Vision-guided robotic leaf picking

Benjamin Joffe, Konrad Ahlin, Ai-Ping Hu and Gary McMurray

June 11, 2018

Vision-guided robotic leaf picking

Benjamin Joffe[1], Konrad Ahlin[2], Ai-Ping Hu[1], Gary McMurray[1]

Abstract— **The key focus of precision agriculture is the ability to monitor the health of individual plants. While visual monitoring was made easier by incorporating autonomous ground and aerial vehicles, regular sampling of leaves and soil to perform chemical analysis is still necessary. The focus of this work is to propose a leaf sampling system that uses a monocular camera and a 6-DOF robot arm to detect, track, and pick healthy and unhealthy leaves without prior knowledge about the plant. The approach focuses on the challenges of operating in the real world with the aim of increasing the robustness of the system.**

I. INTRODUCTION

The need to monitor crops in fields and orchards is a well-recognized challenge. Crop monitoring for biotic yield-reducing factors (pest organisms such as insects, plant pathogens, and weeds) and abiotic stresses (such as inadequate moisture and nutrient levels) is a pivotal component of integrated crop and pest management systems, but manual crop scouting by growers or crop consultants is often time- and cost-prohibitive. Multi- and hyper-spectral satellite imagery and, more recently, unmanned aerial vehicles (UAVs) has been used as a method of remote crop monitoring. Although such systems can detect plant stresses, they are not capable of autonomously collecting samples for identification and verification of the cause of the stress symptom. Improved field scouting and sampling are instrumental in providing earlier detection of pests as well as abiotic yield-reducing factors, thereby preventing crop loss and improving the efficacy of agrichemical applications.

In order to identify stressed plants we use individual plants data, particularly their growth rates computed using 4D maps based on UAV imagery. An unmanned ground robot with a robotic arm is sent to each of these plants to collect leaf and soil samples that will be analyzed in a laboratory to identify the source of the stress. This paper will discuss our approach to autonomous leaf sampling.

Leaf picking is a challenging task due to high variability of leaves, relatively small size, and the fact that the system needs to be robust to various weather conditions in the field. An additional constraint is the rough operating environment where the system's components cannot be expected to maintain calibration due to vibrations from terrain and the tractor's engine. We demonstrate that a vision-based approach can be used to effectively pick leaves using a robotic arm with an eye-in-hand monocular camera. The approach consists of performing a scan of the plant while detecting and tracking leaves, and using matched features to build the estimate of the leaves' 3D positions.

In each frame leaves are detected using a Deep Neural Network that is capable of discriminating between healthy and unhealthy leaves. The SURF features [1] are computed in order to match the leaves in spatial and temporal domains, thus constructing a dictionary of candidate leaves having information from multiple views. The matched SURF points associated with a candidate leaf from multiple views are coupled with the camera poses inferred from the robot arm odometry. By combining matched feature points in the image domain with real-world camera positions, a point cloud of the leaf is constructed in real-world coordinates [2]. This procedure combines what [3] refers to as "Open-loop visual control" and "Visual Servo Control". The position of the leaf is estimated over a series of images, using techniques such as Image-Based Visual Servoing (IBVS) [4] and Monocular Depth Estimation [5] to control for accumulated error. We perform filtering of the candidates and compute confidence metrics during the procedure to control the scanning and picking approach steps of the process. This results in a system that is able to track a large number of leaves, dynamically switch between control schemes to maximize the accuracy of leaf position estimate, and be robust to errors in leaf detection and tracking. The following sections will discuss in more detail the vision algorithms, controls pipeline, as well as present the experimental evaluation of the system.

II. VISION

Our approach requires addressing several fundamental computer vision problems. First, it requires accurate identification of leaves in the image. Second, it requires tracking candidate leaves between several frames. Finally, it requires matching feature keypoints between several frames. Additionally, we evaluate applying instance-based mask segmentation to the problem to ensure the feature keypoints belong to the target leaf and not other leaves in the bounding box.

A. Detection

Our prior work [5] demonstrated the applicability of Deep Neural Networks for the leaf detection problem. In particular, complex and variable leaf appearance, challenging backgrounds, and changing natural lighting make devising any hand-crafted features very difficult for this problem. The goal of discriminating between healthy and unhealthy leaves makes the task even more challenging.

[1] Benjamin Joffe, Ai-Ping Hu, and Gary McMurray are with the Georgia Tech Research Institute, USA. {benjamin.joffe, ai-ping.hu, gary.mcmurray}@gtri.gatech.edu

[2] Konrad Ahlin is with the Woodruff School of Mechanical Engineering, Georgia Institute of Technology, USA. kahlin@gatech.edu

Our object detector is based on a Faster R-CNN architecture [6] that integrates a region proposal network with the classification and bounding box regression network. As our feature extractor during the experiments we used Resnet101, although smaller MobileNets and VGG provide sufficient accuracy for the system to function. While some lighter architectures are applicable for the problem (e.g., SSD-based approaches [7]), they generally perform worse on smaller objects. Additionally, Faster R-CNN can be used as a base for additional tasks, such as semantic segmentation (see section II-C).

We perform transfer learning by using a model pre-trained on the ImageNet [8] and COCO [9] datasets and fine-tuning it on a dataset of 138 images, each containing on average 30 leaves. To improve the model's generalizability, the dataset contains images of the leaves from the field in different weather conditions and stages of growth, as well as example images indoors. The experimental results on an indoor plant (see section IV-B) acquired in the setting different from that in the dataset images show the model's ability to pick up general representation of leaves. The training images contain examples of healthy and unhealthy leaves in the field, as identified by an expert. The focus of the work is to identify the abiotic stress (caused by water or nutrients deficiency), so the primary signs of an issue are the yellowish color and more visible leaf veins. We treat healthy and unhealthy leaves as two different classes. This allows us to select which set of leaves to track and sample during the test time. Example detections on the leaves in the field are presented in Fig. 1.

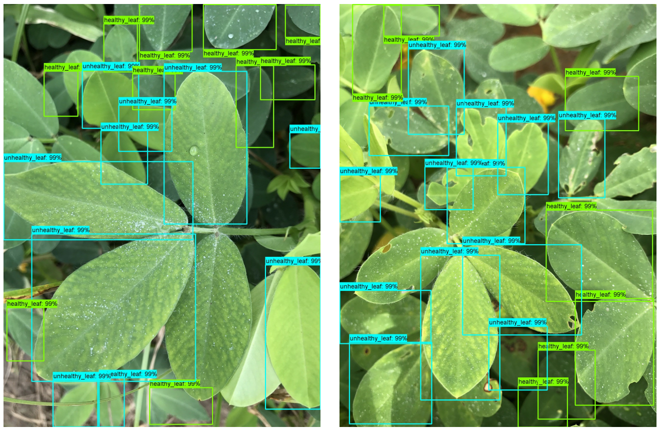

Fig. 1: Detection of healthy and unhealthy leaves.

B. Leaf and Feature Tracking

Once the leaves in the image are identified, we need to track them between several frames. Considering that matched feature keypoints is a required input for the distance estimate, the tracking is based on SURF features [1]. For each new frame, we create a binary mask containing detected leaves and compute SURF features for those leaves. We match the features to N previous frames, which typically results in accurate matches for candidate leaves between frames (see Fig. 2). To filter outliers, we run RANSAC for each leaf's matches to get the final list of matched features. Next, we perform a generic check of the computed homography to make sure it is not extreme, which corresponds to a wrong transformation. We simply take four corners of the image and compute the ratio of the area within the corners before and after applying the transformation. If the ratio is unreasonably high (> 4) or small (< 0.25), we reject the match. Optionally, we compute an epipolar constraint based on known camera movement, which allows pre-filtering potential leaf matches. However, this is beneficial only if one target leaf is tracked at a time and is disabled when matching all the leaves between frames in one step.

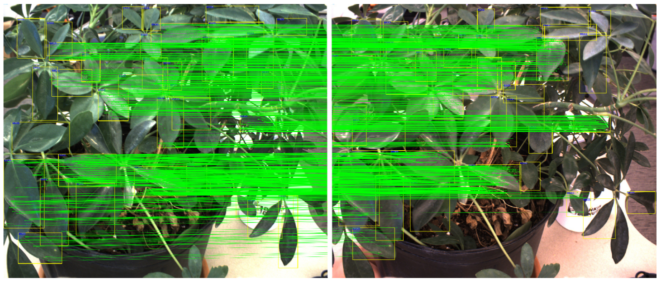

Fig. 2: Matched leaf features between two frames.

The algorithm results in a list of candidate leaves, each containing pairs of matched features and associated frame timestamps. Matching deeper than one previous frame allows us to handle the event when a given leaf failed to be detected. Thus, the controls algorithm has multiple candidates presented to it after each move allowing it to compare several leaves and pick the best candidate. Crucially, temporarily or completely lost leaf track does not require restarting the leaf-picking procedure, as all possible leaf tracks are being initialized and followed during the candidate search phase.

C. Instance-based semantic segmentation

Finally, we address the issue that may affect the accuracy of distance estimate in some cases: features outside of the target leaf. As previously discussed, the individual leaves are described by a bounding box produced by the object detector. Often this leads to the background and other leaves being present in the corners of the bounding box. Since the points in the background reside on a different plane they will affect the distance estimate to the leaf. To mitigate the issue, we trained the leaf detector to additionally compute a segmentation mask for each leaf that can separate the pixels belonging to the leaf from the background within its bounding box. Thus, instead of using mask based on bounding boxes to compute SURF features, we use the associated segmentation mask.

It is known that producing segmentation ground-truth is very labor-intensive, particularly for small and numerous objects like leaves. Hence, we implemented the algorithm that automatically generates masks from bounding boxes based on the Grabcut algorithm [10]. The insights to training from weak labels are available in [11]. Fig. 3 visualizes the segmentations based on the masks used for training.

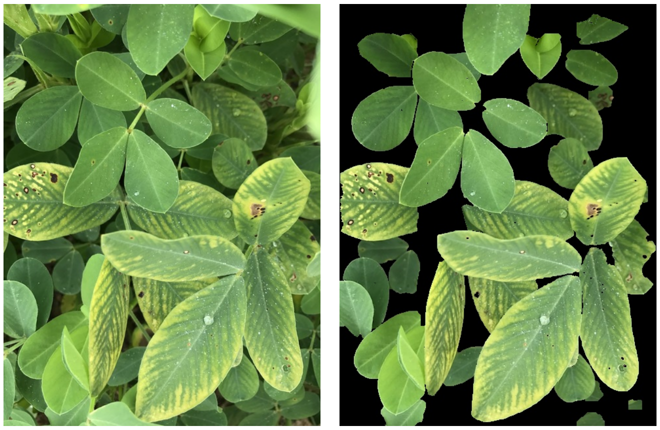

Fig. 3: Automatic segmentation based on Grabcut used for dataset labeling. Left: original image, Right: computed segmentation mask applied to the image.

We trained the instance-based semantic segmentation model using Mask R-CNN [12], the network based on Faster R-CNN that adds a mask regression head. We used an implementation of the approach from Tensorflow Object Detection API. [13]. Examples of predicted masks are presented in Fig. 4.

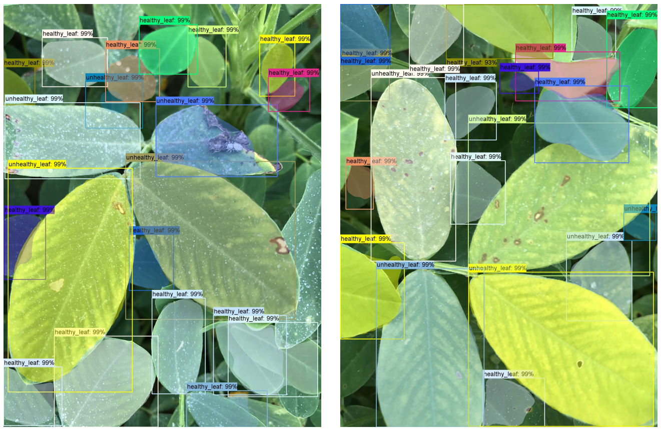

Fig. 4: Predicted bounding boxes and segmentation masks.

III. Controls

Manipulators are often used in situations where the task is known before execution. However, in this research, the manipulator is called on to gather leaf samples: a task which is nearly completely unstructured. Nothing about the size or position of the leaf is assumed to be known other than a general approximation of the dimension of the leaves and the general relative location of the peanut plant (within view of the camera). Thus, the challenge of this research is to localize the peanut leaf position relative to the manipulator position accurately enough so that the end effector can pick the leaf and collect it for sampling.

A. *Previous Method: Single Leaf Approach*

In the previous paper [5], the "Monoscopic Depth Analysis" approach was discussed: a method by which the leaf position was determined through multiple images in multiple camera locations. This method is quite reliable. It accurately gives the leaf position with enough confidence that the leaves are able to be manipulated. Furthermore, the MDA approach does not rely on assuming geometric information about the leaves that are to be manipulated, allowing for the method to be generalizable to other systems if required. However, the system has two aspects that are undesirable:

1) Erroneous data at the beginning of the cycle impacted convergence.
2) Leaves had to be tracked subsequently from beginning to end to be harvested.

1) Erroneous Data: The reason why erroneous data at the beginning of the cycle could impact the convergence is because the system assumes it has an accurate estimation of the leaf position that it is trying to control against. If the error of the leaf position is wildly inaccurate, then the initial commands to control that error will not be directed towards converging on the target. These commands could cause the manipulator to behave in an undesirable motion: leading to a failed attempt at capturing the leaf.

There are three main sources of erroneous data within this system: incorrect point correspondence, correspondence of points not within the leaf, and mispositioning of the camera. Incorrect point correspondence leads to faulty distance estimation, and correspondence of points not within the leaf leads to a distance estimate that does not accurately describe the leaf position. Each of these cases has been mitigated by methods discussed in this paper. High resolution imaging allows for more detected feature points, allowing for a greater confidence thresholding in correspondence and more points to accurately average the leaf position. The mask segmentation ensures that the points found for correspondence are derived from the leaves and not from the background. Camera mispositioning can arise if the base or the joints have slack that is not accounted for in forward kinematic solutions. This issue is resolved by mechanically ensuring that the camera is stable on the end-effector. However, even though the sources of error have been minimized, past testing has shown that an intermediate step is desirable to confidently begin convergence towards a target. This procedure will be discussed in the following section.

2) Subsequent Leaf Tracking: A major time consideration of the previous leaf picking approach was the limitation that a leaf had to be tracked throughout all the stages of the control algorithm. The implication of this method is that if the identification process of a leaf fails between camera movements, the information about the leaf position is lost and the routine has to begin again. Furthermore, even if the leaf is found using this method, the images that are being used to track the leaf are only valuable for that one leaf. Even though other viable candidates are likely within the image space, data on those leaves are discarded. This is a

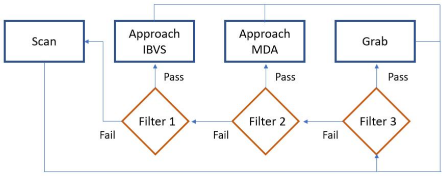

Fig. 5: Flowchart depicting logic used to locate and grab leaves. At each stage, the strictest filter is applied, and if it is passed, the leaf is grabbed. If the filter fails to produce a meaningful point cloud, the next most stringent filter is applied, resulting in appropriate motion from the manipulator.

significant waste in data resources, and although the system works, maintaining data on multiple leaves at a time would be more efficient.

B. Current Method: Multi-Leaf Approach

The current working method has two major additions to the task of harvesting leaves for sampling. The first modification is the addition of an Image-Based Visual Servoing (IBVS) routine for initial convergence. Fundamentally, IBVS methods require feature points in the image space to create an error that can be minimized. This is why the IBVS method was initially dismissed from this application: leaves are variable enough that reliable feature points or characteristics cannot be identified in a meaningful way. However, IBVS could be used to determine initial motion by examining the bounding box used to define the leaf in the image space. By creating a "desired" bounding box in the center of the image, and by setting a low gain on the motion of the arm, even if the bounding box created by identifying the leaves is dissimilar to the desired bounding box, the error values are sufficient to initially determine the motion of the arm towards convergence with the leaf. This initial motion allows for multiple camera angles to examine the leaf, and thus the position of the leaf can be more accurately estimated. Once a certain confidence in the position of the leaf is reached, the MDA approach can take over and minimize the error in Cartesian space. Thus, the approach phase of the previous pipeline has now been broken into: "Approach IBVS" and "Approach MDA". This algorithm is represented in Fig. 5. In this figure, filters are representative of a confidence level about the shape of the point cloud. The confidence level regarding the point cloud and its association to the desired leaf dictates the action of the manipulator.

Another significant change to past algorithms is the consideration of multiple leaves from a given image. This allows for the collection of information about a leaf while approaching other leaves. Thus, the pipeline can now occur for multiple leaves at a time, speeding up the process of estimating leaf position and collection. A diagram of the curent picking pipeline demonstrating interaction of vision and control parts is shown in Fig. 6.

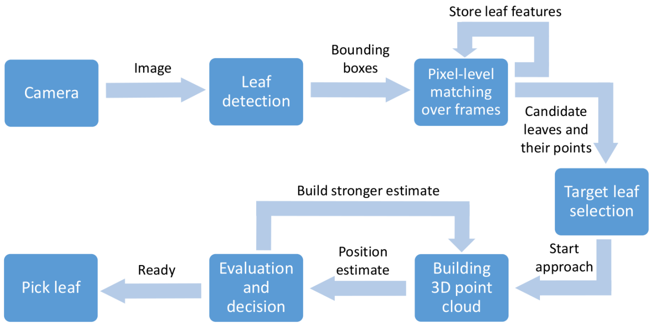

Fig. 6: Diagram showing flow of information in the complete leaf-picking pipeline.

IV. Experimental Results

A. Leaves detection and disease identification

The Mask R-CNN ResNet101 model used in the experiment was trained for 40K iterations with the initial learning rate of 3e-3. We reduced the learning rate by 10X after 20K and 30K iterations. The following data augmentation was used: random horizontal, vertical flip, and 90-degree rotation. The performance of healthy and unhealthy leaves detection was evaluated using Pascal VOC metric [14], which requires intersection over union (IoU) value of at least 0.5. The model achieves mean average precision (mAP) of 0.753 (see Table I). The experiment presented in this paper focuses on leaf picking using a healthy plant, and the practical evaluation with both healthy and unhealthy leaves present will be performed as part of our future work.

Category	**AP**
Healthy leaf	0.759
Unhealthy leaf	0.746
Mean	0.753

TABLE I: Leaf detection performance using Pascal VOC metric.

B. Leaf Sampling

The success of the leaf sampling is reliant on two separate functions: the ability to estimate the leaf position in Cartesian space and the action of grasping the leaf. To accomplish this, feature points are corresponded in disparate images in order to estimate the position of the feature point relative to the camera. The collection of these points is then built into a point cloud for individual leaves. Once the leaf position is confidently estimated, the manipulator reaches for the leaf position and attempts a grasp. For the purpose of the application, it is assumed that the manipulator is sufficiently accurate. Therefore, the priority of this research is to attempt to describe the leaf pose relative to the manipulator as accurately as possible.

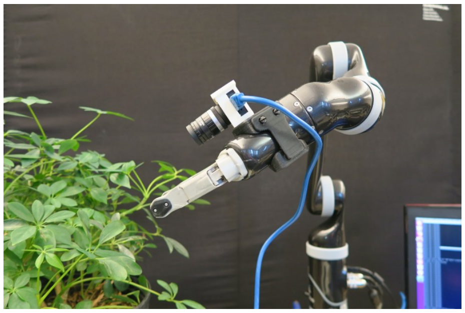

Fig. 7: Experimental setup to determine the effectiveness of the leaf acquisition algorithm. In this setup, the camera begins in a position roughly the same distance to the plant as it would in a field test.

The error exists within the data collected. The position estimation relies on the target being stationary, but small amounts of air movement and other external variables may cause the leaves to move. Furthermore, the position of the camera is subject to small variations. Also, the corresponding feature points may not always be accurate. All of these variables and more can lead to erroneous data in the point cloud. The leaf is estimated to the best of the system's abilities by the methods previously mentioned, but in order to determine where the leaf is, eventually a filtering method is employed that determines the algorithm's confidence in a leaf's position. One of the few assumptions made is that the leaves are roughly flat, and thus, the estimated point cloud should resemble a plane. Therefore, the system can be confident in the position of a leaf if the estimated points lie on a plane relatively close together. The following steps are taken in filtering out "good" vs. "bad" leaf bounding boxes to determine which leaves should be attempted for harvesting.

1) Fit a plane to the leaf data.
2) Measure the normal distance from each point to the plane.
3) Measure the radial distance on the plane from each point to the center.
4) Find the mean and standard deviation values for the distance to the plane and distance to the center.
5) Apply a confidence threshold based on the distance to the plane and radial distance to determine if the point cloud is a leaf.

These steps help to ensure that the point cloud is roughly a plane and that the points are grouped, giving the best insurance possible that erroneous data has not overwhelmed the estimation of the leaf position. An example of a trial set is shown in Figs. 8 and 9. More information on the data set is listed in Table II. The results in these figures show the confident location of various leaves identified by the image processing. Even though only about 20% of the possible leaves that were identified in the image space had suitable point clouds, the 41 leaves found far outweighs the eight leaves required for sampling.

Leaf Filtering Example	
Images Taken:	12
Leaves Found in Image Space:	200
Leaves with Corresponding Points:	76
Point Clouds that Passed Filtering:	41

TABLE II: Data regarding the trial in generating the leaf point clouds.

Trial	Grabbed	Missed	% Grabbed per Total
1	10	2	83.33
2	7	0	100.00
3	10	0	100.00
4	5	2	71.42
5	7	0	100.00
6	10	0	100.00

TABLE III: Given 10 images per trial, this table lists the number of leaves successfully grabbed versus the number of leaves missed.

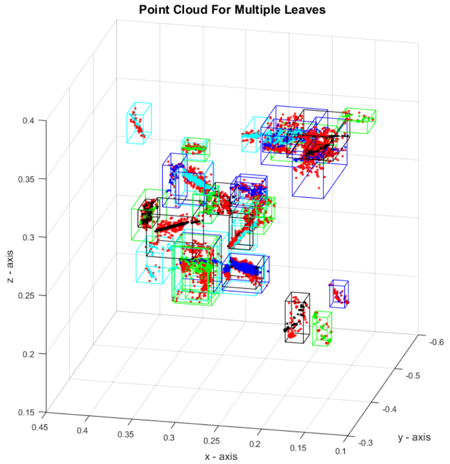

Fig. 8: Example of the plant point cloud. The estimated position of the feature points has been used to limit the chosen bounding boxes to only those which are likely to be leaves. In this image, the red points are the raw data values and the non-red points are the values rectified to a plane. Each leaf is contained in a bounding box of a corresponding color.

Given a suitable point cloud, the leaves can now attempt to be sampled. This was done in a lab environment, shown in Fig. 7. The algorithm in Fig. 5 was implemented and resulted in about 92% of the leaves grabbed that the system attempted to grab, which can be seen in Table III. Even though this was a lab setting, it is expected to behave similarly in the field. The lighting conditions should not be a factor in the imaging of the leaves, and the mechanical setup will be similar, except with the robotic arm attached to a tractor instead of a table. Importantly, this test measured if the leaf was “grabbed” as the test did not attempt to “pick” the leaf for sampling. This procedure was performed to preserve the health of the plant during extensive testing. This experiment gives a good indication that the system will be able to meet performance expectations and a suitable number of samples can be obtained in a full experimental setup.

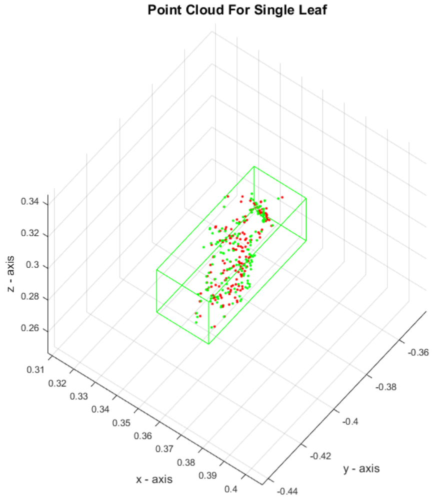

Fig. 9: Example of a single leaf point cloud. Based on the associated feature points, this point cloud is likely the position of a leaf. In this image, the red points are the raw data values, and the green points are the values rectified to a plane.

V. Conclusions

This paper discussed a flexible leaf-picking approach designed for the field and, thus, addresses the challenges of working in the real world. These challenges include variable outdoor operating environments, uniqueness of target objects, and difficulty of having a persistent calibration between any external sensors and the robotic arm. This informed some of our fundamental design choices from using Deep Neural Networks to having eye-in-hand camera-based controls, which allow the system to operate with few assumptions about the robot platform and the environment.

We focused on the components that increase the robustness and convergence speed of our approach and addressed the issues identified during the initial field test. These components include semantic segmentation based on automatically generated labels, multi-leaf multi-frame tracking, and comprehensive candidate filtering. Our future work will focus on performing a field test of the system in the peanut field with the objective of picking healthy and unhealthy leaf samples from multiple plants as part of a continuous autonomous operation.

Acknowledgment

This material is based upon work that is supported by the National Institute of Food and Agriculture, U.S. Department of Agriculture, under award number 2014-67021-22556.

References

[1] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-up robust features (SURF),” *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[2] R. Kalghatgi and N. Sadegh, “Novel Techniques for Line and Point Detection in a 3-D Environment Using a One or Two Cameras,” pp. 1–6.

[3] Y. Zhao, L. Gong, Y. Huang, and C. Liu, “A review of key techniques of vision-based control for harvesting robot,” 2016.

[4] F. Chaumette and S. Hutchinson, “Part I : Basic Approaches,” *IEEE Robotics and Automation Magazine*, vol. 4, no. December, pp. 82–90, 2006.

[5] K. Ahlin, B. Joffe, A.-P. Hu, G. McMurray, and N. Sadegh, “Autonomous Leaf Picking Using Deep Learning and Visual-Servoing,” *IFAC-PapersOnLine*, vol. 49, pp. 177–183, Jan. 2016.

[6] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” in *Advances in Neural Information Processing Systems 28* (C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds.), pp. 91–99, Curran Associates, Inc., 2015.

[7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in *European conference on computer vision*, pp. 21–37, Springer, 2016.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 248–255, IEEE, 2009.

[9] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *European conference on computer vision*, pp. 740–755, Springer, 2014.

[10] C. Rother, V. Kolmogorov, and A. Blake, “Grabcut: Interactive foreground extraction using iterated graph cuts,” in *ACM transactions on graphics (TOG)*, vol. 23, pp. 309–314, ACM, 2004.

[11] A. Khoreva, R. Benenson, J. Hosang, M. Hein, and B. Schiele, “Simple Does It: Weakly Supervised Instance and Semantic Segmentation,” *arXiv:1603.07485 [cs]*, Mar. 2016. arXiv: 1603.07485.

[12] K. He, G. Gkioxari, P. Dollr, and R. Girshick, “Mask R-CNN,” *arXiv:1703.06870 [cs]*, Mar. 2017. arXiv: 1703.06870.

[13] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, “Speed/accuracy trade-offs for modern convolutional object detectors,” *arXiv:1611.10012 [cs]*, Nov. 2016. arXiv: 1611.10012.

[14] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.