aspect ratios for fully covering the arbitrary disease shape. We use 5-fold cross-validation to split the training/validation dataset. In the first and second training stages, the strategies were the same except for the difference between the final classification headers. In the first stage, we trained the detection network with two neural nodes (categories) to distinguish between background and PWD-infected trees in the image. In the second stage, we fine-tuned the whole architecture for 200 epochs to distinguish actual PWD damaged trees as well as the other six categories of “disease-like” objects (e.g., wg, maple, wb, etc.).

4.3. Accuracy Estimation

Detector evaluation: We reported the average test accuracy of 5-fold cross-validation. The standard object detection metrics: mean average precision (*mAP*) and Recall were used to evaluate the performance of the model, and these are, respectively, expressed as Equations (6) and (7).

$$mAP = \frac{1}{|Q|} \sum_{q=1}^Q AP(q) \quad (6)$$

$$Recall = \frac{tp}{tp + fn} \quad (7)$$

where Q is the number of queries in the set and $AP(q)$ is the mean of the average precision scores for the given query. Our goal is to identify potential PWD-infected trees as much as possible, and we use Recall as an improvement metric to evaluate how much potential disease area has been located. Recall is assessed as the number of correctly detect diseases (TP) divided by the number of total diseases.

The setup for real-word environment evaluation: Operating in the real-world is even more challenging, as the cropped patches contain not only the PWD-infected trees but also various background content (including “disease-like” objects). Real-world PWD objects are small, irregular, and distributed across a wide area, and the GT is point-wise annotation (x, y geographic coordinates). Converting the point-wise annotation into bounding box annotation is challenging. Our goal was to correctly identify as many disease objects as possible along with their precise positions. In the field investigation to locate the disease-infected tree, an offset error of less than 8 m was deemed acceptable, so we use the x, y geographic coordinates of disease to generate 8×8 m GT bounding box annotation for each hotspot. We assume that the disease was found correctly when the overlapped area of GT and predicted bounding box (IoU: Intersection of Union) is larger than 0.3; otherwise, the system prediction is considered to be a false detection.

Specially, we use the overlap strategy to scan the orthophotograph and crop the test patches with 25% overlap. The IoU threshold is set to 0.4 for NMS, and 0.6 for merging the predicted bounding boxes in TTA process. Due to the insufficient context information, FP typically appears on the edge of the test patch. Therefore, we remove the bounding boxes that are less than 5 pixels from the boundary to reduce false alarms; for more information, please refer to Section 5.1.2.

5. Results

We compared our proposed system to various network architectures. Table 3 compares the performance of our proposed method to those of alternative structures and backbone networks. The *mAP* accuracy refers to the average mean of five-fold cross validation in the same hyperparameter setting. We found that FPN with a ResNet101 backbone outperformed other structures, as it obtained the best accuracy of 89.44%; this is attributed to the fact that the fusion of bottom-up and top-down features helps capture disease in an arbitrary shape. The EDA scheme improved system performance as much as 2%. EDA overcomes the issue of data shortages and reduces the environmental effects of imaging. However, other findings were more surprising: the HNM achieved 88.35% accuracy in the FPN + Res101 architecture, and it improved by more than 3% compared to no-HNM approach. This phenomenon also occurs in other structures, which implies that the process in our HNM of splitting “disease-like” objects is important in learning the discriminative features. Another strategy we adopted to improve the inference power was the TTA method. TTA helped eliminate incorrect predictions and boost system performance as much as 1%. The improvement was noticeable in all network structures, as presented in Table 3. The RetinaNet architecture performed worse than the FPN structure. We suspect that this was caused by the hyperparameters selected in focal loss [41]. Focal loss has two hyperparameters, α, γ , $FL(p_t) = -\alpha(1 - p_t)^\gamma \log(p_t)$, where α controls the balance

of positive and negative samples and γ adjusts the weight of learning easy and difficult samples; increasing γ makes the model pay more attention to difficult samples. Tuning the two hyperparameter is a challenge task. The goal of this experiment was to evaluate the merit of our proposed strategies, and the results show that they consistently increase the performance of conventional architecture. Further, researchers can apply our proposed strategies in their designed network architectures and obtain better performance. For saving time, For saving time, we used FPN + Res101 structure in the following experiments.

Table 3. The mAP accuracy while using different backbone structures.

Backbone	Structure	Baseline	Baseline + EDA	Baseline + EDA + HNM	Baseline + EDA + HNM + TTA
Res50	Faster R-CNN	0.8248	0.8365	0.8609	0.8689
Res50	FPN	0.8286	0.8419	0.8688	0.8734
Res101	FPN	0.8312	0.8521	0.8835	0.8944
Res50	SSD	0.8169	0.8316	0.8522	0.8619
Res50	RetinaNet	0.8255	0.8445	0.8658	0.8712

5.1. Result of Real-World Environment Evaluation

5.1.1. Software Integration

The real-world inference pipeline of our disease detection system is demonstrated in Figure 6. The UAV with an RGB camera first captured an overlapped image in the survey area. We then processed these images and integrated them into a large orthophotograph. The geographic coordinates are possibly varied due to changes in camera distance, so they had to be carefully recorrected for each object to be kept in its proper position. The resulting 8-bit “*.tif” image was further processed by the GDAL (<https://gdal.org>, accessed on 20 December 2021) library and cropped into 800×800 patches with overlap. The overlap is essential because the boundary region of the patch has few context information which leads to wrong detection results. We then located the presence of disease in each patch and indicated their location with a bounding box through the inference process.

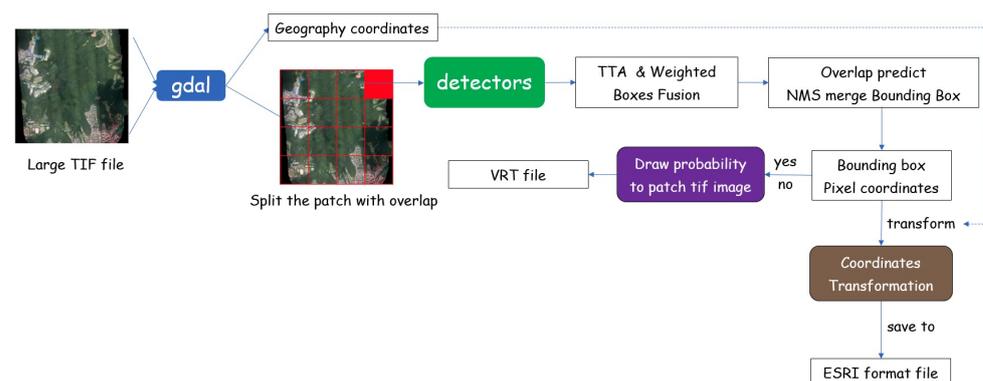


Figure 6. Pipeline of the automatic PWD diagnosis system.

For each input image patch, we generated augmented patches and returned the combined prediction result from TTA. The models provided the predicted bounding box, which had four points to construct a rectangle: the x and y pixel coordinates for the top left corner as well as the corresponding coordinates for the bottom right corner. Next, we used WBF to select the proper bounding box as well as reduce the number of irrelevant detected boxes in multiple results. After obtaining the precise bounding box location for the disease areas, we transformed the pixel coordinates to geographic coordinates based on the coordinate reference system, and stored the results to an output file. Our program produces the output in ESRI format, which includes four types of files—*.dbf, *.prj, *.shp,

*.shx and can be imported into a GIS application for visualization. This application helps experts find the proper GPS coordinates of a potential disease outbreak, and the latitude and longitude information are useful for further field investigation.

5.1.2. The Software Integration Hyperparameter Selection

The selection of proper hyperparameters is critical in the pursuit of better detection software. For real environment evaluations, the important hyperparameters are the stride size, overlap ratio, IoU threshold for NMS, number of augmented methods during TTA, IoU threshold for WBF, and bounding box distance (RBD). Table 4 lists the hyperparameter settings we used to find potential diseases in Goomisi Goaeup (Table 5). The stride is the number of pixels that shift over for cropped patches in the next inference time; for example, 3/4 means that, in the next patch, 3/4 pixels ($800 \times 3/4 = 600$) were moved, thus leaving 200 pixels overlapping in the horizontal and vertical directions. The small stride contributes to the increased overlap areas and potentially increases the inference time. RBD refers to the length of the predicted bounding box all the way up to the edge of the cropped patch. The bounding box was removed if the distance between the bounding boxes and the edge of the patch was less than RBD. In our experiment, we obtained a performance improvement of around 7% after removing the bounding boxes near the edge. The reason for this is a lack of context information in the marginal area, which caused the detector to frequently mislabel “disease-like” objects (wg, wb, yellow land, etc.) as disease. Our overlap strategy ensures that there is an overlapped area between the current patch and the next crop patch, so that the next crop patch contains rich context information by removing the bounding boxes located at the edge of the cropping area. Another hyperparameter was the threshold value for NMS. The two-stage object detector needed to generate a large number of candidate bounding boxes to locate the ROI regions, and the NMS was responsible for selecting the best one by filtering out the low confidence scored boxes which have higher IoU values than threshold. The next one was the proper selection of augmented methods in TTA; we used three geometric augmentation methods (horizontal flip, vertical flip, and 90-degree rotation) to evaluate the trade-off between performance and computation complexity. We also tested different IoU thresholds in WBF. In general, we achieved an improvement of 9% by employing proper parameters.

Table 4. Hyperparameter selection in software integration.

Stride (Image)	RBD	NMS (IoU Thre)	TTA	WBF (IoU_Thre)	mAP
1	-	-	-	-	0.7494
	Not Remove	0.50	-	-	0.7650
3/4	0	0.50	-	-	0.8021
	5	0.50	-	-	0.8315
	10	0.50	-	-	0.8306
3/4	5	0.40	-	-	0.8320
	5	0.45	-	-	0.8315
1/2	5	0.60	-	-	0.8297
1/4	5	0.40	-	-	0.8184
	5	0.40	1	0.70	0.8337
3/4	5	0.40	1	0.60	0.8373
	5	0.40	1	0.50	0.8037
	5	0.40	2	0.60	0.8352
	5	0.40	3	0.60	0.8336

Table 5. Real environment evaluation results.

Name	Resolution (pixel)	GSD (cm/pixel)	GT	Predict	FP	Recall
① Namyangjusi Sudongmyeon	31,495 × 30,527	6.79	34	33	21	0.97
② Namyangjusi Joanmyeon	22,317 × 32,501	6.19	50	47	70	0.94
③ Kwangjusi Docheokmyeon	33,775 × 34,618	6.09	7	5	43	0.71
④ Kwangjusi Chowoleup	35,310 × 25,731	6.12	9	9	33	1.00
⑤ Yangpyeonggun Yangdongmyeon	54,178 × 67,325	3.84	40	36	175	0.90
⑥ Goomisi Geoyidong	41,940 × 44,949	5.22	50	49	80	0.98
⑦ Goomisi Goaeup	50,689 × 37,427	5.21	110	108	90	0.98
⑧ Goomisi Sangdongmyeon	60,438 × 78,747	4.91	305	302	210	0.99
⑨ Goomisi myeon	23,630 × 27,099	7.32	48	47	62	0.98
⑩ Milyangsi Yongpyeongdong	47,612 × 51,334	5.40	77	75	150	0.97
			730	711	934	0.97

5.1.3. Evaluation in Real-World Dataset

We captured another 10 orthophotographs (Appendix A) from different cities in South Korea to test the reliability of the proposed system. Each orthophotograph has more than $30,000 \times 20,000$ pixels ($360,000 \text{ m}^2$ in GSD 6.00 cm/pixel) area (Table 5). We followed the same preprocessing method as that described in the “software integration” section, and the PWD detector captured 711 out of 730 PWD-infected trees in various resolutions (Table 5). To show the robustness of our network, we draw a portion of an orthophotograph in Figure 7. The potential infected pine trees (GT) are labeled by red dots. The blue bounding box denotes the inference result obtained by the trained detector. The green panel shows the sample of TPs, which has various symptoms of PWD-infected trees across the early to late stage. The red panel shows the false-detected PWD-infected trees. Distinguishing those “disease-like” objects in RGB channels remains a challenging task due to the ambiguity in both shape and color. Further investigations using either multi-spectral images or field investigations are needed.

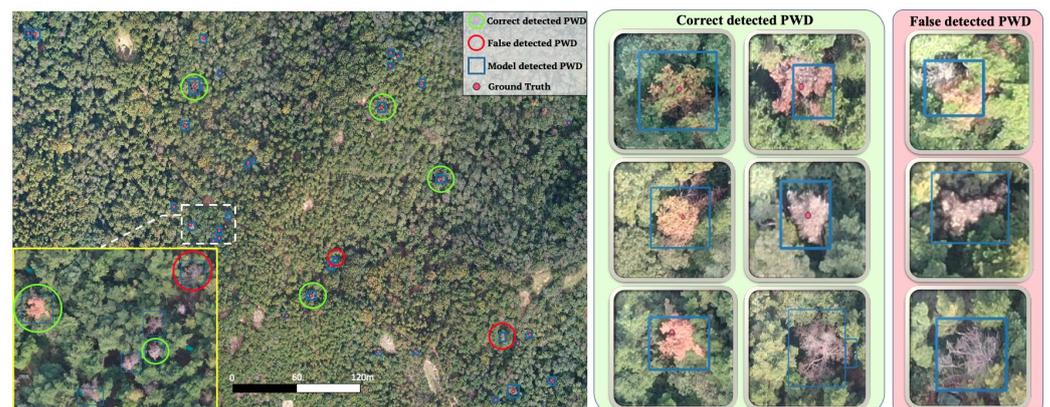


Figure 7. Prediction result using the Goomisi Goaeup dataset described in Table 5. The bottom left corner shows a magnification of the content in the white dash box. We selected some predicted bounding boxes and marked them with circles. The instances of enlarged disease are shown in the right figure; the green panel shows correctly detected PWD while the red panel represents the false alarms that show an appearance similar to that of a diseased tree.

5.2. The Effect of Hard Negative Mining

HNM was proposed to alleviate the high variance and irreducible error in cases of limited training samples. When the model only sees a few types of disease symptom from a limited number of samples, a large number of background regions makes the detector

liable to over-study, whereby it tends to map the true target into no disease. HNM provide many ambiguous “disease-like” objects which share a similar pattern with the disease in the middle and late stages. Including these ambiguous objects helps the model build clear boundaries by learning more discriminative features. In addition, the generated hard negative samples contain a lot of “out of interest regions” (background) such as highways, broadleaf forests, farmland, etc.; this diversity in the training data generalizes the system’s ability to correctly classify real-world problems.

Figure 8 shows some examples of the stark contrast that occurred when we applied finetuning with six “disease-like” categories. HNM successfully suppresses the confidence score of ambiguous objects while locating real disease well. The “wb” category (first column) signifies no PWD-infected dead trees; “wb” and PWD-infected dead trees differ only slightly in the branches. Including “wb” reduces the number of FP bounding boxes and possibly well guides the network to update the network for real PWD-infected dead tree by lowering the confidence score of FPs. Moreover, low-resolution images leads to confusion in identifying the yellow land and small PWD objects. The new category “yellow land” alleviates the effect of ground (confidence score 0.960 \rightarrow 0.003 in column (2)). The same phenomenon happens with the “maple” category. Due to the loss of water in the late disease stage, PWD-infected trees tend to show a red-brown color, thus appearing similar to maple trees. Without HNM, the network predicted maple trees (column 6, first row) as PWD-infected disease (high confidence score), but including the additional maple samples promoted the learning ability to filter errors (confidence score 0.893 \rightarrow 0.630).

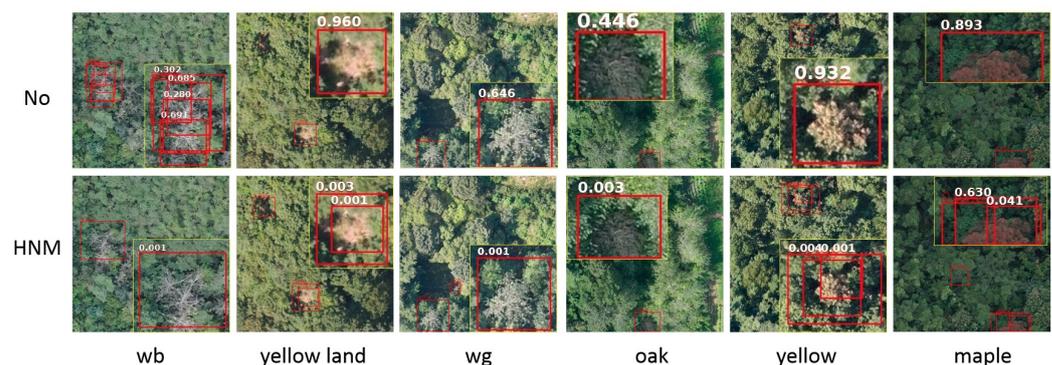
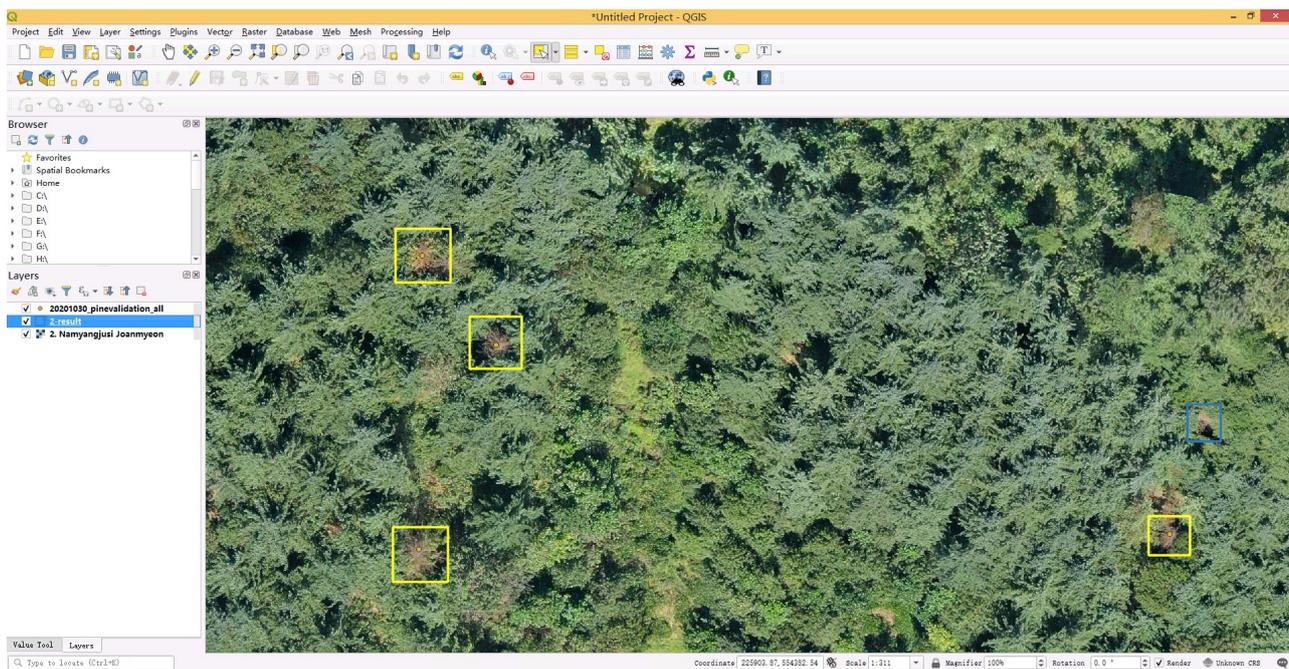


Figure 8. Advantage of using hard negative mining. The detector bounding boxes are the “disease-like” object (FP). For convenience, we zoomed out from the region of interest to highlight the difference when hard negative mining is applied.

6. GIS Application Visualization

We developed a system with which to process a large orthophotograph (“*.tif”) to predict potential PWD-infected trees, as shown in Figure 6. The system automatically detects PWD location and saves the location information in standard ESRI format files. The developed system was integrated with QGIS (<https://qgis.org>, accessed on 20 December 2021) to visualize the input “*.tif” images and potential disease locations (Figure 9). Potential disease regions with a high confidence score are represented by yellow bounding boxes while “disease-like” objects are illustrated by blue bounding boxes. The output file preserves class label, predicted confidence score, as well as left-top and right-bottom position for the target bounding box. As shown in Figure 9b, the column score represents the confidence score within [0,1], and it indicates how much a tree looks like a PWD-infected tree. The expert can reduce the threshold to find more potential disease spots. The columns lr_i and rb_i include the coordinate values of a bounding box in the image coordinate system. The columns lt and rb represent the GPS coordinates in a coordinate reference system [42] (EPSG: 5186 Korean 2000/Central Belt 2010). Using the lt and rb values, expert can locate PWD-infected trees in field investigation.



(a) GIS application

	id	label	score	lt	rb	lt_i	rb_i
1	0	disease	0.99315	(323248.080534411, 398709.26759753696)	(323255.648722411, 398701.595020737)	(19114.0, 34716.0)	(19261.0000000000)
2	1	disease	0.99169	(323365.517934411, 398650.75767513696)	(323375.22609281103, 398641.467071937)	(20235.0, 36966.0)	(20413.0, 37152.0)
3	10	disease	0.98686	(322307.798418411, 398934.434239137)	(322313.852968811, 398928.11871673696)	(14800.0, 16701.0)	(14921.0, 16817.0)
4	100	disease	0.82681	(322050.375637611, 398398.81530633697)	(322053.559496011, 398395.683642337)	(25062.0, 11769.0)	(25122.0, 11830.0)
5	101	disease	0.81994	(322556.243762411, 398839.336042337)	(322562.402701611, 398833.594658337)	(16622.0, 21461.0)	(16732.0, 21579.0)
6	102	disease	0.81827	(321973.075731211, 398764.176106337)	(321982.470723211, 398754.937697537)	(18062.0, 10288.0)	(18239.0, 10468.0)
7	103	disease	0.81568	(323270.680709611, 398695.64485913696)	(323276.683065611, 398689.590308737)	(19375.0, 35149.0)	(19491.0, 35264.0)
8	104	disease	0.81083	(323477.474922411, 398469.590912737)	(323481.650474411, 398465.467555137)	(23706.0, 39111.0)	(23785.0, 39191.0)
9	105	disease	0.8044	(323287.800472811, 398817.62317193695)	(323294.898911211, 398810.68131673697)	(17038.0, 35477.0)	(17171.0, 35613.0)
10	106	disease	0.79423	(322723.944369611, 398408.836631137)	(322729.842336811, 398402.468914337)	(24869.999999999996, 24...	(24992.0, 24787.0)

(b) Predict results

Figure 9. Visualization of the inference result in the QGIS program.

7. Discussion and Future Work

In this paper, we proposed a system for improving the performance of the object detection model that detects PWD-infected trees using RGB-based UAV images. To learn a robust network, we created a large dataset which contains a total of 6121 disease spots from various infected stages and areas. The comparison results show that our proposed system has great consistency across different backbone structures. HNM can select “disease-like” objects from six categories. Trained and fine-tuned networks successfully built better decision boundaries with which to distinguish true PWD objects from the six “disease-like” ones. EDA and TTA achieved significant gains by alleviating the data bias problem. In addition, 711 out of 730 PWD-infected trees were identified in 10 large size orthophotographs, indicating that this method shows great potential in locating PWD in various pine forest resources. Finally, the integrated software can automatically locate potential PWD-infected locations and save to ESRI format. It is also convenient to visualize the results in the GIS application for field investigation.

However, there is still work to be done. For example, the best way to utilize context information during the training remains unclear. PWD only infects the pine family, so tree species classification will help reduce inference time and make it easier to precisely locate

infected regions. The other problem to be overcome is the method of filtering low-quality images. In UAV images, it is difficult to ensure a consistent resolution, implying that poor-resolution images with ambiguous features decrease the performance. Further, RGB-based PWD detection method still has limitations, as it confuses PWD-infected trees with “disease-like” objects in the early and later stages. Reference [25] has demonstrated that PWD infected trees exhibit a reduction in normalized difference vegetation index (NDVI), and [43] showed the effectiveness of conifer broadleaf classification with multi-spectral image. These studies provide insight into the usage of multispectral information to aid in recognition. However, multi-spectral images typically have a lower resolution than RGB images, so the best way to collect and efficiently use multi-spectral images remains a question of interest. We believe our proposed method can be used as preprocessing stage to filter the irrelevant region as well as to find fuzzy PWD hotspot. It is only necessary to reanalyze suspected images with the multi-spectral image, which greatly reduces the time for data collection.

Author Contributions: J.Y. designed the study, conducted the experiments and prepared the original manuscript. R.Z. prepared the dataset. J.L. supervised the research and edited the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in Brain Korea 21 PLUS Project.

Institutional Review Board Statement: Not applicable.

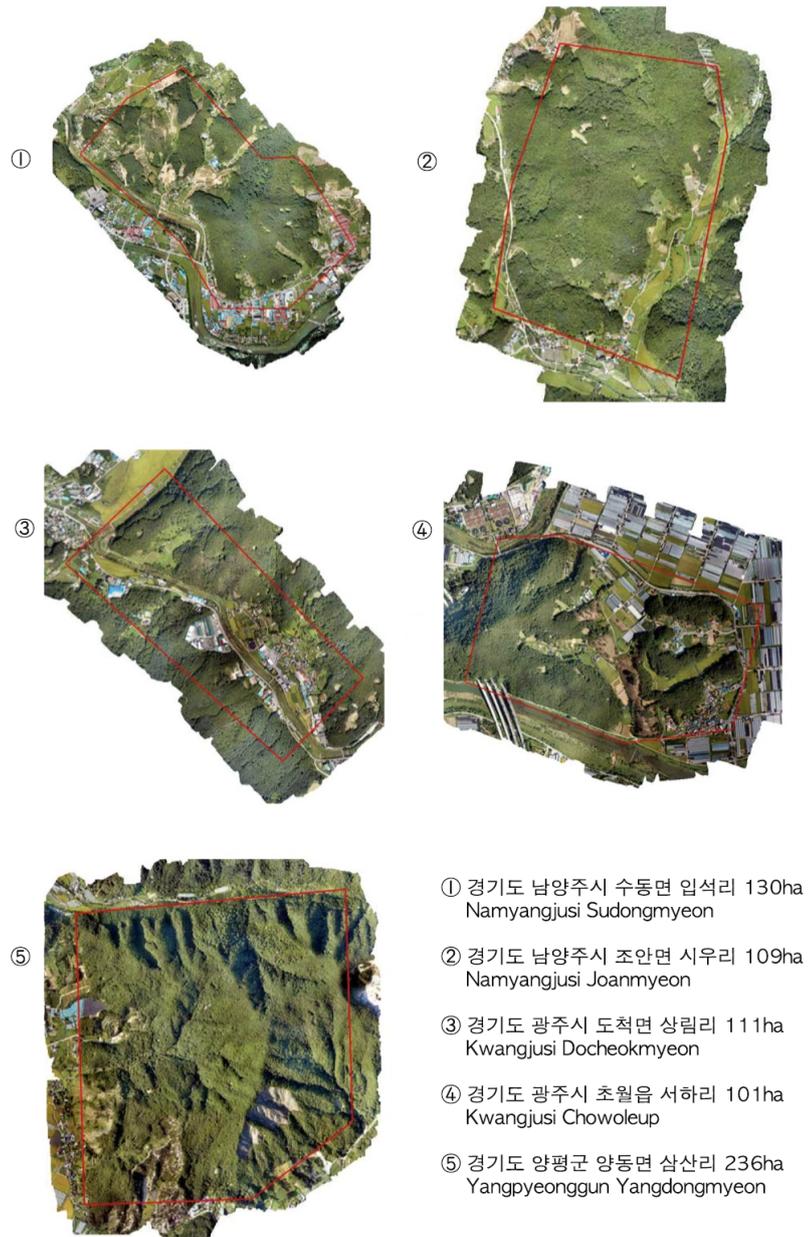
Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing is not applicable to this article.

Acknowledgments: The authors express special thanks to Korea Forestry Promotion Institute which has provided the data for experiments in the paper. I would like to give my sincere gratitude to Yagya Raj Pandeya and Bhuwan bhattacharai, gave me great help for grammar correction.

Conflicts of Interest: The authors declare that there is no conflict of interest.

Appendix A





References

1. Khan, M.A.; Ahmed, L.; Mandal, P.K.; Smith, R.; Haque, M. Modelling the dynamics of Pine Wilt Disease with asymptomatic carriers and optimal control. *Sci. Rep.* **2020**, *10*, 1–15. [[CrossRef](#)] [[PubMed](#)]
2. Hirata, A.; Nakamura, K.; Nakao, K.; Kominami, Y.; Tanaka, N.; Ohashi, H.; Takano, K.T.; Takeuchi, W.; Matsui, T. Potential distribution of pine wilt disease under future climate change scenarios. *PLoS ONE* **2017**, *12*, e0182837. [[CrossRef](#)] [[PubMed](#)]
3. Nevalainen, O.; Honkavaara, E.; Tuominen, S.; Viljanen, N.; Hakala, T.; Yu, X.; Hyyppä, J.; Saari, H.; Pölönen, I.; Imai, N.N. Individual tree detection and classification with UAV-based photogrammetric point clouds and hyperspectral imaging. *Remote Sens.* **2017**, *9*, 185. [[CrossRef](#)]
4. Nezami, S.; Khoramshahi, E.; Nevalainen, O.; Pölönen, I.; Honkavaara, E. Tree species classification of drone hyperspectral and rgb imagery with deep learning convolutional neural networks. *Remote Sens.* **2020**, *12*, 1070. [[CrossRef](#)]
5. Sothe, C.; Dalponte, M.; de Almeida, C.M.; Schimanski, M.B.; Lima, C.L.; Liesenberg, V.; Miyoshi, G.T.; Tommaselli, A.M.G. Tree species classification in a highly diverse subtropical forest integrating UAV-based photogrammetric point cloud and hyperspectral data. *Remote Sens.* **2019**, *11*, 1338. [[CrossRef](#)]
6. Egli, S.; Höpke, M. CNN-Based Tree Species Classification Using High Resolution RGB Image Data from Automated UAV Observations. *Remote Sens.* **2020**, *12*, 3892. [[CrossRef](#)]

7. Maldonado-Ramirez, S.L.; Schmale, D.G., III; Shields, E.J.; Bergstrom, G.C. The relative abundance of viable spores of *Gibberella zeae* in the planetary boundary layer suggests the role of long-distance transport in regional epidemics of Fusarium head blight. *Agric. For. Meteorol.* **2005**, *132*, 20–27. [[CrossRef](#)]
8. Nguyen, H.T.; Caceres, M.L.L.; Moritake, K.; Kentsch, S.; Shu, H.; Diez, Y. Individual Sick Fir Tree (*Abies mariesii*) Identification in Insect Infested Forests by Means of UAV Images and Deep Learning. *Remote Sens.* **2021**, *13*, 260. [[CrossRef](#)]
9. Arantes, B.H.T.; Moraes, V.H.; Geraldine, A.M.; Alves, T.M.; Albert, A.M.; da Silva, G.J.; Castoldi, G. Spectral detection of nematodes in soybean at flowering growth stage using unmanned aerial vehicles. *Ciência Rural* **2021**, *51*. [[CrossRef](#)]
10. Xiao, Y.; Dong, Y.; Huang, W.; Liu, L.; Ma, H. Wheat Fusarium Head Blight Detection Using UAV-Based Spectral and Texture Features in Optimal Window Size. *Remote Sens.* **2021**, *13*, 2437. [[CrossRef](#)]
11. Takenaka, Y.; Katoh, M.; Denga, S.; Cheunga, K. Detecting forests damaged by pine wilt disease at the individual tree level using airborne laser data and worldview-2/3 images over two seasons. In Proceedings of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLII-3/W3, Jyväskylä, Finland, 25–27 October 2017.
12. Yu, R.; Luo, Y.; Zhou, Q.; Zhang, X.; Wu, D.; Ren, L. A machine learning algorithm to detect pine wilt disease using UAV-based hyperspectral imagery and LiDAR data at the tree level. *Int. J. Appl. Earth Obs. Geoinf.* **2021**, *101*, 102363. [[CrossRef](#)]
13. Deng, X.; Tong, Z.; Lan, Y.; Huang, Z. Detection and Location of Dead Trees with Pine Wilt Disease Based on Deep Learning and UAV Remote Sensing. *AgriEngineering* **2020**, *2*, 294–307. [[CrossRef](#)]
14. Zhao, Z.Q.; Zheng, P.; tao Xu, S.; Wu, X. Object detection with deep learning: A review. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3212–3232. [[CrossRef](#)] [[PubMed](#)]
15. Sung, K.K.; Poggio, T. Example-based learning for view-based human face detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1998**, *20*, 39–51. [[CrossRef](#)]
16. Felzenszwalb, P.F.; Girshick, R.B.; McAllester, D.; Ramanan, D. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *32*, 1627–1645. [[CrossRef](#)] [[PubMed](#)]
17. Iordache, M.D.; Mantas, V.; Baltazar, E.; Pauly, K.; Lewycky, N. A Machine Learning Approach to Detecting Pine Wilt Disease Using Airborne Spectral Imagery. *Remote Sens.* **2020**, *12*, 2280. [[CrossRef](#)]
18. Syifa, M.; Park, S.J.; Lee, C.W. Detection of the Pine Wilt Disease Tree Candidates for Drone Remote Sensing Using Artificial Intelligence Techniques. *Engineering* **2020**, *6*, 919–926. [[CrossRef](#)]
19. Wu, B.; Liang, A.; Zhang, H.; Zhu, T.; Zou, Z.; Yang, D.; Tang, W.; Li, J.; Su, J. Application of conventional UAV-based high-throughput object detection to the early diagnosis of pine wilt disease by deep learning. *For. Ecol. Manag.* **2021**, *486*, 118986. [[CrossRef](#)]
20. Qin, J.; Wang, B.; Wu, Y.; Lu, Q.; Zhu, H. Identifying Pine Wood Nematode Disease Using UAV Images and Deep Learning Algorithms. *Remote Sens.* **2021**, *13*, 162. [[CrossRef](#)]
21. Li, F.; Liu, Z.; Shen, W.; Wang, Y.; Wang, Y.; Ge, C.; Sun, F.; Lan, P. A Remote Sensing and Airborne Edge-Computing Based Detection System for Pine Wilt Disease. *IEEE Access* **2021**, *9*, 66346–66360. [[CrossRef](#)]
22. Son, M.H.; Lee, W.K.; Lee, S.H.; Cho, H.K.; Lee, J.H. Natural spread pattern of damaged area by pine wilt disease using geostatistical analysis. *J. Korean Soc. For. Sci.* **2006**, *95*, 240–249.
23. Park, Y.S.; Chung, Y.J.; Moon, Y.S. Hazard ratings of pine forests to a pine wilt disease at two spatial scales (individual trees and stands) using self-organizing map and random forest. *Ecol. Inform.* **2013**, *13*, 40–46. [[CrossRef](#)]
24. Lee, J.B.; Kim, E.S.; Lee, S.H. An analysis of spectral pattern for detecting pine wilt disease using ground-based hyperspectral camera. *Korean J. Remote Sens.* **2014**, *30*, 665–675. [[CrossRef](#)]
25. Kim, S.R.; Lee, W.K.; Lim, C.H.; Kim, M.; Kafatos, M.C.; Lee, S.H.; Lee, S.S. Hyperspectral analysis of pine wilt disease to determine an optimal detection index. *Forests* **2018**, *9*, 115. [[CrossRef](#)]
26. Lee, S.; Park, S.J.; Baek, G.; Kim, H.; Lee, C.W. Detection of damaged pine tree by the pine wilt disease using UAV Image. *Korean J. Remote Sens.* **2019**, *35*, 359–373.
27. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149. [[CrossRef](#)] [[PubMed](#)]
28. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
29. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 30 June 2016.
30. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 26 July 2017.
31. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
32. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
33. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2016.
34. Liu, S.; Huang, D. Receptive field block net for accurate and fast object detection. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
35. Shrivastava, A.; Gupta, A.; Girshick, R. Training region-based object detectors with online hard example mining. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 30 June 2016.

36. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2017**, *60*, 84–90. [[CrossRef](#)]
37. DeVries, T.; Taylor, G.W. Improved regularization of convolutional neural networks with cutout. *arXiv* **2017**, arXiv:1708.04552.
38. Solovyev, R.; Wang, W.; Gabruseva, T. Weighted boxes fusion: Ensembling boxes from different object detection models. *Image Vis. Comput.* **2021**, *107*, 104117. [[CrossRef](#)]
39. Bodla, N.; Singh, B.; Chellappa, R.; Davis, L.S. Soft-NMS—improving object detection with one line of code. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.
40. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2014.
41. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.
42. Janssen, V. Understanding coordinate reference systems, datums and transformations. *Int. J. Geoinform.* **2009**, *5*, 41–53.
43. Abdollahnejad, A.; Panagiotidis, D. Tree Species Classification and Health Status Assessment for a Mixed Broadleaf-Conifer Forest with UAS Multispectral Imaging. *Remote Sens.* **2020**, *12*, 3722. [[CrossRef](#)]