Low-Cost Tree Health Categorization and Localization Using Drones and Machine Learning

Hovannes Kulhandjian[†], Brian Irineo[†], and Jacob Sales[†] and Michel Kulhandjian[‡]

[†]Department of Electrical and Computer Engineering, California State University, Fresno, Fresno, CA 93740, U.S.A.
E-mail: {hkulhandjian, 9brhino9, xenosalch}@mail.fresnostate.edu
[‡]Department of Electrical and Computer Engineering, Rice University, Houston, TX, 77005, U.S.A.
E-mail: michel.kulhandjian@rice.edu

Abstract—**In this work, we propose a low-cost tree health categorization and localization scheme using drones and machine learning. Crop data is collected and employed to inform agricultural decisions, with the ultimate goal of enhancing production. Our solution employs visual imaging obtained from unmanned aerial vehicles (UAVs), commonly known as drones, for the purpose of monitoring the well-being of fruit trees. A user activates the drone from the ground station, initiating a "snake-like" flight path through the orchard to capture images before landing. Subsequently, our models process the data, triggering an alert for any visually identified unhealthy trees. The system then furnishes the farmer with precise locations of these afflicted trees. Our artificial-intelligent (AI)-driven precision agriculture technology is poised to significantly streamline the process and reduce costs for farmers by swiftly identifying and categorizing both unhealthy and marginally healthy trees in the orchard. The proposed framework is a cost-effective solution employing an off-the-shelf visible light camera. It operates on a physical drone equipped with a Raspberry Pi 4 computer running in the background, providing real-time tree classification data. The experimental outcomes demonstrate an impressive average validation accuracy of 92.67%.**

Index Terms—**Artificial intelligence, machine learning, precision farming, drones, tree health classification.**

I. Introduction

Farmers and plantations often manage vast expanses of land, a challenge exacerbated by the increasing global population. As these agricultural operations continue to expand, the tasks of maintenance and crop monitoring become increasingly daunting. Farms must address various crop maintenance activities, including weeding, pest and disease control, pruning, and field sanitation.

The initial and critical step in any maintenance regimen is crop monitoring. Field workers must vigilantly observe and identify potential issues that could hinder crop growth. Integrating an efficient crop monitoring approach with data analytics holds the potential to furnish farmers with comprehensive information for making informed decisions, thereby reducing the need for extensive manual labor.

Data analysis and machine learning offer the prospect of enabling farmers to make well-informed choices regarding resource allocation and yield projections through predictive models. The challenge at hand is the automation of the crop monitoring process to swiftly identify potential issues in fields or orchards, ultimately mitigating the associated costs and labor resources.

Related Work: The majority of literature concerning precision agriculture systems involves computer-implemented methods for generating graphical interface alerts based on diverse field data. These systems are primarily designed to tackle challenges such as detecting farm equipment malfunctions and providing suggested courses of action [1]. Machine learning played a central role in the formulation of these recommendations, and Alveraz *et al.* [2], have outlined the utilization of neural networks for creating artificial intelligence data structures that encompass parameters like seed selection and planting recommendations derived from crop field data. Yaniv Maor in the realm of precision agriculture has proposed innovative systems that extend beyond maintenance activities, including the use of drones for tasks like pruning crop trees and harvesting fruits [3]. The use of drones in data acquisition is not a novel concept [4], and there are existing patents that have elaborated on employing LiDAR point clouds, hyperspectral, and multispectral imaging for conducting thorough visual analyses to assess plant health and estimate future yields [5], [6]. Nonetheless, these systems entail significant costs that may be prohibitive for a farmer.

Among the works examined, machine learning methods have been highlighted as pivotal for generating recommendations. However, the specific models to employ, along with their associated training accuracy statistics, have not been clearly defined. The precision agriculture systems under review aim to address challenges like automating crop maintenance tasks such as fruit harvesting and identifying the causes of crop damage [1]–[3].

All of the proposed systems are better suited for larger fields, where the expenses associated with their implementation could be justified due to the substantial land area. Unfortunately, there were no cost analyses available for these specific systems in agricultural applications, but it is evident that the hardware requirements for these systems are not practical for smaller-scale farms.

In the context of smaller-scale precision agriculture implementations, opting for a monitoring solution over a maintenance-focused one might appear to be a more financially viable choice. We've observed that our relatively cost-effective crop monitoring solution has found comparable applications beyond the agricultural sector, particularly in the use of machine learning for tree classification and localization. In various instances, remote sensing techniques, including LiDAR point cloud data and hyperspectral data from spectrometers, have been harnessed to identify early signs of tree diseases and classify stages of conditions like pine wilt disease in forestry [7].

For our monitoring solution in targeted row crop fields, LiDAR isn't essential to distinguish individual crop plants due to their arrangement. Cost-effective RGB optical imagery reliably categorizes tree species for our specific application.

Similar projects have successfully employed solely RGB

979-8-3503-7099-7/24/$31.00 ©2024 IEEE

images for detecting invasive tree species, classifying various tree species, and even in applications focused on detecting fir and palm trees [8]–[11]. The use of aerial imagery for tree classification is not a novel concept and has previously been applied in forest and urban environments. However, we have not encountered an agricultural application that leverages deep learning and RGB optical imagery exclusively.

One distinctive aspect that sets our approach apart from prior work involving RGB optical images and deep learning is the utilization of classifications to generate informed recommendations. Previous efforts mainly focused on identifying tree species. In our approach, we will compile classifications of crop tree health and the relative positions of trees with similar health statuses to create priority groups. Recommendations for areas requiring immediate attention will be derived from these priority groups.

Maray *et al.* [12], present a scheme to classify coconut tree diseases in a smart farming environment via artificial intelligence. The Bayesian fuzzy clustering-based segmentation method is employed for the detection of the affected leaf regions besides CapsNet as a feature extractor. For the detection of disease, they utilized Harris Hawks optimization with grated recurrent unit models.

Sandric *et al.* [13], develop a framework based on consumer-based UAVs and deep learning techniques for orchard tree segmentation and health assessment. Tree health assessment is based on the use of vegetation indices. Two vegetation indices, including visible atmospherically resistant index (VARI) and green leaf index (GLI) were used with the standard score (which is also known as z-score) for tree health assessment.

In contrast to previous health assessment techniques that rely on costly multi-spectral imaging cameras and are built upon vegetation indices, our approach is grounded in the use of an affordable off-the-shelf RGB camera to classify tree conditions as "healthy," "slightly unhealthy," and "dying."

Our study encompasses two key objectives, each representing a significant milestone. Firstly, we aim to automate the flight path of an unmanned aerial vehicle (UAV) to enable precise image capture of a crop field or orchard. Subsequently, the captured images will undergo manual processing to trim and categorize them based on crop health aspects.

The second milestone involves training machine learning models using categorized images to create a predictive model for large volumes of raw image data. Multiple models are trained using the same dataset, prioritizing classification accuracy. Their performance is compared based on predefined metrics for analysis.

We have successfully designed, developed, and conducted real-world experiments involving the system across multiple orchards. Our efforts have yielded an impressive validation accuracy of 92.67% through the utilization of a deep learning algorithm in our multi-sensor data analysis. This algorithm proficiently classifies individual tree health into three distinct categories and subsequently shares this valuable information with the orchard owner.

The rest of the paper is organized as follows. In Section II, we describe the system overview, followed by the UAV prototype development in Section III, and the drone navigation in III. After illustrating experimentation results in Section IV, we draw the main conclusions in Section VI.

II. System Overview

A. System Design

Our system is purpose-built to furnish orchard owners with regularly updated health assessments for all the trees in their fields, enabling them to optimize resource allocation and stay informed about evolving crop conditions. It employs a machine learning framework in conjunction with an RGB camera mounted on an autonomous UAV, facilitating reasonably accurate health evaluations for various tree species.

The utilization of an RGB camera for this purpose presents notable advantages by obviating the necessity for invasive and stationary sensors to be scattered throughout the field. This approach allows for the monitoring of a larger number of trees at a faster pace and with reduced expenses.

The project is structured around two primary components: data acquisition and machine learning model training. For data acquisition, we have developed Python flight scripts that are integrated into a Raspberry Pi mounted on a drone. This Raspberry Pi communicates with the drone's flight controller using the MAVlink Protocol, enabling it to autonomously navigate the drone along a predefined flight path over an orchard to capture images.

Subsequently, the captured images are categorized into three distinct health classifications: "healthy," "slightly unhealthy," and "dying." These labels are then extracted and stored as individual JPEG files. These labeled images serve as the dataset for training, validating, and testing our convolutional neural network (CNN).

B. Deep Convolutional Neural Network Design

There are several types of CNNs, including Inception-V3, ResNet-50, and VGGNet. For our project, we've chosen to implement ResNet-50 as one of the models. This choice was based on a comprehensive performance comparison between VGGNet and ResNet-50.

When comparing these models, it's worth noting that VGG-16 operates at a rate of 15.3 billion Floating Point Operations Per Second (FLOPS), while ResNet-152 operates at approximately 11.3 billion FLOPS. ResNet-152 exhibits a lower FLOPS rate compared to VGG-16 despite having a greater network depth. This discrepancy can be attributed to the unique design of the ResNet architecture.

In traditional CNNs adding more layers often leads to improved computational efficiency, but only up to a certain point. Beyond that point, adding layers can result in accuracy degradation due to issues such as vanishing gradients and optimization challenges. ResNet addresses the vanishing gradient problem by introducing what are known as residual blocks. These blocks enable the neural network to create skip connections, effectively incorporating the concept of the identity function. This function ensures that higher layers do not perform worse than lower ones. Consequently, this approach aims to reduce errors and enhance the overall efficiency of the neural network.

C. Data Flow

The effectiveness of the proposed data flow primarily hinges on data acquisition. In the initial flight tests, a quadcopter drone is deployed, controlled by a Raspberry Pi 4 running flight scripts to navigate within a defined area over an orchard. To ensure proper data acquisition, unit testing is conducted to determine the drone's load-bearing capacity

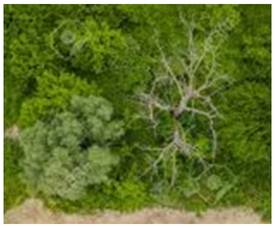


Fig. 1: Top down image showing an unhealthy tree (right) and a healthy tree (left).

and the optimal flight speed required for capturing coherent images. RGB images are captured using an onboard optical camera and stored on the Raspberry Pi's memory card. This process yields a diverse set of images, all of which are stored in a designated data repository.

These images serve as the foundation for training a CNN. Additionally, features necessary for our classification tasks are extracted from both the captured images and an open-source database containing a wider range of tree images with varying health conditions. Figure 1 provides a visual example of the desired image types, which undergo cropping and labeling to facilitate training of the CNN algorithm. The labeled images are fed into the neural network, enabling feature extraction for subsequent categorization during the model training phase. Following thorough training of the neural network using a sufficient volume of image data, the network is ready for implementation.

D. Data Collection and Processing

The data collection relied on a combination of private data obtained from Fresno State and our own collected dataset. This comprehensive dataset comprised 534 images, all captured during drone flights at an altitude of 30 meters. The images primarily depicted orange trees arranged in rows and had an original resolution of 4608×3456 pixels. Each image was taken from a top-down, bird's eye perspective. While we had 534 images in total, further annotation with bounding boxes was necessary to isolate individual orchard trees effectively.

The cropping and labeling of these images constituted a substantial portion of our data collection efforts. However, this step was crucial for training our model effectively. Following the acquisition of the entire dataset at a 30-meter altitude, we developed a Python script to crop and store images of individual trees. Subsequently, these images were organized into local folders, categorized as "healthy", "slightly unhealthy", or "dying"." The "healthy" category contained 2,393 images, the "slightly unhealthy" category comprised 1,814 images, and the "dying" category included 500 images. This resulted in a total of 4,707 individual tree images. The distribution among the three categories was approximately 5:4:1, representing healthy, slightly unhealthy, and dying trees, respectively. An example of the raw data is illustrated in Fig. 2 taken at 30 meters altitude.

These machine learning training results revealed that the uneven distribution of data had a significant adverse impact on our model. The substantial bias towards healthy trees potentially skewed the model's outcomes in favor of the healthy tree classification. To rectify this imbalanced data distribution, we employed data augmentation techniques to augment the dataset for the "slightly unhealthy" and "dying" categories, aligning them with the dataset for "healthy" trees.

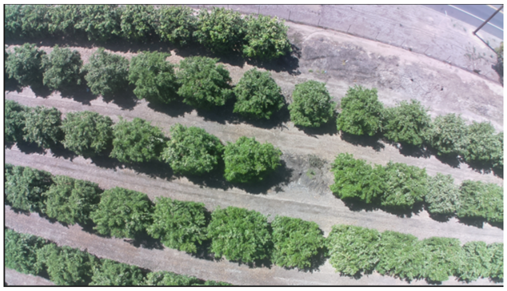


Fig. 2: Sample image taken by Arducam camera.

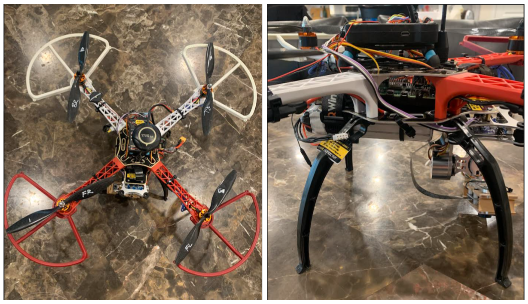


Fig. 3: Drone assembly.

III. UAV Prototype Development and Testing

We utilized a quadcopter drone kit equipped with various components, including an L3GD20 3-axis digital gyroscope, MPU6000 6-axis accelerometer, MS5611 precision barometer interface, an 11.1V 4400MAH 30C 3S1P battery, 4 pieces of 30A Brushless ESC Speed Controllers, and 2 pairs of DJI 920KV CW CCW Brushless Motors. Additionally, we incorporated a camera gimbal with an Arducam day and night vision camera.

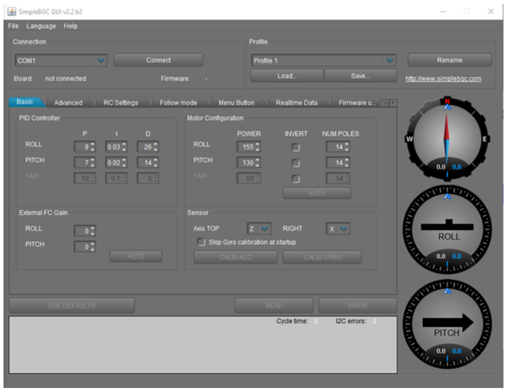
The assembly of the drone, complete with the gimbal camera mount, is depicted in Fig. 3. To ensure the proper functioning of all brushless and servo motors, thorough testing and verification were conducted using the AT9S Pro RadioLink Controller.

We conducted unit testing to confirm the functionality of all hardware components. Custom-fitted propeller guards were installed to mitigate the risk of catastrophic drone crashes. Furthermore, we made adjustments to the orientations of both the Raspberry Pi and Pixhawk flight controller to accommodate the inclusion of anti-vibration padding.

After implementing the modifications to the drone assembly, we affixed and recalibrated the gimbal mount. Since our project entailed capturing top-down images autonomously, there was no need for the pilot to manually adjust the camera angles using the AT9S radio controller.

The calibration of the gimbal mount was carried out using the BaseCam Electronics Camera Calibration System version 2.2 b2. For the roll axis motor and pitch axis motor, we determined the appropriate values for the proportional, integral, derivative, and power (PID and power) settings, as depicted in Fig. 4.

Initially, we set the PID values for the pitch motor to 5.00, 0.00, and 10.00, respectively. Subsequently, we increased



Connection: COM1

Board: not connected Firmware: -

Profile: Profile 1

PID Controller

	P	I	D
ROLL	9	0.03	26
PITCH	7	0.02	14
YAW	10	0.1	5

Motor Configuration

	POWER	INVERT	NUM.POLES
ROLL	155	□	14
PITCH	130	□	14
YAW	50	□	14

External FC Gain

ROLL: 0

PITCH: 0

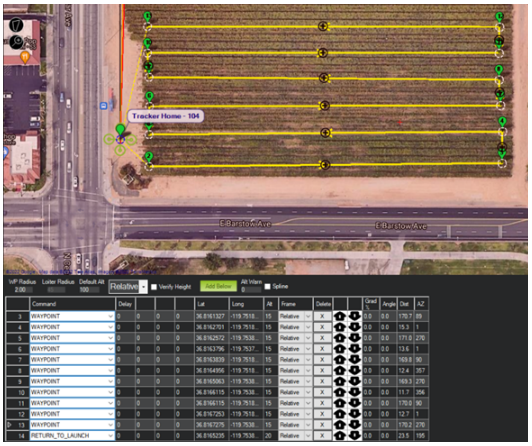
Sensor

Axis TOP: Z RIGHT: X

□ Skip Gyro calibration at startup

Cycle time: 0 I2C errors: 0

Fig. 4: Gimbal camera mount tuning settings.



WP Radius 2.00 Loiter Radius 45 Default Alt 100 Relative □ Verify Height Alt Warn 0 □ Spline

	Command	Delay				Lat	Long	Alt	Frame	Delete			Grad %	Angle	Dist	AZ
3	WAYPOINT	0	0	0	0	36.8161327	-119.7518...	15	Relative	X	⇧	⇩	0.0	0.0	170.7	89
4	WAYPOINT	0	0	0	0	36.8162701	-119.7518...	15	Relative	X	⇧	⇩	0.0	0.0	15.3	1
5	WAYPOINT	0	0	0	0	36.8162572	-119.7538...	15	Relative	X	⇧	⇩	0.0	0.0	171.0	270
6	WAYPOINT	0	0	0	0	36.8163796	-119.7537...	15	Relative	X	⇧	⇩	0.0	0.0	13.6	1
7	WAYPOINT	0	0	0	0	36.8163839	-119.7518...	15	Relative	X	⇧	⇩	0.0	0.0	169.8	90
8	WAYPOINT	0	0	0	0	36.8164956	-119.7518...	15	Relative	X	⇧	⇩	0.0	0.0	12.4	357
9	WAYPOINT	0	0	0	0	36.8165063	-119.7538...	15	Relative	X	⇧	⇩	0.0	0.0	169.3	270
10	WAYPOINT	0	0	0	0	36.8166115	-119.7538...	15	Relative	X	⇧	⇩	0.0	0.0	11.7	356
11	WAYPOINT	0	0	0	0	36.8166115	-119.7518...	15	Relative	X	⇧	⇩	0.0	0.0	170.0	90
12	WAYPOINT	0	0	0	0	36.8167253	-119.7518...	15	Relative	X	⇧	⇩	0.0	0.0	12.7	1
▷ 13	WAYPOINT	0	0	0	0	36.8167275	-119.7538...	15	Relative	X	⇧	⇩	0.0	0.0	170.2	270
14	RETURN_TO_LAUNCH	0	0	0	0	36.8165235	-119.7538...	20	Relative	X	⇧	⇩	0.0	0.0	23.5	195

Fig. 5: SITL flight simulations over potential sites.

the values for its corresponding power incrementally by 15 until the motor demonstrated resistance to external stimuli. Next, we raised the proportional values incrementally until the accelerometer and gyroscope sensors for that axis exhibited smoother responses after the external stimulus. Simultaneously, we adjusted the derivative value to dampen oscillations. The integral value, typically quite small, was increased by 0.01 and controlled the speed at which the gimbal returned to its preset position.

With the hardware properly configured, calibrated, and fine-tuned, we acquired various coordinates essential for planning drone flight missions. These coordinates played a pivotal role in guiding the drone's flight path.

Initially, we conducted flight simulations using Mission Planner and Software in the Loop (SITL), as illustrated in Fig. 5. These simulations served as a foundational step in ensuring the safety of autonomous flights. The drone successfully executed basic Python scripts for actions such as arming, takeoff, hovering, landing, and a simple return to home for coordinated flight path control.

One of the challenges we faced was the requirement for network connectivity at field sites. While it is indeed feasible to initiate flight scripts upon Raspberry Pi startup, establishing a Secure Shell protocol (SSH) network connection to the Raspberry Pi was essential for monitoring purposes and to facilitate real-time customization of flight scripting as needed in the field.

We also explored the Raspberry Pi's potential for deep learning applications. While it can run TensorFlow Lite, we encountered limitations related to its computing power. Specifically, the Raspberry Pi lacks a dedicated GPU, making it insufficient for simultaneously training models while executing flight scripts. We had anticipated this challenge and established that all deep learning training and computations must be carried out on an external computer.

IV. UAV Prototype Field Experimentation

Deploying the drone in the fields involved a well-defined sequence of steps. To execute this process, Python scripts were developed and loaded onto the onboard Raspberry Pi, enabling the drone to operate with just two sets of geographical coordinates. These coordinates corresponded to the starting point of the targeted orchard tree row and its endpoint, as shown in Fig. 5. Subsequent rows were determined by incrementing or decrementing values of 0.0002 for longitude and 0.00015 for latitude. It's important to note that these values could be adjusted based on flight altitude, speed, and desired camera coverage.

The drone's flight parameters were configured to maintain a relative ground speed of 2 meters per second while flying at an altitude of 30 meters over the orchards. These specific values were chosen during testing to achieve an estimated 80% overlap in the captured photos. This deliberate overlap was designed with future project phases in mind, which involve digitally stitching all the images together and performing object detection on the resulting orthophoto.

Furthermore, additional configurations included the implementation of fail-safes and monitoring mechanisms. These included terminal notifications in the event of GPS disconnection or when the battery level reached critical levels. Additionally, real-time monitoring of altitude and speed was enabled. Emergency landing procedures were programmed to execute automatically if GPS connectivity was lost or if the battery reached a critical level. In accordance with FAA regulations and laws, a manual controller was continuously connected to the drone, providing a means for manual override of flight controls if deemed necessary.

The camera settings were adjusted to capture a total of 40 images for each row, both over the orchards at Fresno State and those in Sanger belonging to an acquaintance. In total, we acquired 320 images that were neither cropped nor labeled, alongside other data collected from Fresno State. Figure 6 illustrates the drone experimentation conducted in the orange tree orchard at Fresno State.

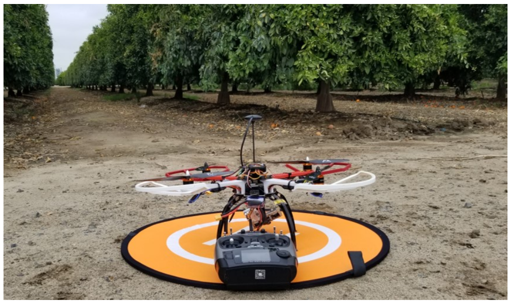


Fig. 6: Experimentation in the orange tree orchard at Fresno State.

Localization data is gathered using the drone's GPS and timestamps linked to image capture. To deploy, a network connection is set up between the drone's Pi and a laptop. This connection allows the operator to SSH into the Raspberry Pi, execute flight scripts, monitor terminal outputs, and smoothly transfer images from the Pi to the laptop using SCP.

V. Performance Results for RGB Camera

The dataset of aerial images of orchard trees was labeled using LabelImg, which is a widely used labeling software in the machine learning community.

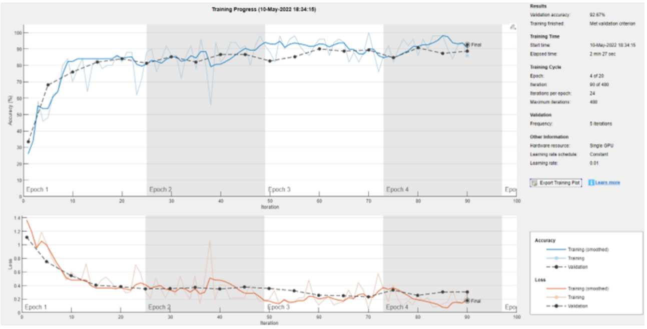
To assess the data quality, we developed an image categorization model to classify our labeled images. Initially, we extracted cropped images for each labeled object in a given image using a dedicated Python script. Subsequently, these extracted objects were organized into different folders according to their respective classifications. This process resulted in over 1,000 labeled trees for training purposes.

We used transfer learning on a ResNet-50 model in Matlab for classification training with a reduced dataset. This step ensured the accuracy of manual labeling before moving to object detection training in TensorFlow. Assessing classification independently is crucial as it influences object detection, validating the labeled images before developing the detection model. The training progress results are illustrated in Fig. 7. The model's parameters included the use of stochastic gradient descent with momentum for optimization. The specific settings were as follows: a momentum value of 0.9, an initial learning rate of 0.01, a maximum of 20 epochs, and a validation patience of 4 epochs. This classification training, with these parameters, achieved an average validation accuracy of 92.67%.

The confusion matrix depicted in Figure 8 reveals that the category where the most misclassifications occurred was "slightly unhealthy" trees. This pattern was consistent across multiple training sessions. The figure illustrates that trees categorized as "slightly unhealthy" are frequently misclassified as "healthy."

VI. Conclusion

In this work, we have created an affordable framework for autonomously categorizing the health of orchard trees, specifically designed for small-scale farmers. Our system seamlessly integrates a drone, a visible camera, a mini-computer, and state-of-the-art machine learning algorithms to accurately categorize and precisely determine the health status of trees in an orchard. Localization data is obtained by using the drone's GPS and timestamping image captures. We conducted extensive training and field experiments at Fresno State and Sanger farms, and the results demonstrate the system's ability to detect and classify tree health with an impressive average validation accuracy of 92.67%.



Results
Validation accuracy: 92.67%
Training finished: Met validation criterion

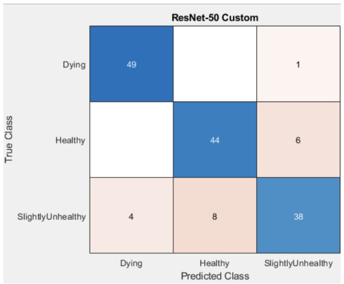
Training Time
Start time: 10-May-2022 18:34:15
Elapsed time: 2 min 27 sec

Training Cycle
Epoch: 4 of 20
Iteration: 90 of 480
Iterations per epoch: 24
Maximum iterations: 480

Validation
Frequency: 5 iterations

Other Information
Hardware resource: Single GPU
Learning rate schedule: Constant
Learning rate: 0.01

Fig. 7: Validation accuracy and loss plot for the RGB camera.



	Dying	Healthy	SlightlyUnhealthy
Dying	49		1
Healthy		44	6
SlightlyUnhealthy	4	8	38

Fig. 8: Confusion matrix classification results.

Acknowledgment

The work was partially supported by the Economic Development Administration (EDA) Project Grant 077907908, Fresno-Merced Future of Food Innovation (F3) Coalition. Special thanks to DPS Telecom, and to the Lyles College of Engineering at California State University, Fresno.

References

[1] P. M. BarsamianDaniel, D. P. Garrison, A. Mathur, J. C. Mendez, and P. Mullan, "Australia Patent AU2017228695B2," *Precision Agriculture System*, pp. 1–41, 2019.

[2] F. Alvarez, M. Ali, J. Melching, and E. Hochmuth, "United States Patent US20200097851A1," *Method and system for executing machine learning algorithms*, pp. 1–41, 2019.

[3] Y. Maor, "Canada Patent CA3034062A1," *System and method for plantation agriculture tasks management and data collection*, pp. 1–95, 2017.

[4] A. Rejeb, A. Abdollahi, K. Rejeb, and H. Treiblmaier, "Drones in agriculture: A review and bibliometric analysis," *Computers and Electronics in Agriculture*, vol. 198, p. 107017, Mar. 2022.

[5] A. Bleiweiss, "United States Patent US9898688B2," *Vision enhanced drones for precision farming*, pp. 1–17, 2016.

[6] K. T. McPeek, "United States Patent US10520482B2," *Systems and methods for monitoring agricultural products*, pp. 1–31, 2019.

[7] R. Yu, Y. Luo, Q. Zhou, X. Zhang, D. Wu, and L. Ren, "A machine learning algorithm to detect pine wilt disease using uav-based hyperspectral imagery and lidar data at the tree level," *International Journal of Applied Earth Observation and Geoinformation*, vol. 101, p. 102363, 2021.

[8] M. Onishi and T. Ise, "Explainable identification and mapping of trees using uav rgb image and deep learning," *Scientific reports*, vol. 11, no. 1, p. 903, 2021.

[9] J.-D. Sylvain, G. Drolet, and N. Brown, "Mapping dead forest cover using a deep convolutional neural network and digital aerial photography," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 156, pp. 14–26, 2019.

[10] K. Yarak, A. Witayangkurn, K. Kritiyutanont, C. Arunplod, and R. Shibasaki, "Oil palm tree detection and health classification on high-resolution imagery using deep learning," *Agriculture*, vol. 11, no. 2, p. 183, 2021.

[11] A. Safonova, S. Tabik, D. Alcaraz-Segura, A. Rubtsov, Y. Maglinets, and F. Herrera, "Detection of fir trees (abies sibirica) damaged by the bark beetle in unmanned aerial vehicle images with deep learning," *Remote sensing*, vol. 11, no. 6, p. 643, 2019.

[12] M. Maray, A. A. Albraikan, S. S. Alotaibi, R. Alabdan, M. A. Duhayyim, W. K. Al-Azzawi, and A. alkhayyat, "Artificial intelligence-enabled coconut tree disease detection and classification model for smart agriculture," *Computers and Electrical Engineering*, vol. 15, no. 104, p. 108399, 2022.

[13] I. Şandric, R. Irimia, G. P. Petropoulos, A. Anand, P. K. Srivastava, A. Pleşoianu, I. Faraslis, D. Stateras, and D. Kalivas, "Tree's detection & health's assessment from ultra-high resolution UAV imagery and deep learning," *Geocarto International*, vol. 37, no. 25, pp. 10 459–10 479, 2022.