Robotics: Science and Systems 2024
Delft, Netherlands, July 15-July 19, 2024

Demonstrating CropFollow++: Robust Under-Canopy Navigation with Keypoints

Arun Narenthiran Sivakumar[1] Mateus Valverde Gasparino[1] Michael McGuire[2]
Vitor Akihiro Hisano Higuti[2] M. Ugur Akcal[1] Girish Chowdhary[1]
[1]Field Robotics Engineering and Sciences Hub (FRESH), University of Illinois Urbana-Champaign (UIUC)
[2]EarthSense Inc.

***Abstract*—We present an empiricially robust vision-based navigation system for under-canopy agricultural robots using semantic keypoints. Autonomous under-canopy navigation is challenging due to the tight spacing between the crop rows (~ 0.75 m), degradation in RTK-GPS accuracy due to multipath error, and noise in LiDAR measurements from the excessive clutter. Earlier work called CropFollow addressed these challenges by proposing a learning-based visual navigation system with end-to-end perception. However, this approach has the following limitations: Lack of interpretable representation, and Sensitivity to outlier predictions during occlusion due to lack of a confidence measure. Our system, CropFollow++, introduces modular perception architecture with a learned semantic keypoint representation. This learned representation is more modular, and more interpretable than CropFollow, and provides a confidence measure to detect occlusions. CropFollow++ significantly outperformed CropFollow in terms of the number of collisions (13 *vs.* 33) in field tests spanning ~ 1.9km each in challenging late-season fields with significant occlusions. We also deployed CropFollow++ in multiple under-canopy cover crop planting robots on a large scale (25 km in total) in various field conditions and we discuss the key lessons learned from this.**

I. Introduction

Agricultural production faces major challenges due to rising costs and reduced supply of farm labor coupled with strong environmental concerns due to overuse of farm chemicals. Autonomous under-canopy robots have great potential to address these challenges by enabling plant-level monitoring and care. In particular, under-canopy robots that traverse the tight space between rows of crops ($\sim$ 0.75 m) can enable various applications like high throughput phenotyping and crop monitoring, cover crop planting, precise weeding and spraying throughout the growing season which is not possible with over the canopy systems like tractors and drones [35, 36, 37, 38, 54, 45, 40]. A major bottleneck in deploying under-canopy robots in farms is the lack of robust, low-cost autonomous navigation solutions for these challenging environments.

Tractors and drones primarily depend on precise positioning from RTK-GPS for navigation in farms. RTK-GPS accuracy degrades due to multi-path errors under the plant canopy, even with an expensive RTK correction service subscription. LiDAR sensors are expensive and suffer from noise in measurements due to excessive clutter in these environments [24, 56, 49]. The lack of semantic information to distinguish the crop of interest from weeds or other objects limits LiDAR capabilities for this problem. Cameras offer rich information about the scene and are far lower in cost, and hence are a better alternative [49, 69]. Prior works in vision-based agricultural row following were primarily focused on using classical computer vision methods from the over-the-canopy views of tractors. In such a viewpoint, multiple crop rows are visible in the field of view of the camera as equispaced parallel lines making it easier to detect and follow the row. On the contrary, from the under-canopy viewpoint, the two crop rows corresponding to the lane in which the robot is present are visible, with some background of remaining crop rows, and their structure is occluded significantly by the corn leaves as well as weeds and crop residue present in the scene. In addition, there is significant variation in lighting, the appearance of crops and the soil surface throughout the growing season as well as across various farms with changes in management practices and this makes it challenging to use classical computer vision techniques for this problem.

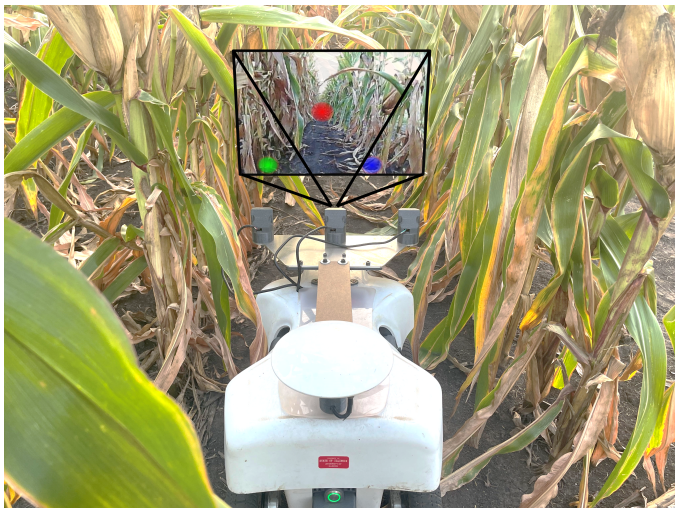


Fig. 1: CropFollow++ is a vision-based navigation system for under-canopy agricultural robots. It uses a neural network to predict the keypoints representing the crop rows from an RGB image and traverses the gap between the crop rows.

Deep learning methods have shown impressive capabilities in learning robust features for various computer vision tasks

Correspondence to {av7,girishc}@illinois.edu

in diverse domains. The question is, for our goal of robust navigation under the plant canopy, what is the appropriate objective for which the deep learning model should be trained? In the similar task of lane following in autonomous driving, learning has been applied for three different task objectives: 1) Mediated perception refers to using learning to detect or segment semantic objects in the image, 2) Direct affordance prediction refers to directly estimating the states of the robot and the environment necessary to follow the lane without explicitly detecting or segmenting objects in the image, and 3) End-to-end control with imitation learning which refers to training a network to directly output the control commands from an input image.

Unlike autonomous driving in which large amounts of expert driving data of humans are available, no large-scale expert data for under-canopy navigation exists to train an end-to-end control. It is because human demonstration data for this task tend to be suboptimal since humans find it challenging to keep the robot perfectly in the middle of the lane under the canopy due to frequent occlusions. Also, end-to-end deep learning architectures are not desired in real-world systems deployed on a large scale because of the lack of interpretability during failures. [49] collected and annotated a large under-canopy dataset with vanishing lines and used it to calculate the state of the robot relative to the crop rows. The dataset was used to train a convolutional neural network to predict these states and they demonstrated significantly better navigation performance than LiDAR baseline in rigorous field experiments. Note that all these experiments were performed on the same robot. This direct affordance prediction approach called CropFollow, though more modular than end-to-end control prediction, is less interpretable and flexible than mediated perception. Also, the network directly predicts only point estimates without any measure of uncertainty in predictions which affects the navigation performance in challenging scenarios with frequent occlusions. Typically, mediated perception approaches for agricultural row following focus on either detecting the crop row lines or segmenting the traversable triangular area in the field of view. However, these approaches require additional heuristics to extract the actual crop row lines from the noisy predictions.

In this paper, inspired by the object pose estimation problem in robot manipulation tasks [55], we propose to use semantic keypoints as a representation for the under-canopy navigation problem. Since crop rows are usually planted in equispaced and parallel straight lines and only one lane is visible in the camera's field of view under the plant canopy, we can parameterize the traversable area by the three semantic keypoints representing the vertices of the triangle formed by the two crop row lines and the bottom of the image - 1) Vanishing point of the crop row lines 2) Intercept of the left crop row line with the bottom of the image 3) Intercept of the right crop line with the bottom of the image. We calculate the robot state from the predicted keypoints using the robot-specific camera intrinsic and roll angle estimation from IMU followed by a model predictive controller that computes the linear and angular velocity to be applied to track the reference path. Since the output of the convolutional neural network are heatmaps representing a distribution for each keypoint, we use the variance of the vanishing point heatmap to define a confidence threshold which enables us to detect occlusion of cameras and improve navigation performance. In real-world controlled field tests of $\sim$ 1.9 km, we show our semantic keypoint approach CropFollow++ results in significantly fewer collisions than CropFollow (13 *vs.* 33) [49].

We deployed this semantic keypoint perception system in large-scale on under-canopy robots developed for a novel application: Cover crop planting. Cover crops are planted to cover the soil during winter and provide various environmental benefits like prevention of soil erosion and nutrient loss, suppression of weed growth, and carbon sequestration. Under-canopy cover crop planting robots can enable planting cover crops earlier in the season at a lesser cost. Moreover, the techniques presented here can be used on under-canopy equipment pulled by high-clearance tractors. Our semantic keypoint perception system was deployed in various field conditions on multiple cover crop planting robots for 25 km.

Our **main contribution** is the largest, empirically robust demonstration of autonomous under-canopy navigation using a novel perception system based on semantic keypoints and the discussion on various failure modes observed and valuable lessons learned from this deployment.

II. Related Work

Recent studies in agricultural robotics have shown significant advancements in autonomous row-following. This specific task involves directing a robot between crop rows, typically for activities like weeding, harvesting, or data collection [12, 57, 54, 51]. The state-of-the-art techniques in this area have emerged through the combination of developments in robotics, computer vision, and artificial intelligence. However, much of the prior research has focused on over-the-canopy scenarios [5, 17, 26, 64, 66] or relatively less complex orchard navigation [1, 7, 46, 53], with only a few studies [56, 19, 63, 22, 15] addressing the more challenging under-canopy situations. Additional notable studies, such as [59, 6, 2], and [9] tackle orchard and over-canopy navigation. However, the exploration of under-canopy navigation remains relatively underrepresented in the field. Navigating below the canopy is a more intricate task due to several factors such as variable lighting conditions, the presence of obstacles at ground level, and the complex visual environment created by the foliage. These conditions necessitate more sophisticated sensing technologies and algorithms than the aforementioned methodologies.

Previous work utilizes various sets of sophisticated sensors. Methods employing LiDARs [34, 67, 24, 56, 19], ultrasonic [13], and infrared sensors [60, 62] may offer robust solutions against varying lighting conditions [3]. Such feature is essential for navigation beneath the canopy where lighting can greatly fluctuate [28]. In addition, these methods are comparatively simpler in computational demands. However,

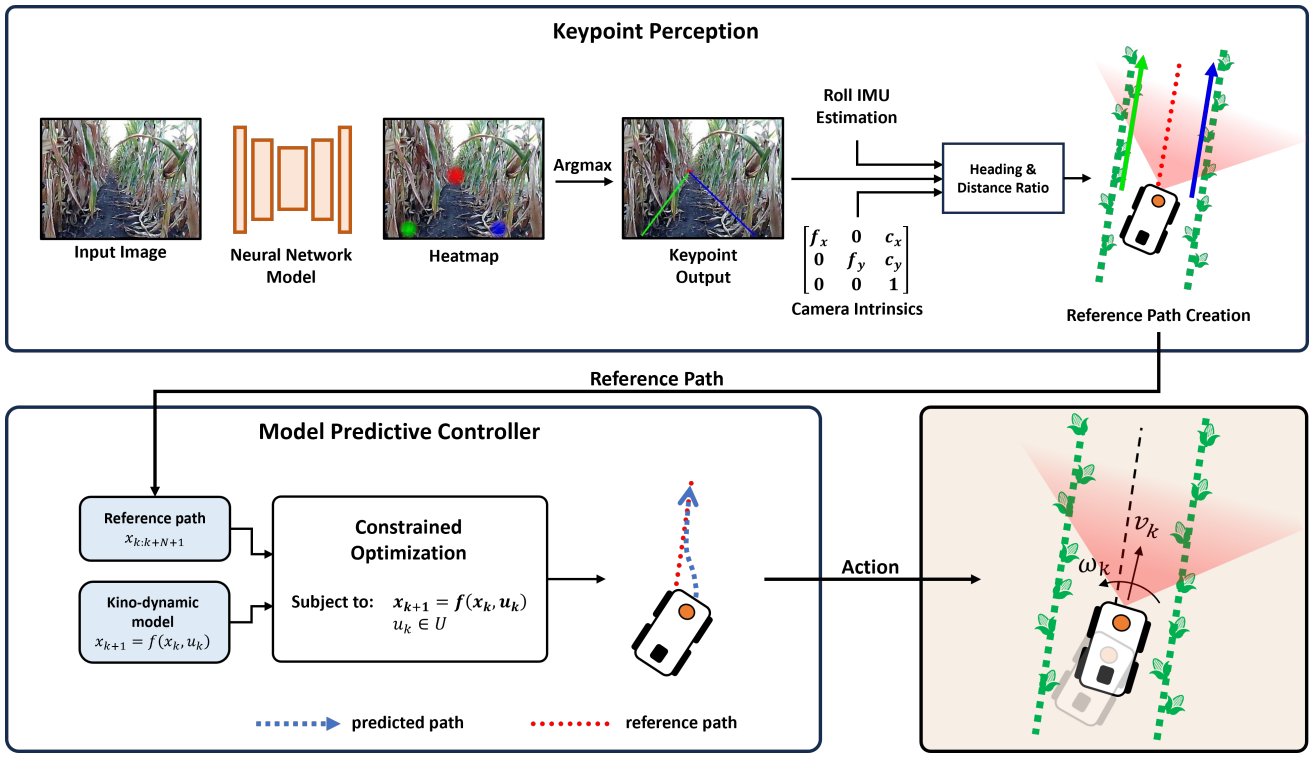


Fig. 2: CropFollow++ overview. The camera RGB image is used as input to our neural network model that predicts keypoints to locate the crop rows. The keypoints are used to create a trajectory that is used as the reference for an MPC to navigate the robot.

sensors like LiDAR and SONAR are limited in their ability to extract semantic information [44, 18, 20]. Furthermore, the substantial cost of LiDAR sensors is a significant constraint on their widespread use in robotics applications [52]. Moreover, the effectiveness of such devices may be compromised due to their susceptibility to environmental elements like mud, dust, and the thickness of foliage, which can impact the precision of the sensors [47, 31].

Techniques based on GPS, particularly those enhanced with Real-Time Kinematic (RTK) positioning [65, 8, 24, 27], offer high accuracy. This technology is especially useful in large fields where precision in long-range navigation is crucial. However, the effectiveness of GPS/RTK systems is hindered by their reliance on satellite signals, which can be obstructed in under-canopy environments or by various external factors [43, 24]. Moreover, these systems can be costly and might not offer the required resolution for detailed tasks such as weeding and harvesting [4].

Employment of cameras coupled with advanced machine learning algorithms for the identification and tracking of crop rows is one of the main paradigms not only in under-canopy row following but also in other sub-fields of agricultural robotics [41, 68, 11, 21, 33] in general. A key strength of this approach lies in its remarkable adaptability [42]; contemporary machine learning models are capable of being trained to recognize and adjust to numerous crop types and row configurations. Particularly, image segmentation [16, 10, 29] and object detection [61, 25] capabilities are of utmost importance for successful under-canopy navigation where the system cannot rely on GPS signals. However, this family of techniques faces challenges with dynamic lighting conditions and demands substantial computational resources to process and analyze visual data, which is limitating for field robots [58].

III. System Design

Figure 2 provides an overview of our proposed keypoint-based under-canopy navigation system. The perception system takes the RGB images from the low-cost camera on the robot as input and outputs three heatmaps. These heatmaps correspond to the three semantic keypoints of interest - vanishing point, left intercept point and right intercept point. The confidence check module uses a heuristic based on the variance of the vanishing point heatmap to check for anomalous scenarios not seen during training. Heading and lateral distance of the robot relative to the crop rows is calculated from the heatmaps using the camera intrinsics and roll estimate from the inertial measurement unit (IMU) if the confidence check is passed. A Model Predictive Controller (MPC) uses the calculated heading and lateral distance, and solves a constrained cost optimization to find the optimal linear and angular velocity to track the reference. We describe these modules in detail below.

Fig. 3: Calculation of heading and distance ratio from keypoints is illustrated here. We use the roll estimation from IMU and calculate the heading angle ϕ from the horizontal offset of the vanishing point from the principal point. The distance ratio is calculated from the ratio of the intercepts of the left and right crop row lines $d_L/(d_L + d_R)$ after correcting for roll, pitch, and heading.

A. Robot Platforms

We use two under-canopy robot platforms - Terrasentia and Cover Crop Robot (CCR) manufactured by Earthsense Inc. in our experiments. Terrasentia is a compact under-canopy robot developed for crop monitoring and high-throughput phenotyping applications. It has a Raspberry Pi3 to interface with motors and sensors such as IMU, wheel encoders, and GPS. It also has an NVIDIA Jetson AGX Orin computer to which we connect the camera and run our navigation modules such as keypoint-based perception and MPC. The control commands from MPC are communicated through the Pi3 to the motors. CCR is an under-canopy robot developed for cover crop planting applications. It has a wider base than Terrasentia to support the load of cover crop seeds. It has a similar sensor interfacing and communication architecture as Terrasentia but uses an Intel NUC 11 computer to run the navigation modules. It also has multiple low-cost cameras mounted on the robot to be robust to occlusion from leaves commonly encountered in these under-canopy settings. Since Terrasentia is a compact robot, we use this as our development platform and we performed our field experiments comparing CropFollow++ and CropFollow on this robot. We deployed CropFollow++ on large-scale on multiple CCR robots.

B. Perception with Semantic Keypoints

Our perception system is a fully convolutional network with U-Net [48] like encoder-decoder architecture. We use a ResNet-18 [23] based encoder and a decoder with bilinear upsampling and convolutional layers. This network is trained to output three 2D heatmaps corresponding to the three semantic keypoints of interest for this task namely the vanishing point, left intercept point, and right intercept point as shown in Fig 2. A spatial softargmax operation is applied to each of the three heatmaps to get the three semantic keypoints of interest. Our output heatmaps are 1/4th in spatial scale compared to the input image.

C. Heading and Lateral Distance Calculation from Keypoints

Figure 3 explains the process of calculating heading and distance ratio from the predicted keypoints. Using the roll angle estimation from IMU, first, the 3 keypoints are rotated along the roll axis. Then using the focal length f of the camera and the vertical offset of the vanishing point after roll rotation from the principal point PP of the camera, we calculate the pitch angle as $\arctan(y/f)$. After applying the homography to the keypoints to correct for the pitch angle, the heading angle is calculated using the horizontal offset x of the vanishing point (after pitch correction homography) from the principal point as $-\arctan(x/f)$. Another homography is applied to the keypoints to correct the heading angle and the distance ratio is calculated using the intercepts of the transformed crop row lines with respect to the horizontal axis along the bottom of the image $d = \lambda d_L/(\lambda d_L + \lambda d_R)$. The lateral distance is calculated from the distance ratio by multiplying with the known perpendicular distance between the rows of crops in the real world. In addition to calculating the heading and distance ratio, we also define a confidence check similar to [55] based on the variance of the vanishing point heatmap.

$$confidence = sigm(3 * \tanh(5 * (1 - \tfrac{\sigma}{5 * \sigma_{vp}})))$$

Here $sigm$ refers to the sigmoid function, $\sigma = 1e-5$ and σ_{vp} refers to the variance of the vanishing point heatmap. We apply a threshold of 0.5 and classify all instances of $confidence < 0.5$ as outliers. We also use a heuristic check based on the known information about the distance between the rows and the camera calibration by computing the upper and lower bounds of pixel locations for the intercept keypoints. These checks help improve the robustness of CropFollow++ to outliers in keypoint model prediction.

D. Model Predictive Controller

To accurately follow the middle of the crop row by using the predictions from the presented neural network model, we design a model predictive controller (MPC). We chose the MPC as the controller because it can handle a non-linear model, as well as deal with the constraints on the motors' maximum speed. For this purpose, we use Eq. (1) to represent the robot's kino-dynamic model, where x_k represents the robot states composed by the 2D position (p_{x_k}, p_{y_k}), and the heading angle θ_k. v_k and ω_k are the linear and angular input velocities, accordingly, that can also be represented as the single input vector u_k. This model is a modified unicycle model, with the

addition of the coefficient ν, which can be understood as a constant friction coefficient used to better represent a skid-steer robot. The value of ν is estimated from data and tuned to provide better navigational performance.

$$x_{k+1} = \begin{bmatrix} p_{x_{k+1}} \\ p_{y_{k+1}} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} \cos(\theta_k) & 0 \\ \sin(\theta_k) & 0 \\ 0 & \nu \end{bmatrix} \begin{bmatrix} v_k \\ \omega_k \end{bmatrix} \tag{1}$$

As illustrated in Figure 2, the Keypoint Perception module generates a straight reference path that represents the middle of the row to be followed by the robot. This path is transformed to a state/control trajectory by sampling $N+1$ points at a sampling time t_s from this line with constant speed v, starting from the closest point to the robot's geometric center. The created trajectory is composed of tuples (x_k, u_k), with $k \in \mathbb{N}$, where the heading angle θ is the angle of the line, v is the desired speed, and $\omega = 0$. In this work, we provide an improvement over [49] by adding the linear speed as a control parameter. In [49], only the angular velocity was controlled, however, the addition of the linear speed as a control parameter adds the capability of decelerating the robot to perform sharper turns, which increases the responsiveness of the system.

We choose an optimization horizon $N+1 \in \mathbb{N}$ and positive definite matrices Q, Q_N, and R to define the cost function expressed in Eq. (2). The matrix Q is responsible for weighting the trajectory error across the horizon, with Q_N being the terminal weight corresponding to the terminal cost. The matrix R weights the control actions and is responsible for making them follow the desirable action values u_k^r. Increasing Q over R increases the reactiveness of the system while increasing R over Q provides a damping-like behavior.

The following finite horizon optimization formulation is solved to obtain a sequence of control actions $u_{k:k+N}$

$$\begin{aligned} \min_{u_{k:k+N}} \sum_{i=k}^{k+N} \left\{ ||x_i\text{-}x_i^r||_Q^2 + ||u_i\text{-}u_i^r||_R^2 \right\} \\ + ||x_{k+N+1}\text{-}x_{k+N+1}^r||_{Q_N}^2 \end{aligned} \tag{2}$$

such that, at every iteration, the optimization framework is subject to the constraints $u_k \in \mathbb{U}$, where $\mathbb{U}$ is the set of velocities that are achievable by the motors in our robot. The first element u_k is used as the control action applied to the motors to make the robot follow the reference path, as shown in Figure 2.

IV. Dataset and Model Training

A. Dataset

We used the dataset from [49]. This dataset contains 25,296 labeled images with vanishing line labels from various growth stages of corn in different lighting conditions. However, this dataset does not contain images from the very late season when the corn plants are brown in color and more frequent occlusion of the camera. Hence we collected and annotated an additional 2,977 images from very late season conditions. This combined dataset of 28,273 images was used to train both our proposed keypoint model as well as CropFollow baseline [49]. Fig 4 shows a sample of the diverse field conditions represented in the training dataset. Note that from the vanishing line labels, we calculate the three semantic keypoints - vanishing point, the intercept of the left crop row line with the bottom of the image, and the intercept of the right crop row line with the bottom of the image whereas CropFollow [49] calculates the heading and distance ratio of the robot relative to the crop rows to train the model. We used a dataset split of 82% training and 18% validation. These 28,273 images were from 54 unique videos and while splitting the dataset we ensured all images from a video were assigned as either training or validation data without mixing.

Fig. 4: Samples of images representing the diversity of the dataset. Our dataset includes all growth stages from early-season corn to very late-season corn and also with large variations in lighting conditions and soil appearance.

B. Model Training

Our labeled dataset represents the three semantic keypoints as discrete pixel locations on the image. During training, we define a 2D Gaussian distribution with variance $\sigma = 1$ for each semantic keypoint centered around the ground truth keypoint pixel. Note that the input RGB image to the network is of size 320×224 whereas the output heatmaps are 1/4th the dimension of the input image i.e. 80×56. The 2D Gaussian distributions that we create as label heatmaps are also of the same dimension 80×56. We use a U-Net [48] like encoder-decoder architecture with a ResNet-18 [23] based encoder that is pre-trained on Imagenet [32] and a decoder with bilinear upsampling and convolutional layers. We used KL Divergence loss and trained this network for 50 epochs with a learning rate of 1e-4. Blur, color jitter and horizontal image flip augmentations were used. With KL Divergence loss on the validation set as the model selection criteria, we chose the model checkpoint trained for 17 epochs which had a mean validation loss of 0.000783.

We train the direct affordance prediction architecture of CropFollow [49] on the same dataset as our proposed method to use as a baseline. In [49], two separate convolutional neural networks were used to predict heading and distance ratio. We found that by normalizing the heading prediction in the loss function, both heading and distance ratio can be predicted

using a single convolutional neural network without a drop in accuracy. We use a Resnet-18 [23] backbone pre-trained on Imagenet [32] followed by three fully connected layers. $L2$ loss on normalized heading prediction and distance ratio was used and the network was trained with a learning rate of 1e-4 for 50 epochs.

V. Offline evaluation and comparison with baselines

We discuss the prediction outputs of a classical and a learning-based segmentation method on different images from our validation dataset and show a comparison with our semantic keypoint representation. We also evaluate the performance of a foundational model for navigation [50] on our under-canopy data and show the visualization. For quantitative evaluations, we compare our semantic keypoint representation in CropFollow++ with the end-to-end perception representation from CropFollow [49] in terms of mean and median L1 error in heading and distance ratio prediction.

A. Qualitative comparison with baselines

1) Segmentation methods as baseline: Prior works in vision-based agricultural navigation in row crops such as corn are primarily focused on over-canopy applications. A common approach in such cases is to use a color segmentation module to mask the plant pixels from the background and a line fitting module on the mask. However, the visual appearance of the image from under the canopy is significantly different. In an over-canopy viewpoint, there is no occlusion of cameras and multiple crop rows are visible as clear lines in the image but in under-canopy environment only the two crop row lines are in the field of view and frequent occlusions are encountered. We qualitatively compare the semantic keypoint predictions of CropFollow++ with a classical color-based segmentation method (index called ExG commonly used for plant segmentation [39]) as well as the state-of-the-art learning-based segmentation method [30] to illustrate that pixel-wise segmentation of the scene as an intermediate step is not useful for the task of under-canopy navigation (Figure 5. Though segmentation could be helpful in directly identifying the traversable triangle in a clean field during the early growth stages of the crop (in the first column in 5), it is not useful at all in the large majority of scenarios encountered in under-canopy navigation as illustrated in rows 2-4 in Figure 5.

2) Foundational model for navigation as baseline: We show visualization to illustrate the performance of a foundational model for navigation [50] trained with large indoor and outdoor navigation datasets in generating collision-free paths in this environment. The model was trainined in a variety of emboddiments to provide transferrability to new robotic platforms in a zero-shot manner. We tested the trained model without further finetuning and evaluated in an under-canopy dataset that corresponds to the same environment for which we developed CropFollow++.

As we demonstrated by Figure 6, the generated trajectories cannot correctly identify the crop row, generating trajectories that intersects the plants and would cause the robot's failure. Furthermore, NoMaD does not meet the necessary speed to be deployed on a high speed robot, since it ran at no more than 2 Hz on a laptop computer with a RTX3060, compared to CropFollow++ running on a Jetson AGX Orin at 30 Hz.

B. Quantitative evaluation of heading and distance ratio error

Our quantitative evaluation in Table I shows the L1 error in heading and distance ratio for the semantic keypoint representation proposed in CropFollow++ and the end-to-end perception architecture in CropFollow[49]. Note that both CropFollow++ and CropFollow were trained with the same training set and data augmentations with the only difference being the model architecture and the output representation. We also indicate the metrics for a trivial baseline that always predicts the median heading and distance ratio of the training set. We show the metrics for the entire validation dataset of 4873 images (in the first row of Table I, a subset of this dataset in which the data was obtained by manually driving in a zigzag manner to ensure large variations in the viewpoints in the dataset, and a subset representing data from fields with uneven terrain (roll $>$ 0.05 rad). The semantic keypoint representation in CropFollow++ significantly outperforms the direct prediction of heading and distance ratio from CropFollow in terms of median error in all cases. The difference is more significant in the challenging cases with the zigzag subset and uneven terrain subset. The difference is lesser in the case of mean error in the entire dataset because of the presence of outliers in the dataset. Since the keypoint heatmap predictions of CropFollow++ are spread out in case of outliers, calculating spatial softmax results in large values for heading and distance ratio compared to CropFollow. But note that during field operation, our heuristics based on the variance of the heatmaps and the geometry of the triangle enable rejection of such outlier predictions. Results from Table I show that CropFollow++ outputs more accurate estimates of row geometry compared to CropFollow.

VI. Field Experiment Results and Discussion

A. Comparison of CropFollow++ and CropFollow

We conducted field experiments comparing the navigation performance of our proposed system CropFollow++ with CropFollow using a Terrasentia robot shown in Figure 1. By CropFollow, we only refer to the end-to-end perception architecture that directly predicts heading and distance ratio from images. Otherwise, we ensure every other module in the navigation system remains the same for a fair comparison of the effect of perception representation on navigation performance. The same MPC controller that solves the constrained optimization problem to control linear and angular velocity was used in both cases. The objective of these comparative field experiments is to show the significantly improved navigation performance with the keypoint representation proposed here compared to end-to-end perception representation and so we used our compact development robot Terrasentia for these experiments.

Validation Dataset	Model	Heading Error (in °)		Distance Ratio Error	
		Mean	Median	Mean	Median
Entire Dataset (4873 images)	Baseline	5.17	4.02	0.085	0.071
	CropFollow	1.27	1.12	**0.042**	0.035
	CropFollow++	**1.2**	**0.66**	0.044	**0.026**
Zigzag trajectories (1576 images)	Baseline	7.41	6.87	0.107	0.098
	CropFollow	1.41	1.37	0.042	0.037
	CropFollow++	**0.72**	**0.45**	**0.026**	**0.018**
Uneven terrain (213 images)	Baseline	8.87	9.49	0.088	0.074
	CropFollow	1.9	1.85	0.041	0.035
	CropFollow++	**0.36**	**0.29**	**0.025**	**0.02**

TABLE I: **Offline comparison of CropFollow and CropFollow++ perception:** We report the L1 error in heading (in °) and distance ratio predictions for CropFollow and CropFollow++ representations with different validation splits. The trivial baseline always predicts the median heading and distance ratio from the training set. CropFollow++ significantly outperforms CropFollow in both predictions as seen from the median error.

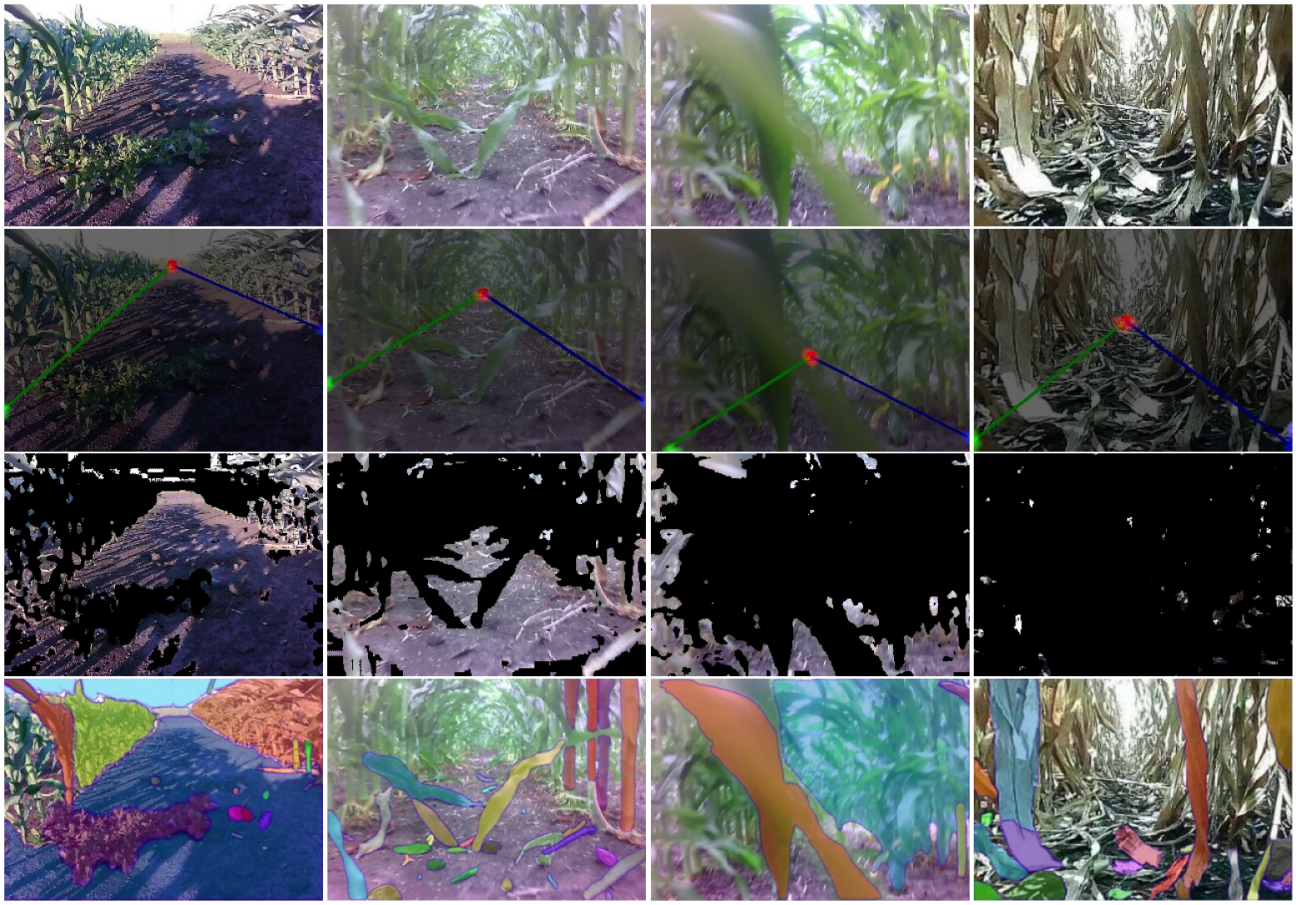


Fig. 5: Top row shows images from our validation dataset representing the different field conditions. The second row shows the output of our semantic keypoints prediction model. The third row shows the output of color segmentation of the ground using a classical method. Bottom row shows the output of Segment Anything Model [30]. It is clear from these examples that the traversable area represented by our semantic keypoints cannot be obtained by just segmenting the scene because of the challenges posed by occlusion and clutter.

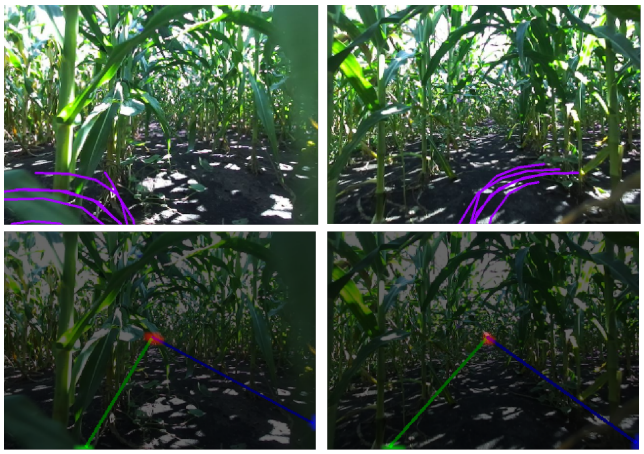


Fig. 6: We tested the foundational model NoMaD [50] with validation images collected in our under-canopy settings. The violet curves in the top row images represent the trajectories corresponding to the control commands generated by NoMaD. The bottom row shows the output of our semantic keypoints prediction model on the same images. NoMAD does not contain useful prior for under-canopy navigation.

The field experiments to compare CropFollow++ with CropFollow were performed in very late season growing conditions when the corn plants are brown in color and the leaves significantly occlude the robot's camera. Also, this was conducted in crop rows planted in an east-west direction along the edge of the farm because of which there were challenges due to lighting conditions as well. In addition, the terrain was slightly tilted from the furrows created during plowing posing another challenge. A monocular RGB camera publishing images at 30 Hz was used and the NVIDIA Jetson Orin computer onboard the Terrasentia robot was able to run both the semantic keypoint network as well as the CropFollow network at 30 Hz. The first experiment was conducted with an MPC reference speed of 0.85 m/s whereas the remaining 4 experiments were all conducted with a reference speed of 1 m/s.

1) Evaluation Methodology: We used the number of collisions and the time taken as the metrics for these experiments. During the experiment in autonomous mode, when the robot crashed into or went dangerously close to crashing into corn stalks, we disengaged autonomy and recorded it as an instance of collision. Note that during these scenarios, in the absence of the human intervening, the robot would remain stuck between the corn plants and so there is no ambiguity or subjective differences in this metric across humans or various repetitions of the experiment. Typically, cross-track error with respect to a GPS-defined reference is used as the error metric in outdoor navigation experiments. But GPS is not accurate under the plant canopy and so the number of collisions is used as the metric in under-canopy navigation literature [24, 49]. In addition, we also report the time taken by the robot in autonomous mode for CropFollow++ and CropFollow in each run in Table II. Since the MPC controller tracks the specified constant reference velocity, the time taken by the robot in autonomous mode is an indirect indicator of cross-track error. Note we calculated this carefully by ensuring the same start and end points in each run for both CropFollow++ and CropFollow and by removing the time corresponding to collision and subsequent manual recovery by the human.

2) Results: Table II reports the number of collisions, the maximum distance traveled without collision, and the time taken in autonomous mode in each run for CropFollow++ and CropFollow. In total, CropFollow++ had only 13 collisions while CropFollow had 33 collisions across all five runs. Also, individually in each run, CropFollow++ had either the same or a significantly fewer collisions indicating the robustness of CropFollow++ in these challenging conditions. In terms of maximum distance traveled before an collision, CropFollow++ reached a longer distance than CropFollow in each run. CropFollow++ took less time than CropFollow in all the runs except 1. But note that the reference velocity is lesser in run 1 (0.85 m/s) compared to other runs (1 m/s) so CropFollow is less affected by the lack of outlier detection and rejection. But at the velocity of 1 m/s (runs 2-5), CropFollow++ significantly outperforms CropFollow in all three metrics. All five runs in total were a distance of 1860 meters and CropFollow++ traveled 143 meters on average before a collision compared to 56 meters in the case of CropFollow. Figure 7 shows a cumulative histogram of the distance traveled before collision for all runs normalized by the length of the corresponding run. We use a normalized distance since one of the runs was of a different length compared to the other four. This histogram shows that CropFollow++ had fewer instances of very short distances and more instances of large distances covered before collision compared to CropFollow.

Both CropFollow++ and CropFollow were trained on the same dataset. But the modular perception architecture CropFollow++ and the properties of the keypoint heatmap representation enable the improved robustness of CropFollow++ compared to CropFollow. The confidence check based on vanishing point heatmap variance enables us to filter occlusions. The top row in Fig 8 shows examples of outliers such as occlusion and end-of-row scenario detected by the large variance in vanishing point heatmap (indicated by the red color spread across the image rather than concentrated at a point). In addition, by using the prior knowledge about the fixed gap between the rows of crops and the known calibration of the camera, we are able to create a heuristic that filters outlier predictions in intercept keypoints that are outside the bounds defined by the heuristic. The middle row in Fig 8 shows examples of outliers in the prediction of intercept keypoints that are filtered by this heuristic check. On average across the five runs, 67.77% of the predictions from the semantic keypoint model had higher confidence than the threshold and 50.77% of the predictions are within the bounds of the intercept heuristic. Note that CropFollow++ does not compute the heading angle when the vanishing point confidence threshold is not met. Similarly, when the heuristic bound criterion is not reached

	Length of experiment [m]	Number of collisions		Max distance without collisions [m]		Total time taken in autonomous mode [s]	
		CropFollow++	CropFollow	**CropFollow++**	CropFollow	**CropFollow++**	CropFollow
Run 1	420	2	2	**412**	310	500.1	**494.2**
Run 2	420	**5**	8	**262**	115	**415.0**	417.8
Run 3	420	**2**	10	**366**	165	**412.6**	416.2
Run 4	180	**1**	7	**170**	74	**193.6**	199.0
Run 5	420	**3**	6	**390**	260	**413.0**	422.2

TABLE II: We conducted 5 runs of field tests and report here the total number of collisions, the maximum distance traveled without collision, and the time taken in autonomous modein each run for both CropFollow and CropFollow++. Note that CropFollow++ has significantly fewer collisions, high maximum distance without collisions, and less time taken in autonomous mode than CropFollow.

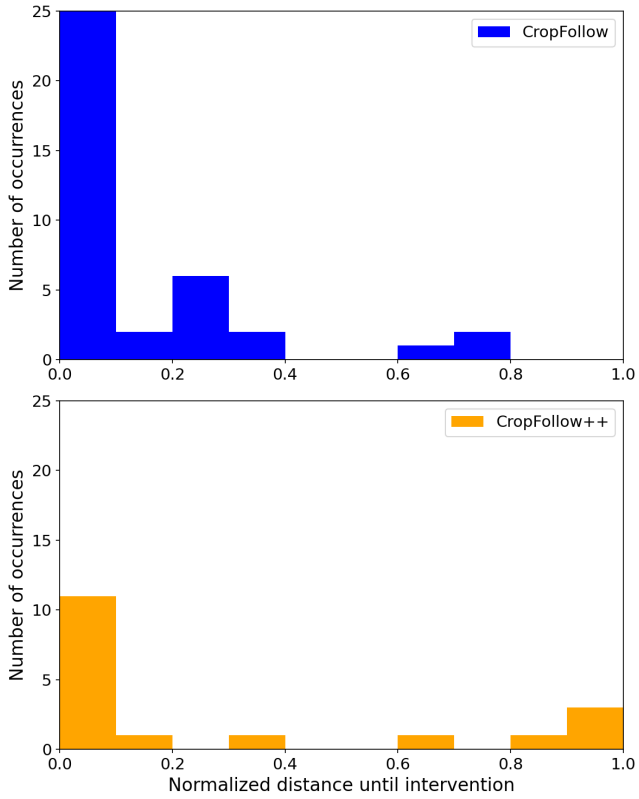


Fig. 7: We report a cumulative histogram of the distance traveled before collision for all runs normalized by the length of the runs. CropFollow++ had fewer instances of very short distances and more instances of large distances covered before collision compared to CropFollow.

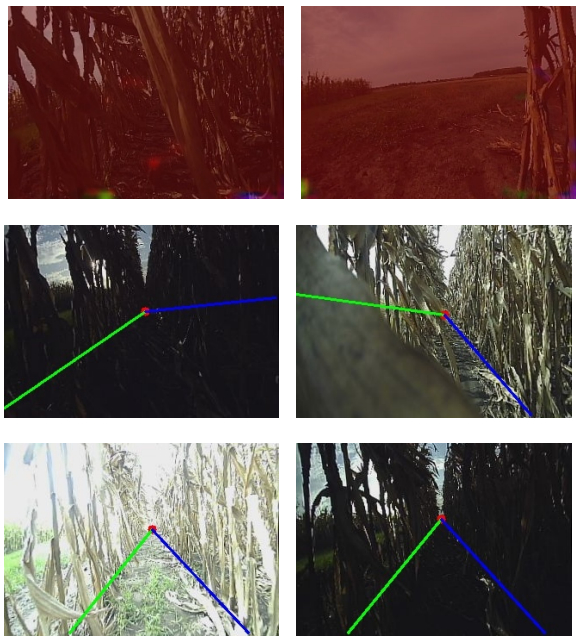


Fig. 8: The top row shows the anomalous scenarios such as occlusion and out-of-row detected by large variance in vanishing point heatmap. The middle row shows cases with outlier predictions for intercepts detected by our triangle area-based heuristic check. The bottom row shows accurate model predictions in challenging scenarios.

for both intercept keypoints, CropFollow++ does not compute the lateral distance. When both heading and lateral distance states are not updated, the controller continues to apply the previous control action. Therefore, though these two checks provide robustness to occlusions and outlier predictions from the model momentarily, if those scenarios persist continuously for several frames, failing to recompute an updated control action causes the robot to crash. CropFollow only outputs point estimates with no measure of uncertainty. State-of-the-art methods for uncertainty estimation of neural networks in regression problems require multiple forward passes through the network and hence are not feasible during real-time operation. The confidence check and heuristic check together can be used for unsupervised online domain adaptation using pseudo-labeling methods.

Another reason for CropFollow++ 's improved navigation performance could be attributed to the modular architecture enabling to use roll estimates from IMU while calculating the heading and lateral distance. Because of the end-to-end perception architecture, CropFollow does not generalize well

Fig. 9: CCR navigating between the crop rows. Note the minimal space available for the robot between the rows.

Fig. 10: CCR Camera Views. CCR has three front cameras and one rear camera which enables robustness to occlusions and recovery from crashes.

Fig. 11: Various field conditions in which the cover crop robots have been tested. Note the variations in crop and soil appearance, terrain flatness, and lighting conditions.

to terrains with roll variation if not seen during training.

B. Demonstration of CropFollow++ on CCR

We also deployed our proposed CropFollow++ on multiple under-canopy cover crop planting robots over large distances and discuss the observed performance, common failure modes, and lessons learned.

1) EarthSense CoverCrop Robot (CCR) Platform: The EarthSense CCR Platform is designed for autonomous planting of cover crops in crop fields. It is equipped with four cameras. Three of the cameras are in the front, to provide redundancy

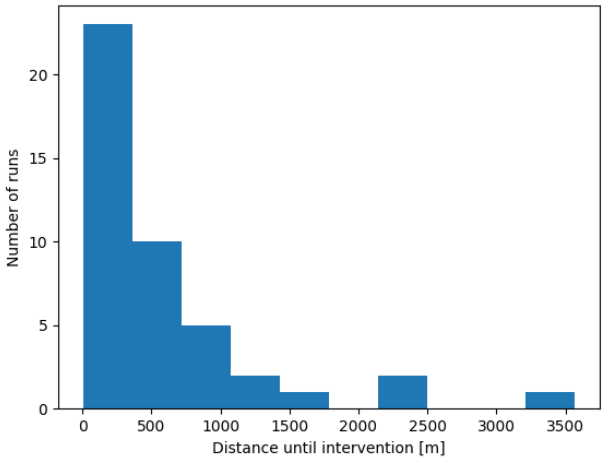


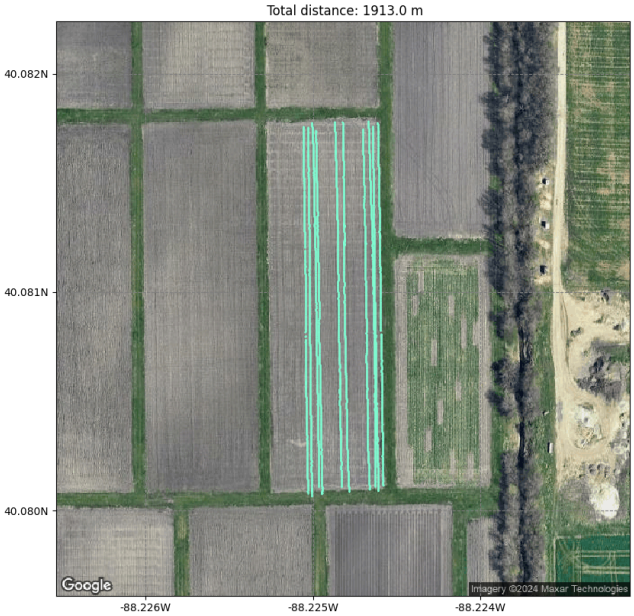
Fig. 12: We report the histogram of the distance traveled before an irrecoverable collision across all the autonomous runs from three CCRs. Though the majority of the runs are less than 250 m, we show three instances of autonomous runs with more than 2000 m.

against occlusion. One camera is in the rear to provide capacity for crash recovery. A snapshot of in-field images from the front left, front center, front right, and rear cameras is shown in Figure 10. The CCR is 0.45 m wide, leaving only 0.15 m of space on each side of the robot in a standard 0.75 m corn row. Note the gap between the robot and the plants is lesser in CCR compared to Terrasentia. The CCR robot has an overall power capacity of 600Wh from two batteries. The power usage of the robot while operating at 0.9 m/s is 300W out of which only around 6.5W is attributable to neural network inference. The vast majority of power usage is attributable to the wheels which are responsible for carrying a heavy load (up to 100kg when fully loaded with cover crop seeds).

2) CCR Autonomy Protocol: Multiple CCRs were configured to run CropFollow++. All four cameras were calibrated using Kalibr [14]. The front three cameras were calibrated as a multi-sensor system, with estimates of relative poses between the sensors provided by the calibration procedure. These pose transforms were used in sequence with the robot pose to provide more precise estimates of heading and side distance per camera.

To achieve long-range autonomy with minimal human interventions, we also configured the CCRs to dynamically switch between a forward row follow mode and a reverse crash recovery mode [19]. We implemented a proprietary crash detection method based on odometry data. When a collision was detected by this method, the robot would stop, and then execute the autonomus crash recovery procedure by running CropFollow++ on the rear camera for a fixed time and then switch back to the front camera . During row-follow, the CCR ran keypoint detection, heading estimation, and side distance estimation independently on the three front cameras. These outputs were fused, and then fed into the waypoint generation and MPC controller. During crash recovery, the only difference

(a) Farm 1



(b) Farm 2

Fig. 13: Trajectories of CropFollow++ deployed in three CCR in two different farms.

was that the CCR ran keypoint detection, heading estimation, and side distance estimation only on the rear camera, and the waypoint generator provided a backward target path. Fusing the predictions from multiple cameras improved the robustness of the system. Note that in CCR deployment, heuristics that check the confidence of the vanishing point heatmap and the area of the triangle formed by the crop row lines were not used.

3) CCR Field Testing: We did extensive autonomy runs on three CCRs, many of which were done while actively carrying a heavy covercrop payload and planting covercrop. We ran our experiments at 0.9 m/s for row follow and 0.4 m/s for crash recovery. Experiments were run in a variety of plant growth stages and weather conditions, as shown in Figure 11. The majority of corn rows we tested on were of length 200 m or less, which the robot almost always completed autonomously. At the end of each row, the robot was manually turned to enter another lane, where it then resumed autonomous navigation.

4) CCR Autonomy Results: Since the CCR robot used autonomous collision detection and recovery systems, we report two metrics relevant to that namely the number of collisions and the number of irrecoverable collisions. Note that only in case of irrecoverable collisions, human intervention is needed whereas in other cases robot recovers autonomously from collisions. Due to the limited length of corn rows, a full estimate of distance between irrecoverable collisions is provided by adding together the lengths of sequential autonomous runs. By this metric, we achieved an average of 767 meters between irrecoverable collisions across the three robots we evaluated, and a best-case scenario of 3571 meters before an irrecoverable collision (Fig 12). These statistics were generated across 25315 meters of autonomy. In total, there were 109 collisions during CCR deployment out of which the robot successfully recovered autonomously in 75 cases. 34 collisions were irrecoverable and needed human intervention.

5) Analyzing Failure Modes: We analyzed the interventions and established a classification schema for causes of autonomy failures. This classification schema consists of the following failure modes.

- **Vision Keypoint Error** - These failures were caused by errors in our autonomy algorithm during relatively normal conditions.
- **Physical Robot Failure** - These failures were caused by physical failures in the robot, such as motor failure or excessive mud accumulation on the wheels.
- **Corn gap** - These failures were caused by sudden long gaps in the corn on one or both sides. This typically confused the probability map for at least one keypoint, leading the algorithm to be confused as to which row was the true left or right row.
- **Bad start** - These failures were caused by a severely suboptimal initial pose of the robot relative to the corn rows. The pose of the robot was too severe to be corrected even with perfect perception.
- **Weeds and Occlusion** - These failures were caused by objects such as weeds physically blocking the robot, or occluding the cameras. Occlusion would make it impossible to directly perceive one or more of the keypoints, confusing the network. In addition to causing occlusion, the weeds would also sometimes physically impede the robot.
- **Bumps** - These failures were caused by bumpy terrain and ground obstacles that significantly jolted the robot. In these cases, typically the heading of the robot spiked in one direction, making it difficult for the robot to recover.
- **Planting error** - These failures were caused by plants

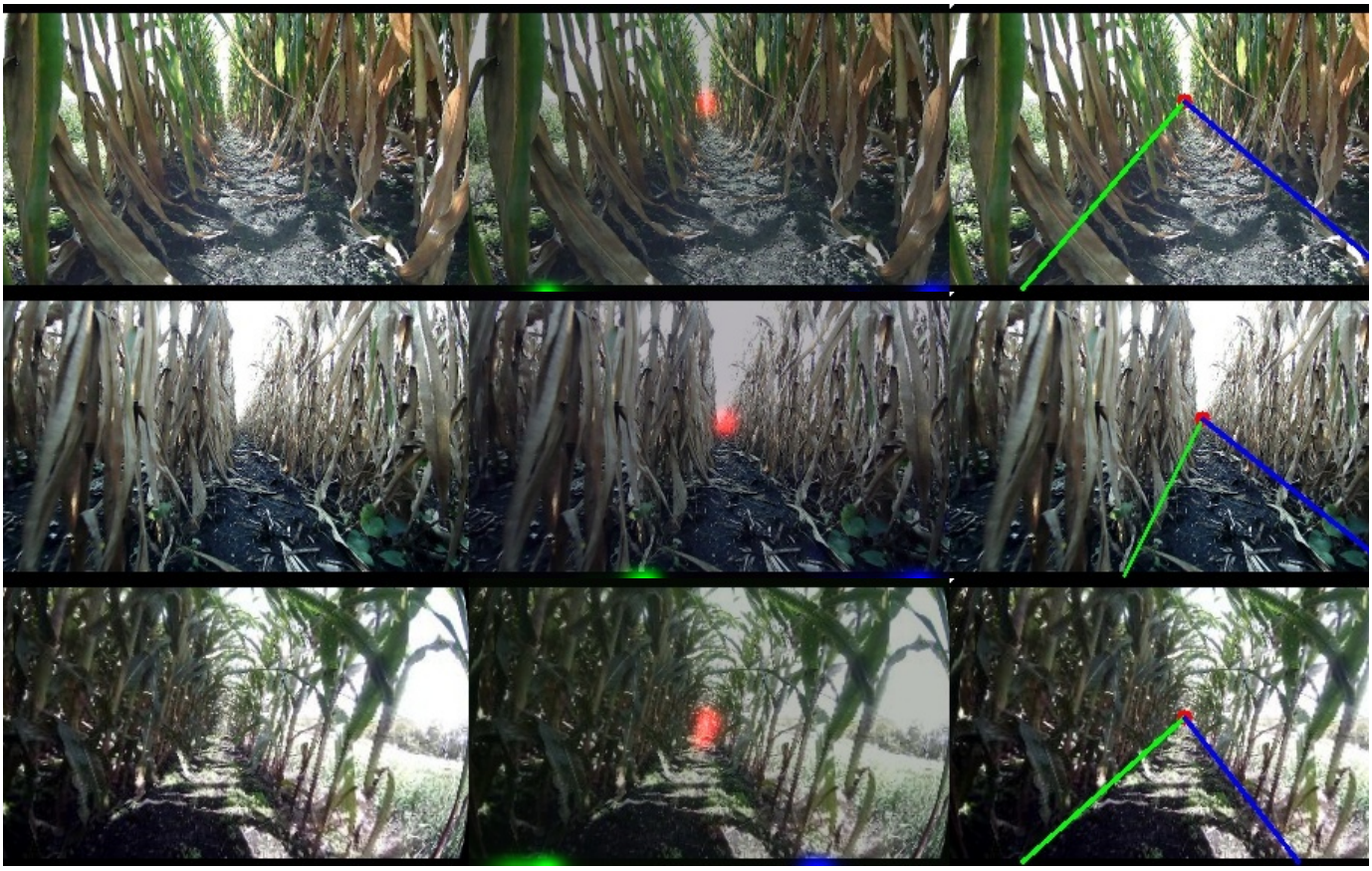


Fig. 14: **Keypoint Detection in Action:** Here we show three randomly sampled images from CCR deployment with keypoint detections. **Left:** Raw images from the CCR front camera. **Center:** Visualizations of the heatmaps for the vanishing point (red), the left point (green), and the right point (blue). **Right:** Visualizations of the keypoint outputs and the extracted vanishing lines, computed as the argmax of each heatmap.

that were erroneously planted in the middle of the gap, instead of in the plant row lines. This is unusual and unexpected behavior in corn. Planting errors would both confuse the vision keypoint model and act as a physical barrier that the robot struggled to overpower.

Examples of some of these failures are shown in Figure 16. A summary of our failures according to this schema is shown in Figure 15

6) Key Takeaways from CCR Deployment:

- CropFollow++ shows promise as a key part of solving full-field under-canopy autonomy in agriculture. Our experiments show that row follow is very nearly solved in normal situations, and that achieving full-field autonomy would require improving robot hardware and enabling CropFollow++ to handle anomalies encountered due to domain shift such as gaps in the crop rows, presence of weeds, occlusion, and planting errors. Future work could focus on developing a semi-supervised offline learning and self-supervised online learning system to tackle domain shift.
- Furthermore, the vision keypoint method provided interpretability of success and failures. Visualizations of the keypoint probability maps are key to deducing how and why various anomalies can cause interventions. We have observed qualitatively that the most reliable vision keypoint estimates tend to come from the cameras closest to the center of the row since they are more represented in the training dataset than other viewpoints. This suggests that more images from other viewpoints are needed in

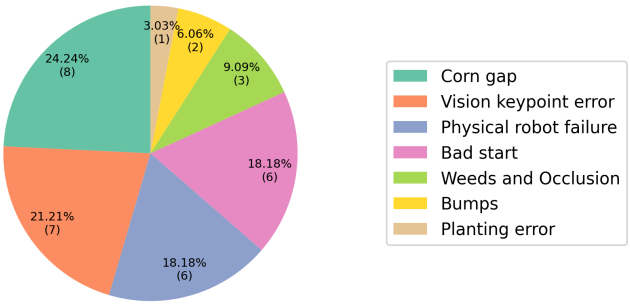


Fig. 15: We show the distribution of various causes of CCR autonomy interventions.

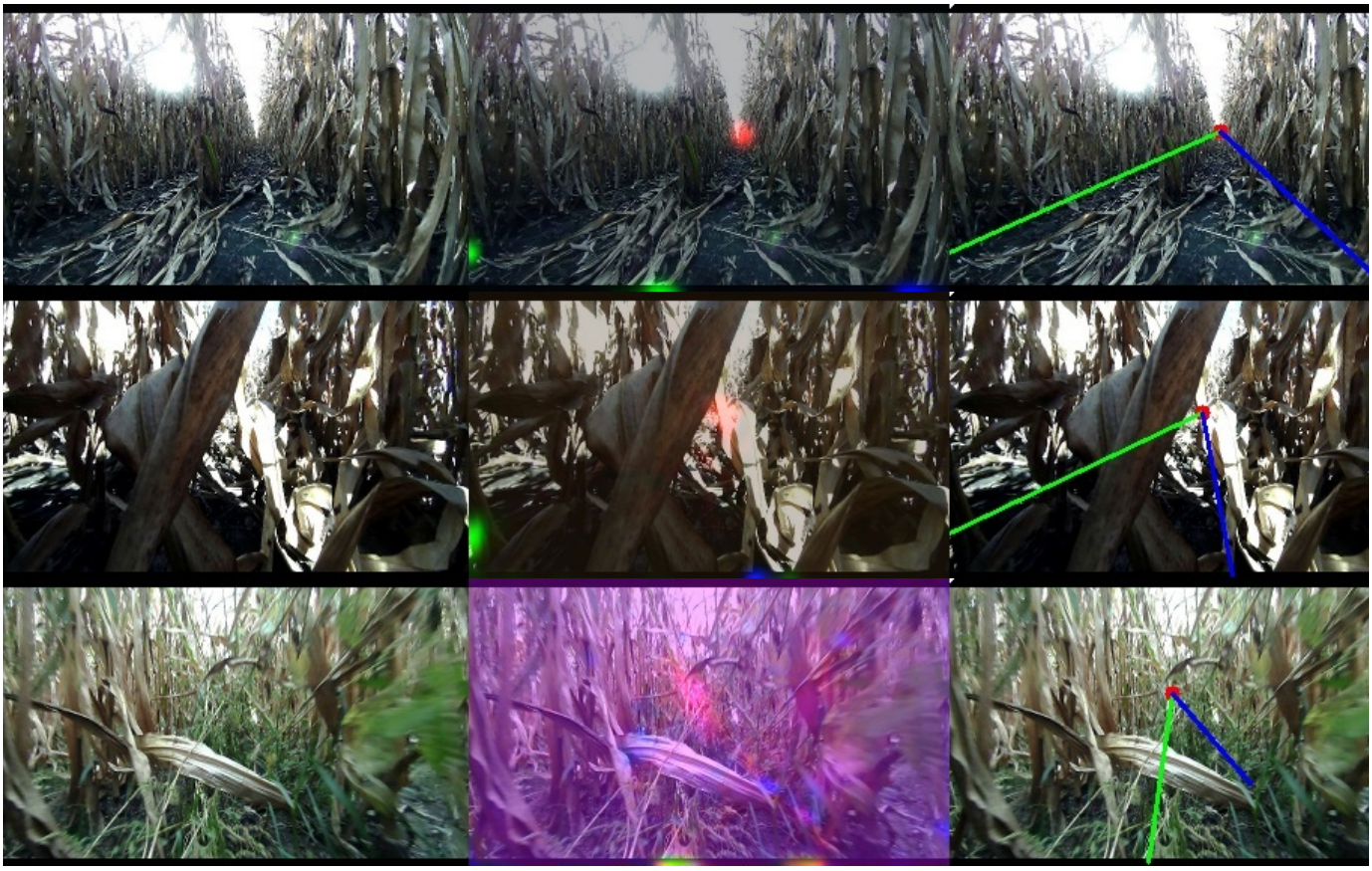


Fig. 16: **Examples of challenging environmental conditions**. **Top:** An example of a corn gap that led to an autonomy intervention. The keypoint network is confused by the gap and mistakes the neighboring row on the left for the current left row. **Center:** An example of an occlusion event that led to an autonomy intervention. None of the true keypoints are visible in the image. **Bottom:** An example of weeds covering the soil which led to an autonomy intervention. The keypoint network is unable to determine the keypoints.

the training dataset to improve the robustness of CropFollow++.

- Currently, CropFollow++ only considers the argmax values from the heatmaps as keypoint locations to calculate heading and distance. Other sophisticated heuristics that look at all the clusters in the heatmap would help improve performance in cases such as in the top row of Fig 16. Also, currently CCR pipeline fused predictions from multiple cameras with a simple average. Incorporating the heuristic checks discussed in III while fusing the predictions could also be helpful. In addition, fusing temporal information from multiple inexpensive sensors such as IMU, wheel encoders and magnetometer using estimation techniques could help improve the heading and distance estimation and thereby the robustness of the system.
- We found that correcting for the robot's roll was among the most important refinements we added. The robot's estimation of side distance is sensitive to robot angles, particularly the roll parameter. Adding in this real-time correction was critical to securing performance on rows with non-level ground.
- We also found that precise calibration was essential to successful row follow, particularly the estimation of the principal point. Our cameras were cheap and did not come with pre-computed calibration, requiring us to find a separate solution. Any error in the principal point would appear as a skew in heading estimation and could lead to frequent crashes.
- The results we achieved for CCR autonomy could not have been achieved without the implementation of crash detection and recovery modules. Many times we would crash due to an anomaly or vision keypoint error, and then recover and resume without a problem, having success the second time we encountered the anomaly. Row follow paired with crash detection and recovery will be a key part of solving full-field autonomy.

VII. Conclusion

We presented an empiricially robust vision-based under-canopy navigation system with semantic keypoints called CropFollow++ that is more modular and interpretable than

prior work CropFollow. In our field experiments in challenging field conditions, CropFollow++ significantly outperformed CropFollow in terms of the number of collisions (13 *vs.* 33). We also deployed CropFollow++ on multiple cover crop planting robots over 25 km in various field conditions. These experiments and our follow-up analysis have provided several lessons for future work.

Acknowledgments

This work was supported in part by NSF STTR #1951250, NSF NRI 2.0 NIFA #2021-67021-33449, AI-FARMS #1024178, NSF-USDA COALESCE #2021-67021-34418, USDA grants iCOVER(#NR233A750004G066) and iFARM(#2022-77038-37306). We would like to thank Naveen Uppalapati's help in coordinating the collaboration on under-canopy cover crop planting through I-FARM.

References

[1] Diego Aghi, Vittorio Mazzia, and Marcello Chiaberge. Local motion planner for autonomous navigation in vineyards with a rgb-d camera-based algorithm and deep learning synergy. *Machines*, 8(2):27, 2020.

[2] Björn Åstrand and Albert-Jan Baerveldt. A vision based row-following system for agricultural field machinery. *Mechatronics*, 15(2):251–269, 2005.

[3] Yuhao Bai, Baohua Zhang, Naimin Xu, Jun Zhou, Jiayou Shi, and Zhihua Diao. Vision-based navigation and guidance for agricultural autonomous vehicles and robots: A review. *Computers and Electronics in Agriculture*, 205: 107584, 2023.

[4] Marianne Bakken, Richard JD Moore, and Pål From. End-to-end learning for autonomous crop row-following. *IFAC-PapersOnLine*, 52(30):102–107, 2019.

[5] David Ball, Ben Upcroft, Gordon Wyeth, Peter Corke, Andrew English, Patrick Ross, Tim Patten, Robert Fitch, Salah Sukkarieh, and Andrew Bate. Vision-based obstacle detection and navigation for an agricultural robot. *Journal of field robotics*, 33(8):1107–1130, 2016.

[6] Gustavo BP Barbosa, Eduardo C Da Silva, and Antonio C Leite. Robust image-based visual servoing for autonomous row crop following with wheeled mobile robots. In *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, pages 1047–1053. IEEE, 2021.

[7] Marcel Bergerman, Silvio M Maeta, Ji Zhang, Gustavo M Freitas, Bradley Hamner, Sanjiv Singh, and George Kantor. Robot farmers: Autonomous orchard vehicles help tree fruit production. *IEEE Robotics & Automation Magazine*, 22(1):54–63, 2015.

[8] Luciano Cantelli, Filippo Bonaccorso, Domenico Longo, Carmelo Donato Melita, Giampaolo Schillaci, and Giovanni Muscato. A small versatile electrical robot for autonomous spraying in agriculture. *AgriEngineering*, 1 (3):391–402, 2019.

[9] Simone Cerrato, Vittorio Mazzia, Francesco Salvetti, Mauro Martini, Simone Angarano, Alessandro Navone, and Marcello Chiaberge. A deep learning driven algorithmic pipeline for autonomous navigation in row-based crops. *arXiv preprint arXiv:2112.03816*, 2021.

[10] Jiqing Chen, Hu Qiang, Jiahua Wu, Guanwen Xu, Zhikui Wang, and Xu Liu. Extracting the navigation path of a tomato-cucumber greenhouse robot based on a median point hough transform. *Computers and Electronics in Agriculture*, 174:105472, 2020.

[11] Shoubin Chen, Baoding Zhou, Changhui Jiang, Weixing Xue, and Qingquan Li. A lidar/visual slam backend with loop closure detection and graph optimization. *Remote Sensing*, 13(14):2720, 2021.

[12] Girish Chowdhary, Mattia Gazzola, Girish Krishnan, Chinmay Soman, and Sarah Lovell. Soft robotics as an enabling technology for agroforestry practice and research. *Sustainability*, 11(23):6751, 2019.

[13] Matteo Corno, Sara Furioli, Paolo Cesana, and Sergio M Savaresi. Adaptive ultrasound-based tractor localization for semi-autonomous vineyard operations. *Agronomy*, 11 (2):287, 2021.

[14] Paul Furgale, Joern Rehder, and Roland Siegwart. Unified temporal and spatial calibration for multi-sensor systems. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1280–1286. IEEE, 2013.

[15] Jingyao Gai, Lirong Xiang, and Lie Tang. Using a depth camera for crop row detection and mapping for under-canopy navigation of agricultural robotic vehicle. *Computers and Electronics in Agriculture*, 188:106301, 2021.

[16] Guandong Gao, Ke Xiao, and YuChen Jia. A spraying path planning algorithm based on colour-depth fusion segmentation in peach orchards. *Computers and electronics in agriculture*, 173:105412, 2020.

[17] Iván D García-Santillán, Martín Montalvo, José M Guerrero, and Gonzalo Pajares. Automatic detection of curved and straight crop rows from images in maize fields. *Biosystems Engineering*, 156:61–79, 2017.

[18] Mateus V Gasparino, Arun N Sivakumar, Yixiao Liu, Andres EB Velasquez, Vitor AH Higuti, John Rogers, Huy Tran, and Girish Chowdhary. Wayfast: Navigation with predictive traversability in the field. *IEEE Robotics and Automation Letters*, 7(4):10651–10658, 2022.

[19] Mateus V Gasparino, Vitor AH Higuti, Arun N Sivakumar, Andres EB Velasquez, Marcelo Becker, and Girish Chowdhary. Cropnav: a framework for autonomous navigation in real farms. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11824–11830. IEEE, 2023.

[20] Mateus Valverde Gasparino, Arun Narenthiran Sivakumar, and Girish Chowdhary. Wayfaster: a self-supervised traversability prediction for increased navigation awareness, 2024.

[21] Javier Gimenez, Sebastian Sansoni, Santiago Tosetti, Flavio Capraro, and Ricardo Carelli. Trunk detection in tree crops using rgb-d images for structure-based icm-

slam. *Computers and Electronics in Agriculture*, 199: 107099, 2022.

[22] Yili Gu, Zhiqiang Li, Zhen Zhang, Jun Li, and Liqing Chen. Path tracking control of field information-collecting robot based on improved convolutional neural network algorithm. *Sensors*, 20(3):797, 2020.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[24] Vitor AH Higuti, Andres EB Velasquez, Daniel Varela Magalhaes, Marcelo Becker, and Girish Chowdhary. Under canopy light detection and ranging-based autonomous navigation. *Journal of Field Robotics*, 36(3):547–567, 2019.

[25] Kosuke Inoue, Yutaka Kaizu, Sho Igarashi, and Kenji Imou. The development of autonomous navigation and obstacle avoidance for a robotic mower using machine vision technique. *IFAC-PapersOnLine*, 52(30):173–177, 2019.

[26] Guoquan Jiang, Zhiheng Wang, and Hongmin Liu. Automatic detection of crop rows based on multi-rois. *Expert systems with applications*, 42(5):2429–2441, 2015.

[27] Erkan Kayacan, Sierra N Young, Joshua M Peschel, and Girish Chowdhary. High-precision control of tracked field robots in the presence of unknown traction coefficients. *Journal of Field Robotics*, 35(7):1050–1062, 2018.

[28] Kitae Kim, Aarya Deb, and David J Cappelleri. P-agbot: In-row & under-canopy agricultural robot for monitoring and physical sampling. *IEEE Robotics and Automation Letters*, 7(3):7942–7949, 2022.

[29] Wan-Soo Kim, Dae-Hyun Lee, Taehyeong Kim, Gookhwan Kim, Hyunggun Kim, Taeyong Sim, and Yong-Joo Kim. One-shot classification-based tilled soil region segmentation for boundary guidance in autonomous tillage. *Computers and Electronics in Agriculture*, 189:106371, 2021.

[30] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[31] Johannes Kneip, Patrick Fleischmann, and Karsten Berns. Crop edge detection based on stereo vision. *Robotics and Autonomous Systems*, 123:103323, 2020.

[32] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[33] Yong Li, Zhiqiang Guo, Feng Shuang, Man Zhang, and Xiuhua Li. Key technologies of machine vision for weeding robots: A review and benchmark. *Computers and Electronics in Agriculture*, 196:106880, 2022.

[34] Flavio BP Malavazi, Remy Guyonneau, Jean-Baptiste Fasquel, Sebastien Lagrange, and Franck Mercier. Lidar-only based navigation algorithm for an autonomous agricultural robot. *Computers and electronics in agriculture*, 154:71–79, 2018.

[35] Wyatt McAllister, Denis Osipychev, Girish Chowdhary, and Adam Davis. Multi-agent planning for coordinated robotic weed killing. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7955–7960. IEEE, 2018.

[36] Wyatt McAllister, Denis Osipychev, Adam Davis, and Girish Chowdhary. Agbots: Weeding a field with a team of autonomous robots. *Computers and Electronics in Agriculture*, 163:104827, 2019.

[37] Wyatt McAllister, Joshua Whitman, Allan Axelrod, Joshua Varghese, Girish Chowdhary, and Adam Davis. Agbots 2.0: Weeding denser fields with fewer robots. In *Robotics: Science and Systems*, volume 5, pages 1–9, 2020.

[38] Wyatt McAllister, Joshua Whitman, Joshua Varghese, Adam Davis, and Girish Chowdhary. Agbots 3.0: Adaptive weed growth prediction for mechanical weeding agbots. *IEEE Transactions on Robotics*, 38(1):556–568, 2021.

[39] George E Meyer and Joao Camargo Neto. Verification of color vegetation indices for automated crop imaging applications. *Computers and electronics in agriculture*, 63(2):282–293, 2008.

[40] Andres Milioto, Philipp Lottes, and Cyrill Stachniss. Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in cnns. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 2229–2235. IEEE, 2018.

[41] Anjana K Nellithimaru and George A Kantor. Rols: Robust object-level slam for grape counting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 0–0, 2019.

[42] Luiz FP Oliveira, António P Moreira, and Manuel F Silva. Advances in agriculture robotics: A state-of-the-art review and challenges ahead. *Robotics*, 10(2):52, 2021.

[43] Samwel Opiyo, Cedric Okinda, Jun Zhou, Emmy Mwangi, and Nelson Makange. Medial axis-based machine-vision system for orchard robot navigation. *Computers and Electronics in Agriculture*, 185:106153, 2021.

[44] Zachary Pezzementi, Trenton Tabor, Peiyun Hu, Jonathan K Chang, Deva Ramanan, Carl Wellington, Benzun P Wisely Babu, and Herman Herman. Comparing apples and oranges: Off-road pedestrian detection on the national robotics engineering center agricultural person-detection dataset. *Journal of Field Robotics*, 35(4):545–563, 2018.

[45] Redmond R Shamshiri, Cornelia Weltzien, Ibrahim A Hameed, Ian J Yule, Tony E Grift, Siva K Balasundram, Lenka Pitonakova, Desa Ahmad, and Girish Chowdhary. Research and development in agricultural robotics: A

perspective of digital farming. 2018.

[46] Josiah Radcliffe, Julie Cox, and Duke M Bulanon. Machine vision for orchard navigation. *Computers in Industry*, 98:165–171, 2018.

[47] Giulio Reina, Annalisa Milella, Raphaël Rouveure, Michael Nielsen, Rainer Worst, and Morten R Blas. Ambient awareness for agricultural robotic vehicles. *biosystems engineering*, 146:114–132, 2016.

[48] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[49] Arun Narenthiran Sivakumar, Sahil Modi, Mateus Valverde Gasparino, Che Ellis, Andres Baquero Velasquez, Girish Chowdhary, and Saurabh Gupta. Learned visual navigation for under-canopy agricultural robots. In *Robotics: Science and Systems*, 2021.

[50] Ajay Sridhar, Dhruv Shah, Catherine Glossop, and Sergey Levine. Nomad: Goal masked diffusion policies for navigation and exploration. *arXiv preprint arXiv:2310.07896*, 2023.

[51] Adam Stager, Herbert G Tanner, and Erin Sparks. Design and construction of unmanned ground vehicles for sub-canopy plant phenotyping. In *High-Throughput Plant Phenotyping: Methods and Protocols*, pages 191–211. Springer, 2022.

[52] Nikolaos Stefas, Haluk Bayram, and Volkan Isler. Vision-based monitoring of orchards with uavs. *Computers and Electronics in Agriculture*, 163:104814, 2019.

[53] Vijay Subramanian, Thomas F Burks, and AA Arroyo. Development of machine vision and laser radar based autonomous vehicle guidance systems for citrus grove navigation. *Computers and electronics in agriculture*, 53 (2):130–143, 2006.

[54] Naveen Kumar Uppalapati, Benjamin Walt, Aaron J Havens, Armeen Mahdian, Girish Chowdhary, and Girish Krishnan. A berry picking robot with a hybrid soft-rigid arm: Design and task space control. In *Robotics: Science and Systems*, page 95, 2020.

[55] Mel Vecerik, Jean-Baptiste Regli, Oleg Sushkov, David Barker, Rugile Pevceviciute, Thomas Rothörl, Raia Hadsell, Lourdes Agapito, and Jonathan Scholz. S3k: Self-supervised semantic keypoints for robotic manipulation via multi-view consistency. In *Conference on Robot Learning*, pages 449–460. PMLR, 2021.

[56] Andres Eduardo Baquero Velasquez, Vitor Akihiro Hisano Higuti, Mateus Valverde Gasparino, Arun Narenthiran Sivakumar, Marcelo Becker, and Girish Chowdhary. Multi-sensor fusion based robust row following for compact agricultural robots. *arXiv preprint arXiv:2106.15029*, 2021.

[57] S Vougioukas. Annual review of control, robotics, and autonomous systems. *Agricultural robotics*, 2(1):365–392, 2019.

[58] Tianhai Wang, Bin Chen, Zhenqian Zhang, Han Li, and Man Zhang. Applications of machine vision in agricultural robot navigation: A review. *Computers and Electronics in Agriculture*, 198:107085, 2022.

[59] Wera Winterhalter, Freya Fleckenstein, Christian Dornhege, and Wolfram Burgard. Localization for precision navigation in agricultural fields—beyond crop row following. *Journal of Field Robotics*, 38(3):429–451, 2021.

[60] Marco FS Xaud, Antonio C Leite, and Pål J From. Thermal image based navigation system for skid-steering mobile robots in sugarcane crops. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 1808–1814. IEEE, 2019.

[61] Hongzhen Xu, Shichao Li, Yuhan Ji, Ruyue Cao, and Man Zhang. Dynamic obstacle detection based on panoramic vision in the moving state of agricultural machineries. *Computers and Electronics in Agriculture*, 184:106104, 2021.

[62] Rui Xu and Changying Li. A review of high-throughput field phenotyping systems: Focusing on ground robots. *Plant Phenomics*, 2022.

[63] Jinlin Xue, Lei Zhang, and Tony E Grift. Variable field-of-view machine vision based row guidance of an agricultural robot. *Computers and Electronics in Agriculture*, 84:85–91, 2012.

[64] Zhiqiang Zhai, Zhongxiang Zhu, Yuefeng Du, Zhenghe Song, and Enrong Mao. Multi-crop-row detection algorithm based on binocular vision. *Biosystems engineering*, 150:89–103, 2016.

[65] Chi Zhang and Noboru Noguchi. Development of a multi-robot tractor system for agriculture field work. *Computers and Electronics in Agriculture*, 142:79–90, 2017.

[66] Qin Zhang, John F Reid, and Noboru Noguchi. Agricultural vehicle navigation using multiple guidance sensors. In *Proceedings of the international conference on field and service robotics*, pages 293–298. Citeseer, 1999.

[67] Quan Zhang, Qijin Chen, Zhengpeng Xu, Tisheng Zhang, and Xiaoji Niu. Evaluating the navigation performance of multi-information integration based on low-end inertial sensors for precision agriculture. *Precision Agriculture*, 22(3):627–646, 2021.

[68] Wei Zhao, Xuan Wang, Bozhao Qi, and Troy Runge. Ground-level mapping and navigating for agriculture based on iot and computer vision. *IEEE Access*, 8: 221975–221985, 2020.

[69] Yifan Zhao, Hashim Sharif, Peter Pao-Huang, Vatsin Shah, Arun Narenthiran Sivakumar, Mateus Valverde Gasparino, Abdulrahman Mahmoud, Nathan Zhao, Sarita Adve, Girish Chowdhary, et al. Approxcaliper: A programmable framework for application-aware neural network optimization. *Proceedings of Machine Learning and Systems*, 5, 2023.