Robotic Assessment of a Crop's Need for Watering

Automating a Time-Consuming Task to Support Sustainable Agriculture

By Amel Dechemi, Dimitrios Chatziparaschis, Joshua Chen, Merrick Campbell, Azin Shamshirgaran, Caio Mucchiani, Amit Roy-Chowdhury, Stefano Carpin, and Konstantinos Karydis

©SHUTTERSTOCK.COM/FUN FUN PHOTO

Agricultural robotics and automation technology plays an increasingly critical role across several crop production stages to improve sustainability (e.g., water use optimization). Robotics has been applied for remote and proximal sensing as well as physical sampling. The procedure of physical sampling and follow-on analysis of plant specimens, such as leaves or shoots, which is the focus of this work, often constitutes the only accurate way to measure some essential parameters that affect crop production, such as stem water potential (SWP), which helps determine a crop's need for watering.

However, both physical sampling and specimen analysis can be quite laborious and often vary among different types of crops. Given growing agricultural workforce shortages [1], the labor-intensive nature of these measurements poses a severe limitation on how many samples can be collected and analyzed and restricts how growers and farm consultants can assess local conditions and optimize operations in support of sustainable crop production. This is particularly critical in high-value perennial crops, such as avocados, citrus, almonds, nuts, and vines. Hence, robotics and automation technology can be employed to support physical sampling and specimen analysis.

The goal of this work is to develop robotics and automation technology tools that can help assess watering needs for tree crops. The primary focus lies in automating the process of measuring SWP. SWP is a metric frequently used by agronomists and growers to optimize irrigation schedules for crops [5], with the intent of reducing water waste and maximizing profits [15], [19]. The current industry standard for obtaining SWP measurements is the Scholander pressure chamber method [16], which involves the insertion of a leaf sample into a pressure chamber, with the stem excised end exposed [Figure 1(a)]. After the sample is secured, a human operator slowly activates

Digital Object Identifier 10.1109/MRA.2023.3321391
Date of current version: 26 October 2023

a valve to pressurize the chamber while also observing the water expression at the end of the exposed stem through a magnifying glass [Figure 1(b)]. The pressure necessary to force water out of the stem determines the SWP. The process measures the capacity of the cells to retain water by pressurizing the leaf. The less free water there is in the plant, the greater the pressure required to cause the leaf to exude water. When taken in predawn conditions (i.e., performed before sunrise) and when plant stomata are closed, the measurement is at equilibrium with soil moisture conditions. Subsequent measurements can precisely determine the water deficit, and, thus, the irrigation demand, to meet evapotranspiration loss. A precise measurement of the SWP is essential to assess the water deficit of the plant and, therefore, to adjust irrigation. Given that 80% of the managed fresh water in the United States is consumed by agriculture [21] and that evapotranspiration model estimates widely diverge [8], even modest improvements in irrigation practices can have huge impacts, especially in the semiarid southwestern United States, which periodically undergoes drought periods while providing a large fraction of the country's fruits and vegetables.

> "PHYSICAL SAMPLING AND SPECIMEN ANALYSIS CAN BE QUITE LABORIOUS AND OFTEN VARY AMONG DIFFERENT TYPES OF CROPS."

Considering the aforementioned labor-intensive steps in measuring SWP using the pressure chamber method, but recognizing its significance in crop production, some alternative methods have been proposed. Some rely on remote sensing using spectral reflectance or multispectral imaging to determine water potential [20], [25]. Their greatest values are their noninvasive characteristic, much like the pressure chamber method, and scalability since these methods are intended as a faster alternative to the pressure chamber for mass assessment. Although initially promising, these methods are highly sensitive to an intractable external factor, i.e., light variability due to weather and solar motion. Zhao et al. [25] mounted multispectral cameras on a small unmanned aerial vehicle to take high-resolution multispectral images of orchards for SWP prediction using the canopy Normalized Difference Vegetation Index but mentioned the high variability in data collected from different flights within the same day, due to solar motion. Vila et al. [20] used remote sensors and spectral reflectance as a proxy for SWP measurements but had a low correlation coefficient and, thus, concluded that this method cannot serve as a replacement for the pressure chamber method. Since the concepts of spectral reflectance and imaging may not be mature yet for this application, our work seeks to directly automate the pressure chamber method for assessing SWP.

Specifically, we are developing a robotic system that can physically collect and analyze plant specimens in the field

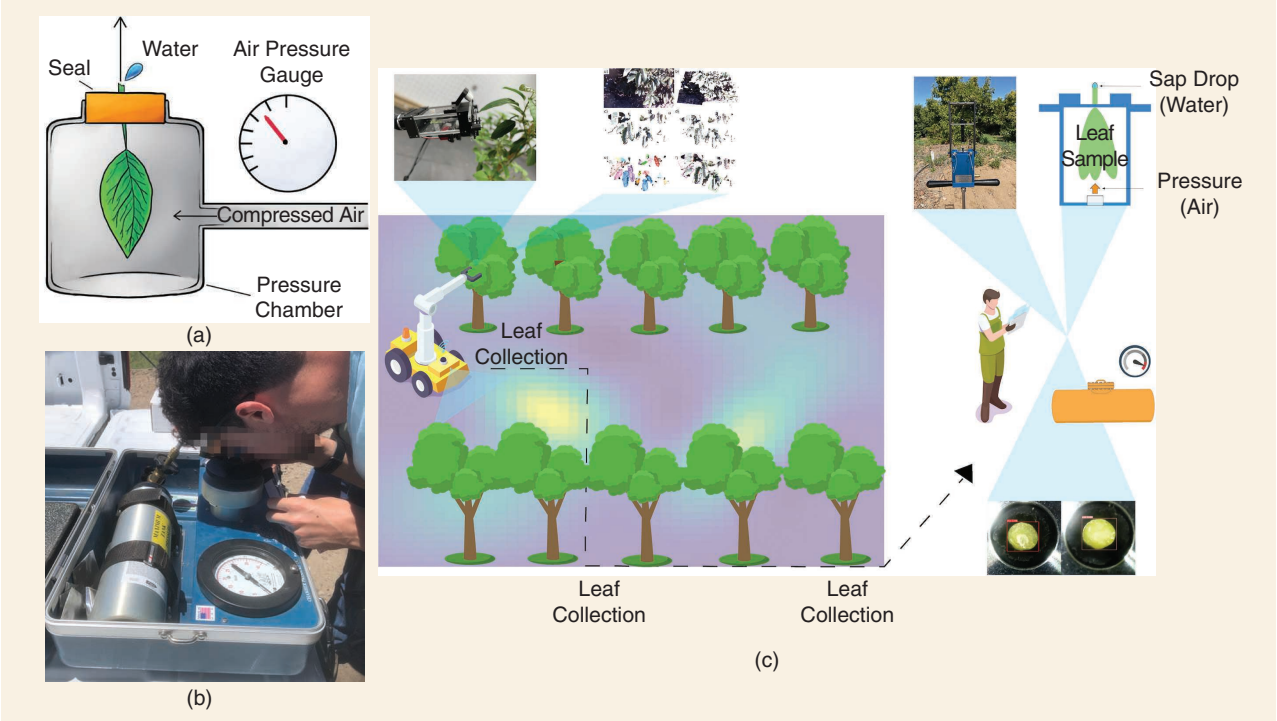

FIGURE 1. (a) The working principle of the pressure chamber. (b) Manual visual inspection is currently performed in the field for SWP analysis. (c) The concept proposed in this work. A mobile robot autonomously selects multiple measurement locations for sample collection and retrieves leaves. These are conveyed to a human operator, who uses a pressure chamber retrofitted with our machine vision-assisted technology to determine the pressure defining the SWP.

at scale to determine the SWP, thus removing some of the main practical obstacles to the implementation of precision agriculture and providing growers and farm consultants with granular data currently not available. The overall system developed in this work aims to automate time-consuming and difficult-to-perform tasks, namely, collecting multiple leaves from tree crops and automating some parts of the follow-on SWP analysis. With reference to Figure 1(c), which outlines the overall idea, our system first determines the location of trees from which to sample leaves in the field (the "Informed Planning for Selection of Sampling Tree Locations" section). This part contributes to informed path planning under motion energy constraints for autonomous mobile robots. Once sentinel trees have been selected, a mobile manipulator with a custom-designed end effector is tasked to autonomously retrieve leaf samples and transport them back to an SWP analysis station (the "Robot Design and Deployment for Autonomous Leaf Retrieval" section). Our work here contributes key information regarding component selection, system design, and system integration features underlying autonomous perception–actuation loops for mobile manipulation. When the robot has reached the SWP analysis station, a human operator loads the samples into a machine vision-assisted pressure chamber to determine the exposed leaf stem's wetness (the "Machine Vision-Assisted Identification of Leaf Stem Wetness in SWP Analysis" section). This final part contributes data and methodologies to employ machine vision and machine learning to help automate the very labor-intensive process of determining the point of xylem water expression during SWP. In the remainder of the article, we detail the aforementioned process and contributions. Finally, we summarize lessons learned and sketch ideas for future research enabled by the work described herein (the "Discussion and Outlook" section).

> "A PRECISE MEASUREMENT OF THE SWP IS ESSENTIAL TO ASSESS THE WATER DEFICIT OF THE PLANT AND, THEREFORE, TO ADJUST IRRIGATION."

INFORMED PLANNING FOR SELECTION OF SAMPLING TREE LOCATIONS

The overall process initiates by determining a sequence of trees to collect leaf samples from. A direct brute-force approach would be to sequentially sample from all trees in the field. However, such a process would not scale with commercial field sizes. To make this process tractable, growers currently rely on sampling from a small number of so-called sentinel trees. When measurements are taken manually, a small fixed set of sentinel trees is preselected and sampled throughout the growing season. A drawback of this approach is that samples from a small number of trees are used to extrapolate the values of relevant parameters over significantly larger areas, a process that may yield inaccurate results and hence lead to suboptimal watering decisions.

Deployment of one [10] or more [9] autonomous mobile robots in the field to collect samples opens up the possibility of considering a larger set of sentinel trees and also to plan a sequence of sampling locations on the fly, ideally based on the outcome of former measurements. This process is subject to an energy constraint; i.e., the motion planning algorithm that determines the next tree for sample collection has to ensure that the robot will not run out of energy before returning to a preassigned location, where its battery can be either recharged or swapped for a new one. With this approach, a much larger set of potential sentinel trees can be considered, and the planner task is to select at runtime the most suitable subset for sampling. While in previous precision agriculture works we studied this problem as an instance of the orienteering problem [11], we here cast it as an instance of the informed path planning problem, where the goal is to solve the following optimization problem:

$$p^* = \operatorname{argmax}_{p \in \Pi} f(p) \qquad \text{s.t. } C(p) \leq B\,.$$

Here, Π is the set of all paths through the set of sentinel trees, $f(p)$ is the quality of the path, and $C(p)$ is the cost of the path that is bounded by the available energy budget B. Ideally, the quality of a path $f(p)$ would be assessed by the root-mean-square (RMS) error of the scalar field being estimated. Because in practice the ground truth is unknown, the RMS cannot be directly computed, and we aim, rather, at collecting samples that reduce this metric. To this end, we use Gaussian processes (GPs) to model the underlying scalar parameter being estimated, i.e., SWP [17]. GPs are an appealing modeling approach because of their efficient posterior update, and they are also widely used in geostatistics to model physical phenomena (where the approach is usually referred to as *kriging*.) Specifically, we use a Matérn kernel with smoothness $\nu = 1.5$, as this has proved to be an appropriate choice in agricultural applications.

Using GPs, after each sample is collected, we can update the uncertainty associated with the scalar field being estimated and then iteratively build a path collecting more samples at sentinel trees located in regions with high uncertainty. Therefore, after each sample is collected, the importance of each sentinel tree is reassessed based on the values collected up to that point. Note that in our approach, samples are collected only at the selected sentinel trees and not along the way. This differs from our previous work [11], where each candidate sampling location's value was predetermined and fixed. Different metrics can be used to estimate the value of collecting a sample at a given location. In some instances, the variance of the GP estimate is used [2], with the goal of collecting samples in high-variance regions to reduce uncertainty. In our work, we are currently focusing on mutual information (MI). Let $\mathcal{S}$ be the set of sentinel trees and

$\mathcal{A} \subset \mathcal{S}$ a subset of already-sampled sentinel trees. The MI between these two sets is defined as

$$MI(\mathcal{A}; \mathcal{V} \backslash \mathcal{A}) = H(\mathcal{V} \backslash \mathcal{A}) - H(\mathcal{V} \backslash \mathcal{A} | \mathcal{A})$$

where H is the entropy associated with the given sets, based on the current estimate. Owing to the fact that GPs are used for the underlying estimates, closed-form expressions can be obtained for MI [17]. Then, the path p can be iteratively extended by adding the sentinel tree

$$t = \underset{t \in \mathcal{V} \backslash \mathcal{A}}{\operatorname{argmax}} \; [H(\mathcal{V}_t \backslash \mathcal{A}_t) - H(\mathcal{V}_t \backslash \mathcal{A}_t | \mathcal{A}_t)]$$

where $\mathcal{V}_t = \mathcal{V} \backslash \{t\}$ and $\mathcal{A}_t = \mathcal{A} \cup \{t\}$ subject to the constraint that after having reached the tree being added to the path, there must be enough energy left to reach the final destination, i.e., the location where the pressure chamber is located. Once a new sampling point is selected, a path to it is computed using the rapidly exploring random tree motion planning algorithm informed by the fixed locations of the trees in the orchard. This process is iterated until the endpoint is selected and reached.

To test the viability of this approach, we performed extensive tests in a simulated environment, where, under controlled conditions, we varied the underlying scalar field being estimated as well as the number of sentinel trees. Figure 2 displays a subset of our obtained results and shows that the proposed approach is capable of reconstructing the underlying field, even with a small budget, by visiting the most significant (i.e., informative) locations. Figure 2(c) also displays the path produced by the planner. [The path in Figure 2(e) is omitted for figure clarity.] Additionally, deployment of the developed strategy for determining sampling locations during actual field experiments is presented in the following section.

ROBOT DESIGN AND DEPLOYMENT FOR AUTONOMOUS LEAF RETRIEVAL

Once a desired sequence of sentinel trees has been selected, the deployed robot (a wheeled mobile manipulator; see Figure 3) needs to navigate toward the corresponding field locations to physically sample leaves. We can perform this process autonomously by integrating four core functionalities: 1) end-effector design to cleanly cut and retain leaves, 2) onboard visual perception to detect and localize candidate leaves, 3) robot arm manipulation to enclose a candidate leaf with the end effector and cut it, and 4) waypoint navigation of the wheeled mobile robot based in the field to travel from one sampling location to another and back to the SWP analysis station. Our previous work [3] has focused on studying the third functionality, namely, co-optimizing perception and actuation for autonomous leaf retrieval in static cases. Here, we present research findings on the important components of the development and assessment of the end effector, camera selection, and assessment to enable visual leaf identification and pose estimation as well as overall system integration and field deployment and testing. We anticipate that the information included herein can also help other researchers working on related topics of robotic visually informed physical sampling.

END-EFFECTOR DESIGN

To be appropriate for SWP analysis, the end effector needs to cleanly cut the leaf stem to separate the test specimen from the host tree as well as retain the cut leaf for the subsequent pressure chamber analysis since a damaged specimen can negatively impact the analysis.

REQUIREMENTS AND BENCHMARK TESTING FOR EFFICIENT CUTTING

Given the radius of a leaf stem (r), we can calculate the required cutting force as $F = \pi r^2 \tau$, with shear stress $\tau \in [0.85, 5.90]$ MPa

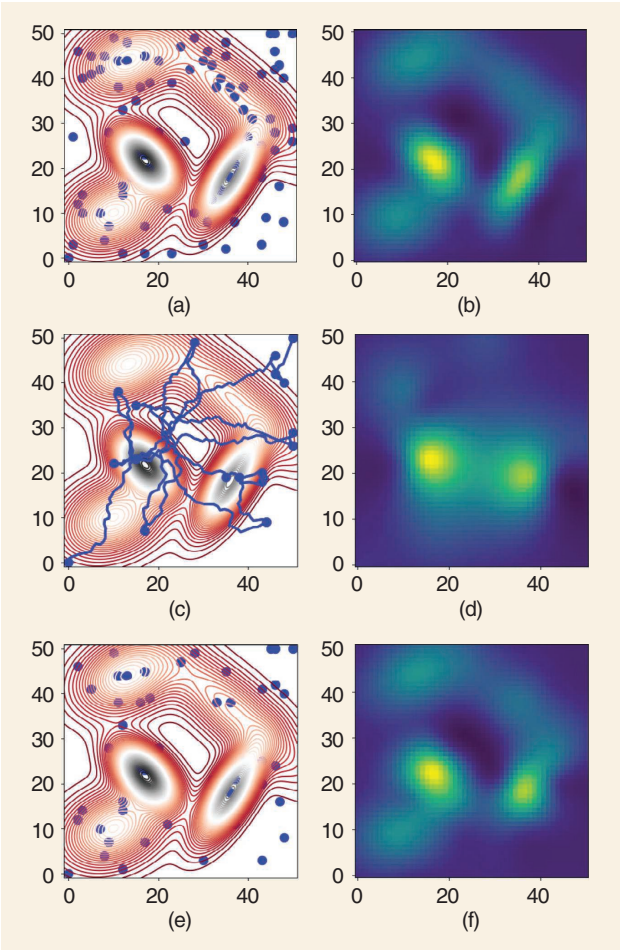

FIGURE 2. (a) The underlying scalar field (ground truth) and the location of 100 candidate sentinel tree locations. (b) The GP reconstruction obtained if the robot is allowed to collect samples at all locations (used for benchmarking purposes). The (c) sampling points and path selected by the algorithm, with a budget C of 750 (small value), and (d) reconstructed scalar field obtained using these sampling locations. The (e) sampling points (path not shown for image clarity) selected by the algorithm, with a budget C of 2,000 (average value), and (f) reconstructed scalar field obtained using the latter locations. In the simulation, distances and budgets are in meters.

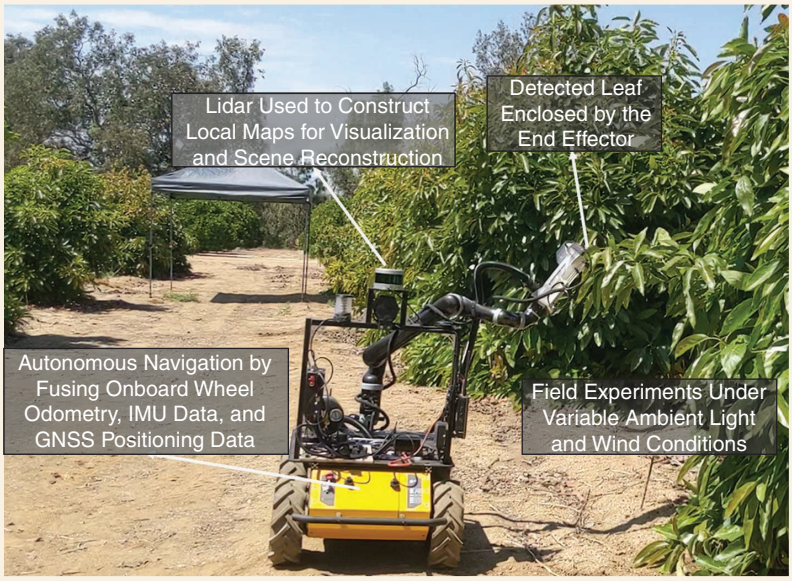

FIGURE 3. The agricultural robot used in this work for robotic assessment of a crop's need for watering. IMU: inertial measurement unit; GNSS: global navigation satellite system.

for cutting organic plant matter [24]. We measured the diameter of 10 leaves from four different tree crops (avocado, clementine, grapefruit, and lemon), for a total of 40 leaves. The average leaf stem diameter was 2.09 mm, with a standard deviation of 0.51 mm. With this information, we were able to estimate that the force required to cut the average leaf ranged from 2.9 to 20 N. However, organic matter, such as leaf stems, exhibits viscoelastic properties. When stress is applied, the material resists deformation linearly with time. When stress is removed from an elastic material, the material returns to the original nondeformed state. Based on viscoelastic material principles, faster cuts require less force and result in less deformation of the leaf stem. Hence, the rate of cut is equally important to the delivered force. For this reason, we conducted a set of cutting experiments to determine the optimal cutting rate.

Specifically, an initial prototype leaf cutter was placed on a level platform above a measuring stick, with a high-speed camera positioned to face the cutting blades. Three distinct gear sets were 3D-printed so that they could be inserted between the servo motor and the cutting mechanism to adjust the speed. For each gear ratio (7:13, 22:13, and 41:13), four leaves were inserted into the mechanism, and the high-speed camera recorded the cutting attempt for each leaf. The selected motor had sufficient torque margins so that the desired cutting force could be delivered with all tested gearing setups. Recorded frames were analyzed to determine the terminal speed of the cutting mechanism. With a known camera frame rate [240 frames per second (fps)] and travel distance (19.1 mm), the cutting speed of the mechanism can be determined as

$$V = \frac{\delta x}{\delta t} \approx \frac{\Delta x}{\text{frame rate} \times \text{frame count}}.$$

Of the three gear ratios, only the fastest gearing resulted in a cleanly cut leaf. Table 1 lists results from all the trials. From this analysis, we determined that the minimum cutting speed for the cutter should be 0.312 m/s. With both the cutting force and rate determined, we proceeded to develop an end effector with a desired target force of 20 N at 1.1 m/s. This rate provides a sufficient margin over the empirically determined minimum cutting speed of 0.312 m/s to account for any losses and work with a wide variety of tree leaves.

TABLE 1. The leaf cutting velocity tests.

GEAR RATIO	FRAME COUNT	TIME (S)	SPEED (M/S)	SUCCESS
7:13	48	0.2	0.095	No
7:13	40	0.167	0.114	No
7:13	39	0.163	0.117	No
7:13	41	0.171	0.112	No
22:13	20	0.083	0.229	No
22:13	25	0.104	0.183	No
22:13	17	0.071	0.269	No
22:13	18	0.075	0.254	No
41:13	16	0.067	0.286	Yes
41:13	20	0.083	0.229	Yes
41:13	11	0.046	0.416	Yes
41:13	14	0.058	0.327	Yes

PROTOTYPE END-EFFECTOR ASSEMBLY

The cutting mechanism utilizes two four-bar linkages to actuate a set of sliding gates, one of which contains a razor blade to cleanly sever the stem without damaging the leaf (Figure 4). The gates also help retain the leaf within the end effector's chamber after removal from the tree. These four-bar mechanisms are connected via a gear train to achieve synchronized motion. A low-cost high-torque radio control servo (FeeTech FT5335M) drives the gear train while being amenable to adequate position control. The end effector's chamber has an opening of 110 × 45 mm and a depth of 185 mm to accommodate typical avocado leaves (which are the largest of the four tree crops considered in this work). The end effector is constructed with miniature aluminum extrusions, lightweight 3D-printed parts, and laser-cut acrylic panels. The assembly weighs a total of 1.09 kg, which is 42% of the robotic arm's 2.6-kg payload onto

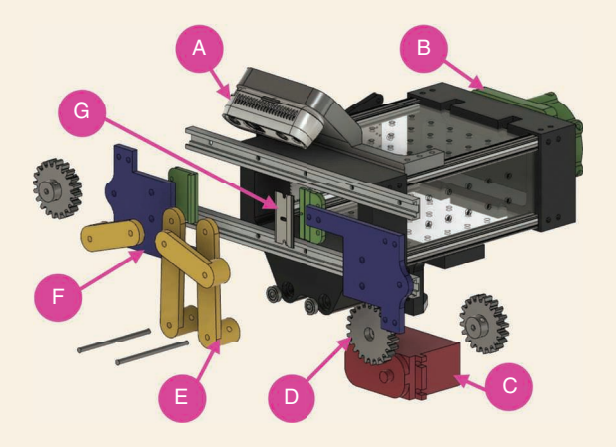

FIGURE 4. An exploded view of the end effector. Key components of the assembly include a RealSense D435i depth camera (A) and an interchangeable robotic arm mount (B). The FeeTech FT5335M radio control servo (C) is connected via a gear train (D) to four-bar linkages (E). This mechanism closes the gates (F) to cut the leaf with the razor blade (G). This separates the leaf from the tree and retains it within the enclosure for subsequent SWP analysis.

which we integrated our end effector (Kinova Jaco, six degrees of freedom).

The end effector was designed to operate symbiotically with the Robot Operating System (ROS). High-level control commands can be handled via an ROS node. The node receives commands from published ROS topics and issues commands to the end effector via serial universal asynchronous receiver/transmitter communication. The end effector contains an embedded microcontroller (Arduino Due) to parse the received serial commands and control the motor that drives the cutting mechanism. A breakout board connected to the Arduino contains a "safe/armed" switch along with LED indicators to reduce the risk of accidental injury from the razor blade (for redundancy, the high-level ROS control node also has a software "safe/armed" switch). A 7.4-V 2S lithium polymer battery powers the assembly. The overall prototype was mounted on top of a commercial mobile manipulator (a Clearpath Husky wheeled robot with a Kinova Jaco robot arm) also running ROS.

ONBOARD PERCEPTION

Our work considers an "eye-on-hand" configuration whereby a camera mounted on top of the end effector (Figure 4) is used to provide information regarding leaves to sample as well as their position and orientation (pose) in 3D space.

TABLE 2. The candidate camera specifications.

CAMERA	BASELINE (MM)	DEPTH RANGE (M)	FIELD OF VIEW
ZED	120	0.3–25	90 × 60 × 100°
ZED 2	120	0.3–20	110 × 70 × 120°
ZED Mini	63	0.1–15	90 × 60 × 100°
RS D435i	50	0.2–3	87 × 58 × 95°
RS D455	95	0.4–6	87 × 58 × 95°

RS: RealSense.

CAMERA SELECTION AND PLACEMENT EVALUATION

Several cameras were considered as the sensing modality for the proposed end effector (Table 2). Although the ZED and ZED 2 have solid performance, they were excluded because of their wide baselines, which do not fit our intended eye-on-hand configuration. We evaluated the performance of three other cameras in different conditions, including indoor and outdoor environments. The obtained results show that the RealSense (RS) D435i has the best performance, especially outdoors, where it is able to provide a viable depth image at close ranges. Furthermore, we were able to obtain high-quality point clouds at depth ranges lower than those provided in manufacturer specifications (0.1 m). Sample images collected using the RS D435i are provided in Figure 5.

Two eye-on-hand configurations were considered, one looking straight ahead and one looking downward at a 45°

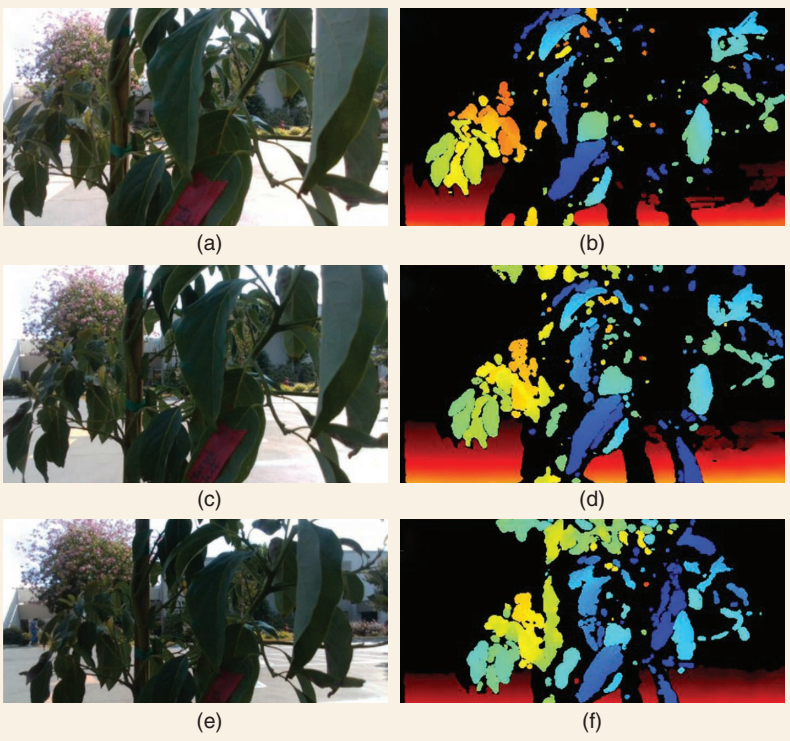

FIGURE 5. Sample red–green–blue (RGB) and depth images collected from the RS D435i in an outdoor environment at (a)–(b) 15 cm, (c)–(d) 20 cm, and (e)–(f) 25 cm.

FIGURE 6. Key steps in our proposed leaf detection and localization process. The sample here corresponds to an outdoor point cloud: the (a) corresponding RGB image of the tree, (b) raw point cloud, (c) background removal, (d) downsampled point cloud, (e) segmented clusters, and (f) detected candidate leaves without 6D pose bounding boxes. (Image taken from [3].)

angle. While the former case can lead to longer look-ahead distances, the latter one was ultimately selected. This configuration balances providing useful depth information about the tree (needed for obstacle avoidance and navigation around tree branches) and allowing for leaf detection and localization (needed for aligning the end effector with the leaf to cut it).

SCENE UNDERSTANDING

The onboard camera yields a 3D point cloud, which we are consequently using to detect and localize appropriate leaf candidates via 3D bounding boxes. The process is visually exemplified in Figure 6 and described next. First, we removed outliers caused by noise in sensor measurements and segmented out the background. The degrees of outlier detection and background removal depend on user-tuned hyperparameters. Then, we applied downsampling for computational expediency without a loss of performance. Next, we grouped the remaining point cloud segments into clusters using the Density Based Spatial Clustering of Applications With Noise (DBSCAN) approach. DBSCAN relies on two key parameters: the minimum distance between two points to be neighbors (*eps*) and the number of minimum points to form a cluster (*MinPoints*). Each resulting cluster is considered a potential leaf and described by a 3D bounding box defined by center $C = [c_x, c_y, c_z]^T$, dimensions $D = [h, w, d]$, and orientation $R(\theta, \Phi, \alpha)$. Then, we filtered the clusters by using geometric features of the bounding box: the number of points, volume, and leaf ratio. Finally, the pose of the center of each bounding box was set as the 6D pose of a leaf candidate.

To validate our approach, ROSbags were collected both in indoor (a lab with constant light conditions) and outdoor (a local orchard with varying light conditions) settings. In indoor experiments, the robot arm with the camera was placed at different distances ($0.2 - 0.3$ m) from a potted tree, whereas outdoor data were manually collected over a wide range ($0.5 - 1.6$ m) of distances from trees. A total of 25 point clouds were collected (10 indoor and 15 outdoor) and tested offline with different combinations of *eps* and *MinPoints* parameters to determine appropriate values.

The results demonstrated 80% detection on average (maximum: 90%) in the indoor dataset and 79.8% detection on average (maximum: 85%) outdoors (see Table 3). Further, we observed that the distance between the camera and the tree affects parameter selection for point cloud processing. As the distance increases, values for *eps* and *MinPoints* should increase and decrease, respectively.

TABLE 3. The leaf point cloud detection.

	POINT CLOUDS	TOTAL LEAVES	AVERAGE DETECTION	PERCENTAGE
Indoor	10	20	16	80%
Outdoor	15	99	79	79.8%

EVALUATION OF AUTONOMOUS LEAF EXTRACTION EFFICACY IN CONTROLLED SETTINGS

The system operation flow diagram regarding how to detect and extract a leaf from a tree is depicted in Figure 7. First, the robot base needs to be in close proximity to a tree to sample leaves from. In detail, the mobile base needs to be at an offset distance from the tree canopy so that the intersection of the reachable workspace of the robot arm and the outer canopy is maximized. Note that considering the leaf and end-effector 3D orientations is crucial; hence, the robot should also avoid being too close to the canopy, as this can lead to singular configurations for the arm. This tradeoff is captured in our work via the notion of viable leaves. These are leaves that are detected within the reachable workspace of the robot and

away (in a topological sense) from singular configurations that could bring the robot arm to a locked status.

Once in place, obtained point cloud data from the camera mounted on the end effector are processed in real time using Open3D [26] running on an Intel i7-10710U CPU without any additional GPU acceleration. Identified and segmented leaves serve as targets for the arm to move to and align the end effector along a detected leaf candidate within the end effector's workspace, at an offset position from the center of the leaf. The offset distance can be selected in relation to the length of the leaf. When the end effector reaches the target offset, the arm is commanded to move along a linear path toward the leaf. When the leaf is enclosed, the end effector cuts the leaf and retains it. Then, the arm returns home, and the process may repeat as part of a larger field sampling procedure; see the next section.

We evaluated our developed autonomous leaf extraction pipeline in controlled settings indoors with a real potted avocado tree. The end effector was initialized at random poses near the base of the tree at distances ranging between 0.2 and 0.3 m from the edge of the tree canopy. An experimental trial consisted of collecting a point cloud, storing the identified and localized potential leaves in a queue, and then sending the queued leaves to the arm for a retrieval attempt. Each trial concluded once the queue was depleted, and the tree was repositioned for the next trial. Figure 8 outlines this process. Out of 46 trials, 63 potential leaves were detected by the point cloud. (Note that each point cloud in the trial could produce a variable number of leaves, hence the higher number of potential

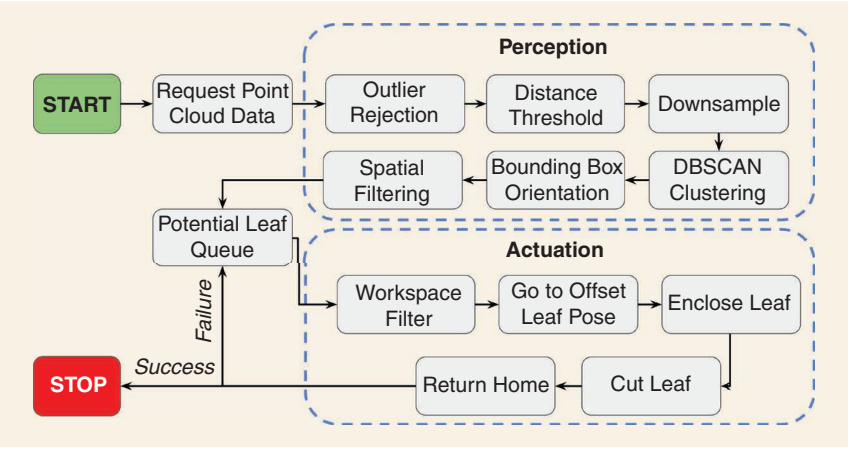

FIGURE 7. Our approach for autonomous leaf extraction integrates perception and actuation. Point cloud data are processed at runtime to segment leaves and add leaf candidates to a queue. Candidate leaves are then passed to the robot arm and end effector. If a cut is successful, the routine ends, but if not, the next leaf in the queue is requested, and the process repeats.

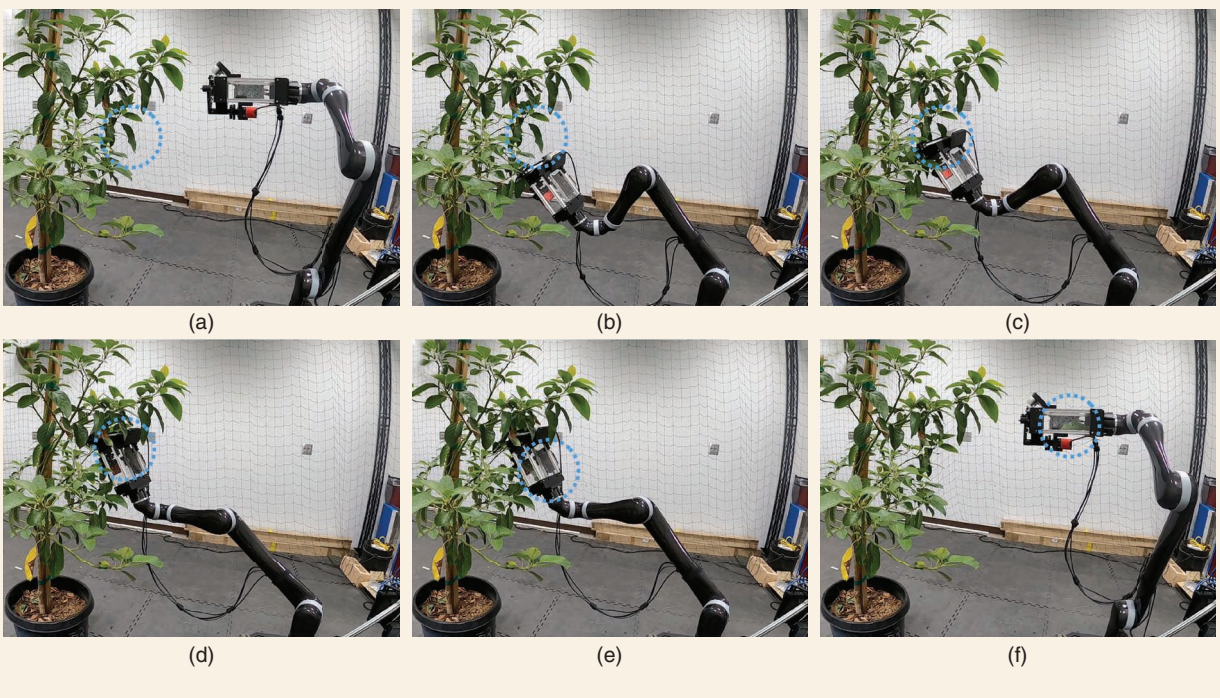

FIGURE 8. The sample leaf retrieval process. (a) Point cloud data are processed to determine a potential leaf. (b) Given a detected leaf within the workspace, the arm moves to an offset position. (c) The arm moves in a linear motion toward the leaf to enclose it. (d) and (e) When in place, the end effector cuts the leaf and places it in the enclosed chamber. (f) After completing the cut, the arm returns to the home position. (Image taken from [3].)

leaves than trials.) After filtering the potential leaves to remove the leaves outside of the work space, 39 viable leaves remained. Out of these leaves, 27 were captured successfully (69.2%), while 21 of the 27 captured leaves were cut (77.8%). The mean point cloud processing (perception) time was 5.6 s, and the mean cutting (actuation) time was 10.6 s. The mean total retrieval time was 16.2 s. These findings demonstrate that the developed system can successfully cut and retain leaves in controlled settings.

OVERALL SYSTEM INTEGRATION AND FIELD EXPERIMENTS

The final step toward enabling autonomous leaf retrieval consists of the integration of waypoint navigation and deployment in real-world field experiments. To assess the robustness of our framework, consider a list of sentinel tree locations. This information is used to create desired waypoints that serve as the locations the robot should visit to attempt to sample leaves from corresponding sentinel trees. Then, we integrate the leaf extraction steps outlined in Figure 7 for every encountered tree. When the leaf extraction process terminates for each sentinel tree, the robot moves to the next sampling area, and the process repeats, or the robot returns to the analysis station, where collected leaves are retrieved by a human coworker and placed in the pressure chamber for automated SWP analysis (see the "Machine Vision-Assisted Identification of Leaf Stem Wetness in SWP Analysis" section).

Field experiments were conducted in an avocado tree field at the Agricultural Experimental Station (AES) (33° 58′ 3.2592″ N, 117° 20′ 7.0296″ W) at the University of California, Riverside, Riverside, CA, USA. We used satellite imagery to construct an outline of the geometry of the field, including tree positions. As an example demonstrating how to reduce the general planner to practice, we consider the case of sampling from three sentinel trees within an area of interest measuring 11×6 m^2. The selected sentinel trees and underlying GP reconstruction are presented in Figure 9. The computed path for the robot to follow per our planning algorithm in the "Informed Planning for Selection of Sampling Tree Locations" section is also highlighted in the figure.

The satellite-based map is described in World Geodetic System 1984, but to be usable from the robot, it needs to be linked with the mobile robot's local map, which is, in turn, used for robot navigation. We use the Universal Transverse Mercator (UTM) projection to express the satellite-based sampling points into desired waypoints in the robot's local coordinate system. The mobile robot base we use (Clearpath Husky) can obtain odometry information from its wheel encoders; orientation, linear velocity, and angular acceleration measurements from its embedded inertial measurement unit module; and positioning data from its onboard global navigation satellite system receiver. This information is used onboard and in real time by the *navsat_transform_node* from the built-in ROS navigation stack to broadcast the pose (i.e., position and orientation) of the mobile robot base in the UTM local coordinate system. The robot's pose is updated in the local frame while moving, by using fused information from the three onboard sensory modules. The movement actions in the local frame are handled by the *move_base* ROS package, which generates velocity commands for the mobile platform in order to acquire the desired pose in the local frame. For additional safety, our developed system allows a human operator/supervisor to trigger when the robot switches between navigation and leaf picking modes as well as to skip a sampling location and move to the next one. The robot arm is rigidly affixed to the mobile base; coordinate transforms between the mobile base and each arm link frame as well the end-effector frame are all readily computed via closed-form forward kinematics expressions.

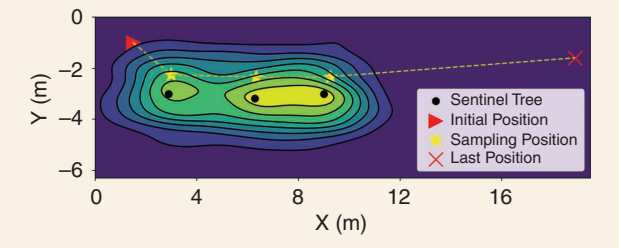

FIGURE 9. The GP reconstruction of the avocado field, with the three sentinel trees considered during field experiments.

In our experiments, the mobile robot starts from a known position on the map. The robot arm is initialized turned toward the right-hand side of the mobile base so that the camera mounted on the end effector has an unobstructed view of the tree canopies from the right-hand side of the robot as the latter moves forward. This configuration helps distribute the load to the mobile base as evenly as possible given the other embedded parts, and it minimizes occlusions to an embedded lidar sensor that is currently used to collect data during operation to create the visualizations in Figure 10. The first desired pose is transmitted to the platform, which, in turn, moves toward the target tree at the desired position using the generated trajectory. As the first goal pose is reached, the mobile platform stops, and the leaf retrieval process subsequently initiates. The perception module processes the collected point cloud and returns the center and dimensions of each candidate leaf with respect to the camera frame. The tip and stem positions are then estimated with respect to the end effector's frame, and the manipulation planning procedure outlined in Figure 7 is executed. Figure 10(a) illustrates an instance of the mobile robot when sampling at the third desired location. Three candidates' leaves have been successfully detected, and the process can then proceed to the actuation procedure. Figure 10(b) depicts the path followed by the robot in the field experiment until reaching the third sentinel tree. The complete field experiment can be viewed in a supplemental video at https://youtu.be/xu4zrTe_S-U.

MACHINE VISION-ASSISTED IDENTIFICATION OF LEAF STEM WETNESS IN SWP ANALYSIS

The final stage of this work concerns the automated identification of leaf stem wetness during SWP analysis. We consider

that leaves have been collected by the mobile manipulator and transported back to the SWP analysis station. There, a human operator loads/unloads a leaf into the pressure chamber. The latter is retrofitted with additional hardware as well as machine vision software to enable automated visual identification of the status of the leaf xylem (dry/wet). Our approach provides a live feed of the stem as the pressure in the chamber increases. When the status changes from dry to wet, our developed system identifies the transition and alerts the operator so that they can note down the pressure. To accommodate the large variability that is typically present in this process, our proposed solution hinges on learning-based machine vision. In the following, we detail the data acquisition, hardware and software development, and algorithm training and evaluation components that are necessary to automate this labor-intensive process of determining the leaf xylem wetness status.

A key part of our work is the creation of a new stem image dataset, collected over the course of two sessions (two weeks in July 2022 and another week in June 2023) at AES fields in an arid hot climate. We collected upward of 80 avocado and orange leaf samples (for system training and validation) and 20 lemon and grapefruit leaf samples (for preliminary evaluation of the developed method's generalization capacity), performed conventional SWP analysis while video recording each session, and extracted frames from the recordings to build the dataset.

DATA ACQUISITION

SETUP AND PROCEDURES

We considered two pressure chamber setups, the Pump-Up Chamber and 600D from PMS Instruments. The former is powered manually by performing alternating upward and downward strokes that increase the pressure inside the chamber (similar to a hand pump). The latter uses compressed nitrogen gas, which enters the chamber by rotating a knob connected to a valve. (For more details about these chambers, please see the manufacturer's manuals at https://www.pmsinstrument.com/.)

We also considered two different miniature cameras to be mounted onto the chambers. These included a high-quality miniature camera (Pi Camera) together with a Raspberry Pi 3 B+ single-board computer to store the video footage and a wireless endoscope camera that was streaming the video footage wirelessly to a smartphone. The rationale for these two selections is that the first camera setup allows for a fully customizable means that can support further robotics and automation integration. The second camera (a low-cost commercially available endoscope) can serve as an off-the-shelf solution to make the work herein self-contained as an intuitive and low-cost way to stream and assess video footage by the user in a stand-alone device fashion. Both cameras (one at a time) were mounted to the chambers via a custom-made 3D-printed carbon fiber reinforced mount, which attaches directly on top of the chambers' point of loading the stem of the cut leaf (Figure 11). The assembly can be removed between measurements and facilitates focal and aperture adjustments to produce clear images of stem samples.

We followed the proper protocol when performing SWP analysis and data acquisition. Sampling was performed midday (12–3 p.m.) per recommendations from relevant literature [5], and leaf samples were bagged in reflective foil bags for at least 10 min before excision to mitigate water loss through transpiration. During the first session, the excised leaves were placed in an opaque icebox storage container (not touching the ice) and transported back to the lab for SWP analysis. While earlier literature recommended making the measurements on the spot and avoiding transportation of samples, the work of [14] suggested immediate storage of excised leaf samples in a cold and moist environment to help stabilize the samples' water potential and preserve the condition for hours or even days, depending on the plant species [14]. We followed the standard SWP analysis procedures and proper protocol for sample collection and storage to ensure that our image dataset is representative of real-world underlying conditions. However,

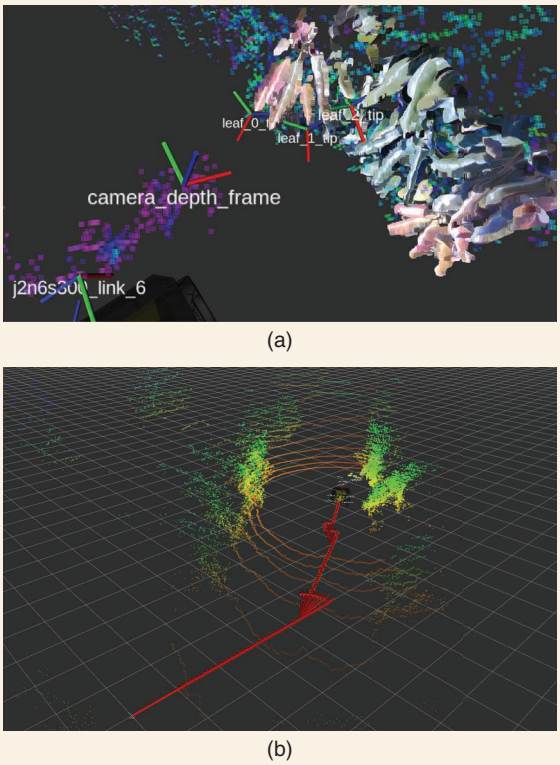

FIGURE 10. (a) A visualization (in ROS RViz) of the mobile robot at the third sentinel tree location. Each depicted coordinate system represents the corresponding state at the captured moment. Three leaf candidates, namely, *leaf_*{0, 1, 2}*_tip*, have been detected. Given these candidates, the actuation module decides to reach the closest one and attempts to cut it for retention. (b) A visualization of the path followed in the experimental avocado field. The captured moment shows the robot in the third leaf sampling position during the leaf detection procedure. Red arrows illustrate the odometry poses along the robot's path from the starting position to the sampling position.

in an effort to gather a dataset with more variability to enable more thorough evaluation of our method, in the second session, we processed the excised leaves on the spot in the field.

DATASET CREATION

The developed image dataset used for network training, testing, and validation consists of 10,713 images of dry and wet excised leaf stem cross sections from avocado and orange trees, with a binary class partition of 50.6% dry and 49.4% wet. These images are video frames extracted from our recordings of SWP analysis through the conventional pressure chamber method using all four different setups (recall that we used two different pressure chambers and two different miniature cameras). Considering that we also use leaves from two distinct tree species, the overall dataset contains images from eight distinctive real-world cases. The majority of the images at this stage, however, are taken from setups involving the static 600D pressure chamber model.

As with any image dataset, the frames were first preprocessed. The video recordings from the Raspberry Pi were first imported to a desktop and converted from the h264 to the MP4 format. The video resolution was redefined from 1,920 × 1,080 to 640 × 640 (the footage was resized to maintain the true aspect) to match the expected input image size for the You Only Look Once (YOLO) networks [13] we employed in our learning-based machine vision approach. Videos from the endoscope camera were already of a uniform aspect ratio, so a quick resize from 720 × 720 to 640 × 640 was applied to all videos from this setup. Once the video resolutions were redefined, we extracted video frames of interest as images and saved them in the working directory. Finally, all images in the working directory were manually annotated with an open source YOLO format annotation tool. The resulting labels and frames were then compiled into our stem image dataset, with a partition of 7,759 training, 1,952 validation, and 1,002 testing images.

A similar procedure was performed for the second session, which contained images sampled from 20 video instances from leaves excised from lemon and grapefruit trees (10 from each species). In this setting, we used one setup, i.e., the pump-up chamber with the high-quality Pi Camera, and conducted measurements in the field under variable ambient light conditions. For this dataset part, we extracted and evaluated a total of 45,000 images and manually classified them based on xylem wetness status (dry/wet). This dataset part was not used in training, testing, and validation at all. Instead, it was used as a way to test the preliminary feasibility of our method to generalize to very diverse out-of-sample cases. In the following section, we employ the first dataset part, whereas the "Preliminary Evaluation of Generalization Capacity" section considers the second dataset part.

OBJECT DETECTION OF DRY-OVER-WET STEM CONDITION

YOLO is a single-stage detector capable of performing both object identification and classification in a single shot, allowing the network to be smaller and faster [13]. Since its original release, the YOLO family of models has continued to evolve and improve in terms of structure and performance. In this work, we focused on three YOLO iterations: YOLO v5, YOLO v6, and YOLO v7. We wish to highlight here that other learning-based machine vision methods are, in principle, applicable. However, we elected to work with YOLO in this work due to its widespread use and solid outcomes in a number of distinctive applications. As we show shortly, this method can help reach very high classification accuracy, which was deemed sufficient in this work.

Each network structure consists of three main parts: backbone, neck, and head. The YOLO v5 backbone is based on a cross-stage partial network (CSP) [23] to reduce the amount of calculation and increase the speed of inference and a spatial pyramid pooling structure [7] to help improve detection accuracy. The adaptive anchor frame calculation on the input image enables automatically setting the initial anchor frame size and adapting to dataset changes. The neck includes feature

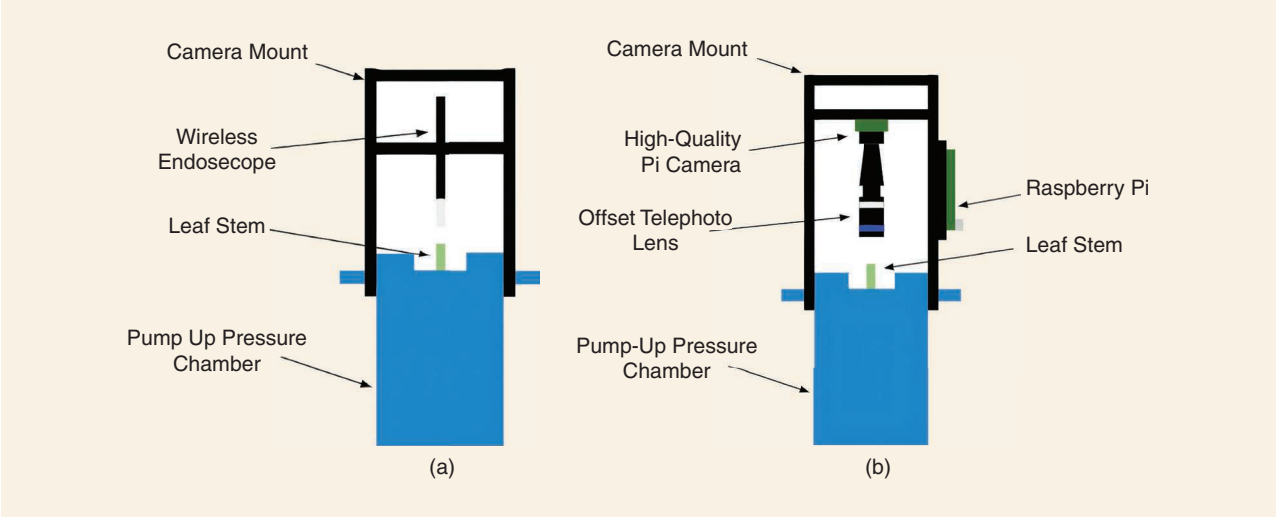

FIGURE 11. The custom (a) endoscope and (b) high-quality Pi Camera mount and setup on the manual pump-up chamber. The mounting on the 600D nitrogen gas pressure chamber follows a very similar manner and is not shown here for brevity.

pyramid structures of PANet [12] that aim to strengthen the features extracted from the backbone and further improve the detection capability. The head predicts targets of different sizes based on the obtained feature maps.

The following version, YOLO v6, introduced the EfficientRep Backbone and Rep-PAN Neck by replacing the CSP-Block used previously with RepBlock [4]. The new structure has a decoupled head, which adds layers separating the features from the final head and has been found able to increase performance. It uses an anchor-free paradigm and SimOTA, a simplified version of optimal transport assignment [6], to enhance the training speed and detection accuracy.

In the YOLO v7 version, several changes and enhancements were made. Notably, extended efficient layer aggregation networks (E-ELANs) [22] serve as the backbone to improve the learning ability. Compared to the original ELAN architecture, the E-ELAN changes only the architecture in the computational block, as the architecture of the transition layer is unchanged. The authors of [22] proposed a new compound scaling method to employ parameters and computation more efficiently and planned a reparameterized model that can optimize the scaling process and be applied to concatenation-based models. The head consists of an auxiliary, a lead head, and a soft label assigner for coarse and fine labels.

The efficacy of the trained models was evaluated in terms of three criteria: 1) localize accurately the stem in an image, 2) perform classification of the two states dry and wet, and 3) have a stable transition from dry to wet in a video setting.

NETWORK TRAINING PARAMETERS

Since the three YOLO networks employ distinctive structures, we trained each version from scratch on 7,759 training images in two ways: baseline and tuned. The baseline variation features the baseline hyperparameters provided by the developers of the methods in their respective repositories. In the tuned variation, we performed tuning of selected hyperparameters to assess the impact of different data augmentations on each model's performance and training time. The tuned hyperparameters include a decrease in saturation augmentations ($\downarrow 0.3$ probability), an introduction of rotation augmentations ($\pm 45^{\circ}$), and a deletion of the mix-up and paste-in augmentations. The baseline hyperparameters feature more emphasis on color space augmentations and less on spatial-level transformations, whereas our tuned hyperparameters place more emphasis on spatial-level transformations and less on color space augmentations. We justify our tuning choices by reiterating the findings of [18], where color is not an invariant feature in wetness detection. By training a model on more spatial-level and fewer color augmentations, we force the model to be more sensitive to color space discriminant features and less sensitive to shapes since every sample is unique. All models were trained on a Tesla P100 GPU for 80 epochs, with an initial learning rate of 0.01. For optimization, a stochastic gradient descent (SGD) optimizer was applied, with a momentum of 0.937 and a weight decay of 0.0005.

EVALUATION OF NETWORKS' INFERENCE CAPABILITIES

We evaluated each trained network on the basis of test classification accuracy, mean average precision (mAP), and inference speed. Inference was performed on both cloud and edge computing hardware with test images and on edge computing hardware with a test video. Table 4 presents the training results of all the YOLO models, with their baseline and tuned hyperparameter variants. Both the baseline and tuned YOLO v5 models achieved the highest mAP, at 0.5:0.95, within their corresponding variants. This means that both YOLO v5 models predicted accurately and with high confidence (≈ 0.95) in their detection. Out of all the models trained, the tuned YOLO v7 scored the highest mAP, 0.5, indicating its ability to detect stems close to their ground truth bounding boxes.

Regarding the models' cloud-based inference, we can observe from Table 5 that our tuning of hyperparameters and augmentations increased the classification performances of YOLO v5 and YOLO v7. Additionally, the inference speed increases marginally for both models. Conversely, YOLO v6 degraded in both classification performance and inference speed under our tuning parameters. Our results for cloud-based inference using a Tesla P100 GPU yielded inference speeds within approximately 30 ms/frame ($\geq$30 fps) and thus prove that real-time water detection with our approach is possible given equivalent or more powerful computing hardware.

With respect to edge-based inference, and with reference to Table 6, both YOLO v5 models are able to perform predictions within every half second on the Nvidia Jetson Xavier NX, and both models also attained a 100% classification accuracy metric on the test images. We observe that all the other models

TABLE 4. The three distinctive YOLO networks and variations considered in this work and the training results on our developed stem image dataset with base and tuned hyperparameters and augmentations.

NETWORK	SIZE (MB)	TRAINING TIME (H)	EPOCHS	RECALL	PRECISION	MAP AT 0.5	MAP AT 0.5:0.95
YOLO v5 base	13.8	2.1	80	0.999	1	0.995	0.929
YOLO v6 base	36.3	6.1	80	0.94	0.995	0.995	0.91
YOLO v7 base	71.4	10	80	0.999	1	0.998	0.91
YOLO v5 tuned	13.8	2.4	80	0.999	0.995	0.995	0.885
YOLO v6 tuned	36.3	6.6	80	0.91	0.995	0.995	0.881
YOLO v7 tuned	71.4	10	80	0.999	0.994	0.998	0.884

TABLE 5. The cloud-based inference results with a Tesla P100 on the image test set.

NETWORK	EXECUTION TIME (S)	AVERAGE INFERENCE SPEED PER IMAGE (MS)	CLASSIFICATION ACCURACY
YOLO v5 base	27.2	10.4	99.3%
YOLO v6 base	25	25.5	99.5%
YOLO v7 base	38.1	21.5	99.7%
YOLO v5 tuned	26.5	8.8	100%
YOLO v6 tuned	30	30.1	99.4%
YOLO v7 tuned	37.6	21.4	100%

TABLE 6. The edge-based inference results with a Jetson Xavier NX on the image test set.

NETWORK	EXECUTION TIME (MIN)	AVERAGE INFERENCE SPEED PER IMAGE (MS)	CLASSIFICATION ACCURACY
YOLO v5 base	8.04	481.6	100%
YOLO v6 base	19.57	1,171.7	99.5%
YOLO v7 base	38.69	2,316.5	99.7%
YOLO v5 tuned	7.49	448.4	100%
YOLO v6 tuned	17.35	1,038.9	99.4%
YOLO v7 tuned	46.24	2,769	100%

TABLE 7. The edge-based inference results with a Jetson Xavier NX on the test video.

NETWORK	AVERAGE INFERENCE SPEED PER FRAME (MS)	STABLE STATE TRANSITION
YOLO v5 base	471.4	No
YOLO v6 base	1,020	Yes
YOLO v7 base	2,863.3	Yes
YOLO v5 tuned	568.1	Yes
YOLO v6 tuned	1,030.9	Yes
YOLO v7 tuned	2,811.6	Yes

attained high classification metrics as well, with the tuned YOLO v7, in particular, scoring a 100% success rate on the test images. Yet, given that it is five times larger than YOLO v5, its inference speed was about five times slower than that of the YOLO v5 model, making it less practical for deployment on less powerful edge devices. YOLO v6 experienced a slight inference speed performance boost with our tuning parameters.

Regarding image inference speed, Tables 5 and 6 indicate a direct relation between model size and inference speed. Smaller models tend to perform faster and are thus suited for deployment on devices with limited computational power. Table 7 contains the average edge device inference speed results for each model on a test video taken from the dataset preprocessing phase earlier in this work. Similar to the image inference results, smaller models have faster video inference speeds. As expected, the average inference speed per frame for each model resembles the corresponding metric in edge-based image inference (Table 6).

QUALITATIVE ANALYSIS

Figure 12 juxtaposes the wetness detection results on a single orange sample test case by all YOLO models trained on our dataset. The detection confidence is approximately 0.95, and it remains consistent with the bulk of our test results. Comparing the baseline and tuned versions of each model, we can observe that the bounding boxes are slightly tighter around the stem detection for the baseline networks compared to

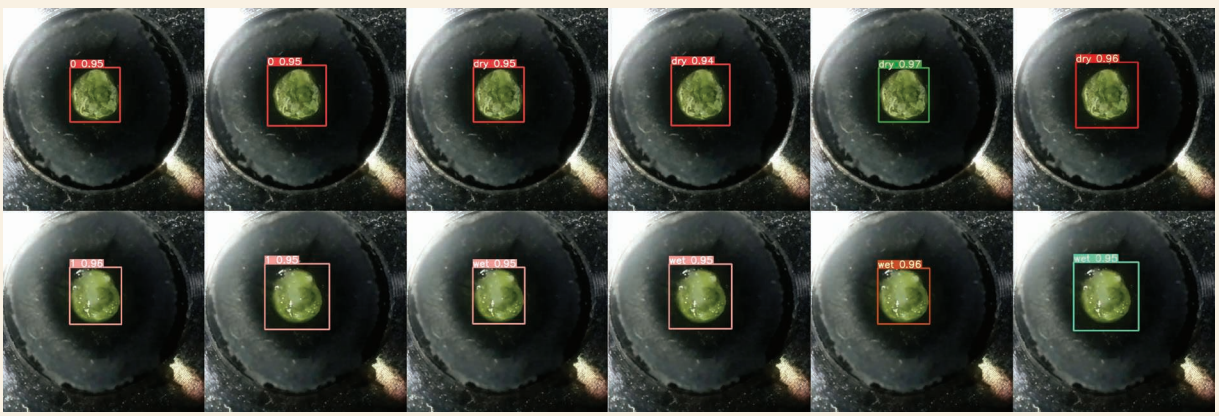

FIGURE 12. Correct wetness detection results on a single orange stem by all the YOLO models: (a) YOLO v5 base, (b) YOLO v5 tuned, (c) YOLO v6 base, (d) YOLO v6 tuned, (e) YOLO v7 base, and (f) YOLO v7 tuned. The top panels correspond to the dry status, while the bottom panels correspond to the wet status.

their tuned counterparts. One possible explanation is that the combination of the rotation augmentation and training samples with oversized bounding boxes may have led to removing less of the background. As a result, the tuned models have a higher propensity for fitting slightly looser bounding boxes.

When inference is executed at a low confidence threshold (i.e., 0.1), some of the models may occasionally raise a false detection in some of the test cases. Figure 13 gives two illustrative examples of falsely detected instances of dry and wet classes by the baseline YOLO v7. The falsely detected dry case is of only a 0.13 confidence score and can hence be filtered out by specifying a higher confidence threshold of 0.6, as implemented during the evaluation phase. This detection occurred most likely because of some of the training instances featuring a dryer and more heavily brown-colored pith. The reflected light from the chamber lid interface does remotely resemble some of the features present in the aforementioned training instance, but its effects are negligible since the model detected with a low confidence rather than a high confidence (this would have been concerning). In contrast, the false wet detection occurred most likely because of the model recognizing the smooth side of the petiole as wet. The wet confidence score is 0.12. (Its view in the figure is obstructed by the dry bounding box since we increased the line width and font size for inference to improve visibility.) This case emphasizes the importance of aligning the stem sample properly with the camera, as the models may be sensitive to misalignment.

In Table 7, the rightmost column refers to the quality of classification during the video's transitional frames, where the state of the stem sample changes from dry to wet. The transitional frames are not considered wet, even with the onset of moisture, because the proper SWP analysis protocol defines the wet point as full water expression, whether that be a full clear water drop or a water drop with vigorous bubbling. This helps enforce consistency with SWP measurements. We observed that the tuned models can reach the final wet state more smoothly through the transition phase compared to the baseline models. Most notably in the tuned variants of YOLO v6 and YOLO v7, there is minimal classification oscillation during the transitional phase, as both models classify consistently dry until full water expression, when they both classify consistently wet after a brief lull in classification accuracy.

PRELIMINARY EVALUATION OF GENERALIZATION CAPACITY

To assess the generalization capabilities of the proposed models, we further tested the six trained networks on the never-seen-before second dataset part. Note that this dataset part is significantly more challenging because of ambient light variations and large oscillations observed while operating manually the pump-up chamber. Figure 14 highlights some of these challenging instances.

The extracted images were used to evaluate the precision and recall for each model. The obtained results are reported in Table 8 (the first six entries in the table). It can be observed that the baseline structures were less efficient, as suggested by their precision and recall scores, compared to their tuned counterparts. The only exception is the YOLO v5 models, which offered the same performance, with a precision of 0.39 and a recall of 0.34. The tuned YOLO v7 network had the highest performance, with a precision score of 0.75 and a recall score of 0.49. A recurring classification error across all the networks was the detection of a dry state as a wet state. This can be attributed to the ambient light variations and motion of the pump-up pressure chamber throughout the analysis, resulting in blurred or distorted images, as shown in

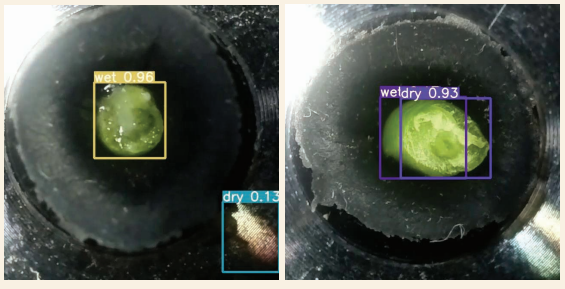

FIGURE 13. (a) A falsely detected dry instance by YOLO v7 base. (b) A falsely detected wet instance by YOLO v7 base.

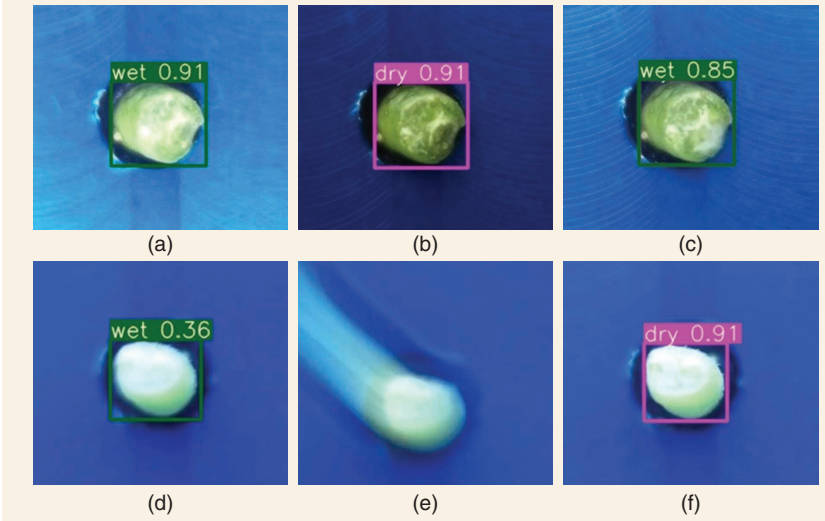

FIGURE 14. Examples of how ambient light variation and the motion of the pump-up chamber can impact the dataset. (a)–(c) The effect of ambient light changes on the xylem state as it evolves from (a) wet to (b) dry and returns to (c) wet. The true classification is dry in (a) and (b) and wet in (c). (d)–(f) Motion effects impact the detection of the stem, in addition to resulting in classification shifts in the xylem wetness state: (d) wet to (e) no detection and then (f) dry. In all the cases, the true classification is wet.

Figure 14. However, and despite those incorrect state detections, all the networks were able to properly detect the stem; this feature can be exploited to facilitate network fine-tuning with additional datasets, such as the second dataset part considered herein.

In an effort to assess the effect of fine-tuning on network performance, we retrained the tuned YOLO v7 network using a small sample of the newly obtained data, that is, 1,180 images, with a ratio of 50% for each wet/dry state. Owing to the fact of training on a small dataset, the model was trained on a Tesla P100 GPU for only 30 epochs, at an initial learning rate of 0.01. An SGD optimizer with a momentum of 0.937 and a weight decay of 0.0005 was used for optimization. The obtained results (the final entry of Table 8) demonstrate a considerable improvement in detection since the accuracy increased from 0.75 to 0.98, suggesting that the structure can now detect the stem in difficult scenarios. The recall increased modestly, to 0.57, yet the fine-tuned network was able to provide over 50% more accurate detections. These results demonstrate that more extended fine-tuning, but not a full network retraining, may, in fact, be sufficient for the methods developed herein to generalize to other tree crop species as well; extending this dataset is part of ongoing work.

> “EVEN MODEST IMPROVEMENTS IN IRRIGATION PRACTICES CAN HAVE HUGE IMPACTS, ESPECIALLY IN THE SEMIARID SOUTHWESTERN UNITED STATES.”

DISCUSSION AND OUTLOOK

CONTRIBUTIONS AND KEY FINDINGS

In this work, we have presented what, to the best of our knowledge, is the first robot-assisted system for SWP measurements. The value of this system lies in its capability to autonomously collect samples at scale by automating time-consuming (e.g., leaf collection) or error-prone processes (e.g., identification of the status of the leaf xylem). These are core building blocks toward enabling complete robot autonomy in physical specimen sampling and transport in the field.

Notably, accurate SWP measurement is instrumental to implement precision agriculture practices to conserve water. Our work demonstrated how it is possible to employ learning-based machine vision to help determine the dry/wet status of the xylem. Results from testing with the static benchtop pressure chamber demonstrated that we can provide a stable, crisp, and high-resolution and high-refresh-rate video feed that is annotated in real time and automatically determines the wetness status of the xylem. This alone can be a major aid to reduce false positive xylem status determination as well as human errors and reading variability when compared to the “look-through-the-hole” approach that is currently the only way to manually observe the wetness status of the xylem. Further, we believe that the produced datasets can stimulate further research on robotics, machine perception, and agronomy and lend themselves to help train human operators in determining SWP via the pressure chamber method.

TABLE 8. The inference results on the dataset evaluating the generalization capacity.

NETWORK	PRECISION	RECALL
YOLO v5 base	0.38	0.34
YOLO v6 base	0.43	0.32
YOLO v7 base	0.59	0.47
YOLO v5 tuned	0.38	0.34
YOLO v6 tuned	0.56	0.35
YOLO v7 tuned	0.75	0.49
YOLO v7 fine-tuned	0.98	0.57

DIRECTIONS FOR FUTURE WORK

The project outcomes achieved thus far enable various opportunities for future work. One key direction involves enhancing the detection of xylem water expression by giving heavier consideration to smoothing the transitional phase. Further, we seek to extend the dataset and evaluate the models’ performance in other tree crops, such as olive and almond trees. Preliminary findings from evaluating the generalization capacity indicated that fine-tuning the networks, or fully retraining with additional data, can help expand the method to other tree crops. Two critical factors affecting the generalization capacity (especially when using the manual pump-up chamber) are ambient light variations and induced motion. The former can be addressed by possibly adding a cover around the exposed stem. The latter is a challenge inherent to the pump-up chamber, which operates by manual upward/downward strokes (like a pump), but image stabilization techniques can be employed. Nonetheless, we would recommend the use of the static benchtop chamber if possible (at the expense of higher acquisition and operating costs) since it can lead to consistent and high-quality footage.

During the leaf cutting experiments, we observed that some successful cuts were not accurate enough to be used for SWP analysis (i.e., leaving a long enough stem to be able to secure the leaf to the pressure chamber’s gasket). One main issue was that the front face of the end effector may push other interconnected leaves and/or branches away; hence, the linear approach may not always suffice. Ongoing work aims to improve the end-effector design to a more minimal assembly and, in addition, enhance the perception capabilities of the end effector to determine when the stem is aligned with the cutting plane to trigger the cutting action. Also relevant is extending the vision pipeline to determine and localize leaves bagged in reflective foil bags (which allows leaves collected in SWP analysis to reach an internal equilibrium, thus providing more accurate information regarding the whole tree’s water stress).

Another aspect worth exploring is whether it is possible to mount the pressure chamber directly on the robot to eliminate

the need for manual insertion of leaves into the pressurized chamber. The main challenge to achieve this functionality is in manipulating the stem without compromising its integrity for assessing the leaf xylem.

Finally, more extensive in-field experimentation is needed to test the validity of the system integration we have developed so far. One critical aspect is to further vet these developed tools through growers and agronomists, who can further assess the advantages and limitations of the robotic means to estimate crops' needs for watering we have developed to date.

ACKNOWLEDGMENT

We gratefully acknowledge the support of U.S. Department of Agriculture National Institute of Food and Agriculture, under Grants 2021-67022-33452 and 2021-67022-33453, and the University of California, under Grant UC-MRPI M21PR3417. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

AUTHORS

Amel Dechemi, Department of Electrical and Computer Engineering, University of California, Riverside, Riverside, CA 92521 USA. E-mail: adech003@ucr.edu.

Dimitrios Chatziparaschis, Department of Electrical and Computer Engineering, University of California, Riverside, Riverside, CA 92521 USA. E-mail: dchat013@ucr.edu.

Joshua Chen, Department of Electrical and Computer Engineering, University of California, Riverside, Riverside, CA 92521 USA. E-mail: jchen142@ucr.edu.

Merrick Campbell, Department of Electrical and Computer Engineering, University of California, Riverside, Riverside, CA 92521 USA. E-mail: mcamp077@ucr.edu.

Azin Shamshirgaran, Department of Computer Science and Engineering, University of California, Merced, Merced, CA 95343 USA. E-mail: ashamshirgaran@ucmerced.edu.

Caio Mucchiani, Department of Electrical and Computer Engineering, University of California, Riverside, Riverside, CA 92521 USA. E-mail: caio.mucchiani@gmail.com.

Amit Roy-Chowdhury, Department of Electrical and Computer Engineering, University of California, Riverside, Riverside, CA 92521 USA. E-mail: amitrc@ece.ucr.edu.

Stefano Carpin, Department of Computer Science and Engineering, University of California, Merced, Merced, CA 95343 USA. E-mail: scarpin@ucmerced.edu.

Konstantinos Karydis, Department of Electrical and Computer Engineering, University of California, Riverside, Riverside, CA 92521 USA. E-mail: kkarydis@ece.ucr.edu.

REFERENCES

[1] D. Bochtis, L. Benos, M. Lampridi, V. Marinoudi, S. Pearson, and C. G. Sørensen, "Agricultural workforce crisis in light of the COVID-19 pandemic," *Sustainability*, vol. 12, no. 19, 2020, Art. no. 8212, doi: 10.3390/su12198212.

[2] L. Booth and S. Carpin, "Distributed estimation of scalar fields with implicit coordination," in *Distributed Autonomous Robotic Systems* (Springer Proceedings in Advanced Robotics Series), Monbellard, FR. Cham, Switzerland: Springer, 2023.

[3] M. Campbell, A. Dechemi, and K. Karydis, "An integrated actuation-perception framework for robotic leaf retrieval: Detection, localization, and cutting," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2022, pp. 9210–9216, doi: 10.1109/IROS47612.2022.9981118.

[4] C. Li et al., "YOLOv6: A single-stage object detection framework for industrial applications," 2022, *arXiv:2209.02976*.

[5] A. Fulton, J. Grant, R. Buchner, and J. Connell, *Using the Pressure Chamber for Irrigation Management in Walnut, Almond and Prune*. Davis, CA, USA: University of California Agricultural and Natural Resources, 2014. [Online]. Available: https://escholarship.org/uc/item/2m2719gm

[6] Z. Ge, S. Liu, Z. Li, O. Yoshie, and J. Sun, "OTA: Optimal transport assignment for object detection," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit. (CVPR)*, 2021, pp. 303–312, doi: 10.1109/CVPR46437.2021.00037.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015, doi: 10.1109/TPAMI.2015.2389824.

[8] J. R. Jankowski, "Consumptive water use in California's Sacramento-San Joaquin delta: A comparison of estimation methods and field data, with implications for water right diversion reporting," Univ. of California-Davis, Davis, CA, USA, 2018. [Online]. Available: https://www.proquest.com/docview/2193403350?pq-origsite=gscholar&fromopenview=true

[9] X. Kan, H. Teng, and K. Karydis, "Multi-robot field exploration in hex-decomposed environments for Dubins vehicles," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, 2019, pp. 449–455, doi: 10.1109/ROBIO49542.2019.8961849.

[10] X. Kan, H. Teng, and K. Karydis, "Online exploration and coverage planning in unknown obstacle-cluttered environments," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 5969–5976, Oct. 2020, doi: 10.1109/LRA.2020.3010455.

[11] X. Kan, T. Thayer, S. Carpin, and K. Karydis, "Task planning on stochastic aisle graphs for precision agriculture," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 3287–3294, Apr. 2021, doi: 10.1109/LRA.2021.3062337.

[12] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit. (CVPR)*, 2018, pp. 8759–8768, doi: 10.1109/CVPR.2018.00913.

[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit. (CVPR)*, 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[14] C. M. Rodriguez-Dominguez et al., "Leaf water potential measurements using the pressure chamber: Synthetic testing of assumptions towards best practices for precision and accuracy," *Plant, Cell Environ.*, vol. 45, no. 7, pp. 2037–2061, Apr. 2022, doi: 10.1111/pce.14330.

[15] G. D. Schaible and M. P. Aillery, "Water conservation in irrigated agriculture: Trends and challenges in the face of emerging demands," U.S. Dept. Agriculture, Econ. Res. Service, Tech. Rep. EIB-99, Sep. 2012.

[16] P. F. Scholander, H. T. Hammel, E. A. Hemmingsen, and E. D. Bradstreet, "Hydrostatic pressure and osmotic potential in leaves of mangroves and some other plants," *Proc. Nat. Acad. Sci. USA*, vol. 52, no. 1, 1964, Art. no. 119, doi: 10.1073/pnas.52.1.119.

[17] A. Shamshirgaran and S. Carpin, "Reconstructing a spatial field with an autonomous robot under a budget constraint," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2022, pp. 8963–8970, doi: 10.1109/IROS47612.2022.9981128.

[18] A. Swarup et al., "Strawberry plant wetness detection using color and thermal imaging," *J. Biosyst. Eng.*, vol. 45, no. 4, pp. 409–421, Feb. 2021, doi: 10.1007/s42853-020-00080-9.

[19] G. Vellidis, V. Liakos, W. Porter, M. Tucker, and X. Liang, "A dynamic variable rate irrigation control system," in *Proc. Int. Conf. Precis. Agriculture*, 2016, vol. 13, pp. 1–9.

[20] H. Vila, I. Hugalde, and M. D. Filippo, "Estimation of leaf water potential by thermographic and spectral measurements in grapevine," *Revista de Investigaciones Agropecuarias*, vol. 37, no. 1, pp. 46–52, 2011.

[21] J. S. Wallace and P. J. Gregory, "Water resources and their use in food production systems," *Aquatic Sci.*, vol. 64, no. 4, pp. 363–375, Dec. 2002, doi: 10.1007/PL00012592.

[22] C. Wang, A. Bochkovskiy, and H. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," 2022, *arXiv:2207.02696*.

[23] C. Wang et al., "CSPNet: A new backbone that can enhance learning capability of CNN," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit. (CVPR) Workshops*, 2020, pp. 1571–1580, doi: 10.1109/CVPRW50498.2020.00203.

[24] S. Wang, L. Ren, Y. Liu, Z. Han, and Y. Yang, "Mechanical characteristics of typical plant leaves," *J. Bionic Eng.*, vol. 7, no. 3, pp. 294–300, Sep. 2010, doi: 10.1016/S1672-6529(10)60253-3.

[25] T. Zhao, B. Stark, Y. Chen, A. L. Ray, and D. Doll, "Challenges in water stress quantification using small unmanned aerial system (sUAS): Lessons from a growing season of almond," *J. Intell. Robot. Syst.*, vol. 88, nos. 2–4, pp. 721–735, Mar. 2017, doi: 10.1007/s10846-017-0513-x.

[26] Q. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," 2018, *arXiv:1801.09847*.