Abstract—This paper presents embedded high precision control and corn stand counting algorithms for a low-cost, ultra-compact 3D printed and autonomous field robot for agricultural operations. Currently, plant traits, such as emergence rate, biomass, vigor and stand counting are measured manually. This is highly labor intensive and prone to errors. The robot, termed TerraSentia, is designed to automate the measurement of plant traits for efficient phenotyping as an alternative to manual measurements. In this paper, we formulate a Nonlinear Moving Horizon Estimator (NMHE) that identifies key terrain parameters using onboard robot sensors and a learning-based Nonlinear Model Predictive Control (NMPC) that ensures high precision path tracking in the presence of unknown wheel-terrain interaction. Moreover, we develop a machine vision algorithm to enable TerraSentia to count corn stands by driving through the fields autonomously. We present results of an extensive field-test study that shows that (i) the robot can track paths precisely with less than 5 cm error so that the robot is less likely to damage plants, and (ii) the machine vision algorithm is robust against interferences from leaves and weeds, and the system has been verified in corn fields at the growth stage of V4, V6, VT, R2, and R6 from five different locations. The robot predictions agree well with the ground truth with \( \text{count}_{\text{robot}} = 0.96 \times \text{count}_{\text{human}} + 0.85 \) and correlation coefficient \( R = 0.96 \).

I. INTRODUCTION

Phenotypic traits are measured manually by field technicians in the field to determine physical differences between plant genotype and the influence of environmental conditions. Frequent and accurate measurement of these phenotypic traits can be utilized to breed improved crops that have more nutritional value, yield, and resilience to weather anomalies. However, manual phenotyping is expensive due to its labor-intensive nature, and prone to human measurement errors. This has led to the so-called phenotyping bottleneck preventing rapid advances in plant breeding [1], [2]. On the other hand, these crop characteristics can also be measured by exteroceptive sensors such as hyperspectral cameras, stereo cameras, thermal cameras, lidar, etc. These sensors can be integrated on a mobile robotic scanning and data processing system. Such robotic phenotyping systems have a tremendous potential to relieve the phenotyping bottleneck by significantly reducing labor cost and eliminating errors due to human-subjective information.

Unmanned robots have potential to capture extremely detailed data of plants without the need of an expert operator. A four-wheeled autonomous rover with variable chassis clearance and width is equipped with 3D time-of-flight cameras, spectral imaging, light curtain, and a laser distance sensor in [3], [4]. The rover has many promising features such as increased battery power, battery capacity, and continuous drive torque; however, its wide tracks and limited clearance are serious limitations for plant phenotyping. A new architecture consisting of two mobile robotic platforms, i.e., a phenotyping rover and a mobile observation tower, has been introduced for high-throughput field phenotyping [5]. The phenotyping rover has been built on a Husky A-200 from Clearpath Robotics and collects data from individual plants, while the observation tower identifies specific plants for further inspections. Thus, it eliminates the need of air vehicles. However, a Husky A-200 is too wide (66 centimeters) and heavy (50 kilograms). A ground-based agricultural robot for high-throughput crop phenotyping was developed in [6]. It is capable of autonomously navigating below the canopy of row crops and deploying a manipulator to measure plant stalk strength and gathering phenotypic data. However, like a Husky A-200, it is too wide (56 centimeters) and heavy (140 kilograms). TerraSentia designed in this paper is much narrower (30.5 centimeters) and light (6.6 kilograms) when compared to aforementioned agricultural robots as shown in Fig. 1. Thus, it easily fits between crop-rows and does not compact the soil, which is highly desirable for improving yield.

This paper describes an ultra-compact (30 centimeters wide), ultra-light (6.6 kilograms), very low-cost and autonomous field-phenotyping robot that can navigate in a variety of field conditions to overcome the aforementioned limitations with existing field-based phenotyping systems. A wide variety of systems, e.g., tractors, unmanned aerial vehicles, rovers, and gantry systems, were developed for field-based
Moreover, the algorithm is data-efficient so that it does not need gigabytes of corn-stand data to be utilizable in practice. This is achieved through adopting transfer learning. A support vector machine (SVM) classifier is trained to classify the features extracted from a convolutional neural network with pre-trained weights. The result is a robust stand-counting algorithm where the only sensor needed is a low-cost (< $30) RGB camera. Our extensive field trials show that the detection is robust against noises such as corn leaves, weeds, varying lighting conditions, and residues from the previous year. Our system achieves high accuracy and reliability throughout the growing season.

This paper is organized as follows: The developed NMHE-NMPC framework for high precision control of the ultra-compact mobile robot in off-road terrain is described in Section II. The developed real-time machine vision algorithm for corn stand counting is given in Section III. Field results are given in Section IV, and Section V concludes the paper.

II. HIGH PRECISION CONTROL ALGORITHM

A. System Model

The effect of terrain characteristics on robot performance and wheel require special attention in field-robot design because the wheels are the unique connections between the ground and robot, and nearly all forces and moments applied to the robot are transmitted through the wheels [12], [13]. Due to the wheel-terrain interaction dynamics, soil characteristic plays a vital role in determining vehicle speed and steering, which in turn are utilized for developing traction control algorithms. The knowledge of soil parameters of unknown terrain is then advantageous for improving vehicle performance [14]–[18]. In particular for lightweight robots meant for persistent field use, like TerraSentia robot described here, careful design of control algorithms is necessary to ensure that the robot does not tread over plants due to loss of precision due to unaccounted for wheel-soil interaction.

A system model including traction parameters is developed theoretically and used to establish an effective control law for the 3D printed field robot traveling on rough terrain in this paper. An adaptive nonlinear kinematic model is derived for the ultra-compact 3D printed field robot as an extension of the traditional kinematic model as follows:

\[
\begin{bmatrix}
\dot{x} \\
\dot{y} \\
\dot{\theta}
\end{bmatrix} = \begin{bmatrix}
\mu v \cos \theta \\
\mu v \sin \theta \\
\kappa \omega
\end{bmatrix}
\]  

(1)

where \(x\) and \(y\) are the position of the field robot, \(\theta\) is the yaw angle, \(v\) is the wheel speed, \(\omega\) is the yaw rate, and \(\mu\) and \(\kappa\) are the traction parameters. The difference between the traditional and developed model above is two slowly changing traction parameters. These parameters provide us a learning mechanism that can be used to minimize deviations between the real-time system and ultra-compact 3D printed field robot with an online parameter estimator. It is noted that they must be between zero and one. If the traction parameters are equal to 1, i.e., \(\mu = \kappa = 1\), there exist no longitudinal or side (lateral) slips. The percentages of the longitudinal and side slips can

Fig. 1. The ultra-compact 3D printed field robot, termed TerraSentia, in soybean breeding plots in Energy Farm, Urbana, IL, USA.

phenotyping. However, these systems have many limitations, including high operational and maintenance costs, low coverage area, safety, logistically difficult to maintain internal combustion engines or the need for experienced operators [7]. On the other hand, the ultralight robot described here mitigates many of these challenges, yet, it leads to a uniquely challenging control problem in uneven and unstructured terrain in crop fields. The difficulties in control arise from complex and unknown wheel-terrain interaction (described in detail in Section II-A) and wheel slip. In this paper, a Nonlinear Moving Horizon Estimator (NMHE) that identifies key terrain parameters using onboard robot sensors, and a learning-based Nonlinear Model Predictive Control (NMPC) that enables high accurate tracking are developed to ensure autonomous high-precision robot mobility in off-road terrain.

The other significant contribution of this study is a novel real-time machine vision algorithm that can estimate plant stand count from image sequences obtained from a side-facing camera on an ultra-compact ground robot. This algorithm is demonstrated on the challenging problem of counting corn (Zea mays or Maize) plants in field-conditions; however, the algorithm may also be re-purposed to count other plants, including Sorghum, Wheat, Soybean, or vegetables. The algorithm leverages a cutting-edge convolution neural network architecture that runs efficiently on mobile platforms. Deep learning methods have been shown capable of recognizing complex structures and features in the presence of heavy noise. Today, deep neural networks are approaching human level at image recognition on Internet data [8]–[11]. However, no machine-vision based (whether utilizing deep-learning or not) corn stand counting algorithm currently exists that is robust to real-world noise, varying lighting conditions, and implementable in real-time on an autonomous ground robot. Moreover, the algorithm is data-efficient so that it does not
be respectively found as $1 - \mu$ and $1 - \kappa$. It is assumed that only a fraction of the realized speed and yaw rate translates into actual vehicle motion. These fractions are given by $\mu$ and $\kappa$, and determine the effective speed $\mu v$ and yaw rate $\kappa \omega$. To avoid bias, these parameters should be estimated along with the full system state in each iteration based on a number of past measurements.

In this paper, the nonlinear system and measurement models are represented by the following equations:

$$
\dot{\xi}(t) = f(\xi(t),u(t),p(t)), \quad z(t) = h(\xi(t),u(t),p(t)) \tag{2}
$$

where $\xi \in \mathbb{R}^{n_x}$ is the state vector, $u \in \mathbb{R}^{n_u}$ is the control input, $p \in \mathbb{R}^{n_p}$ is the system parameter vector, $z \in \mathbb{R}^{n_z}$ is the measured output, $f(\cdot,\cdot,\cdot) : \mathbb{R}^{n_x+n_u+n_p} \rightarrow \mathbb{R}^{n_x}$ is the continuously differentiable state update function and $f(0,0,p) = 0 \forall t$, and $h : \mathbb{R}^{n_x+n_u+n_p} \rightarrow \mathbb{R}^{n_z}$ is the measurement function. The derivative of $\xi$ with respect to $t$ is denoted by $\dot{\xi} \in \mathbb{R}^{n_x}$.

The state, parameter, input and output vectors are respectively denoted as follows:

$$
\dot{\xi} = [x \ y \ \theta]^T, \quad p = [v \ \mu \ \kappa]^T
$$

$$
u = \omega, \quad z = [x \ y \ v \ \omega]^T \tag{3}
$$

B. Nonlinear Moving Horizon Estimation

Although model-based controllers need full state and parameter information to generate a control signal, the number of sensors is less than the number of measurable states and parameters in practice. Therefore, state estimators are required to estimate unmeasurable states and parameters. Extended Kalman filter is the most well-known state-estimation method for nonlinear systems. However, EKFs are not capable of dealing with constrained nonlinear systems [19]. As can be seen in (2), the traction parameter estimates play a vital role for the ultra-compact 3D printed robot, and there exist constraints on these parameters, which makes Extended Kalman filter inconvenient for our system. Nonlinear Moving Horizon Estimation (NMHE) approach has a capability of dealing with constraints on state and parameter and is formulated for the ultra-compact 3D printed robot as follows:

$$
\min_{\xi(t),u(t)} \quad \frac{1}{2} \left\{ \| \dot{\xi}(k-N_c) - \dot{\xi}(k-N_c+1) \|_{H_N}^2 + \sum_{i=k-N_c+1}^{k} \| z_m(t_i) - z(t_i) \|_{H_N}^2 \right\}
$$

s. t. $\dot{\xi}(t) = f(\xi(t),u(t),p)$

$$
z(t) = h(\xi(t),u(t),p)
$$

$$
0 \leq \mu \text{ and } \kappa \leq 1, \quad \forall t \in [t_k-N_c,t_k] \tag{4}
$$

where the deviations of the state and parameter estimates before the estimation horizon are minimized by a symmetric positive semi-definite weighting matrix $H_N$, and the deviations of the measured and system outputs in the estimation horizon are minimized by a symmetric positive semi-definite weighting matrix $H_N$ [20]. The estimation horizon is represented by $N_c$, and lower and upper bounds on the traction parameters $\mu$ and $\kappa$ are respectively defined as 0 and 1. The objective function in the NMHE formulation consists of two parts: the arrival and quadratic costs. The arrival cost stands for the early measurements $t = [t_{k-N_c}]$, and the quadratic cost stands for the recent measurements $t = [t_{k-N_c+1}]$.

The measurements have been perturbed by Gaussian noise with standard deviation of $\sigma_e = \sigma_v = 0.03 \text{ m}$, $\sigma_v = 0.0175 \text{ rad/s}$, $\sigma_v = 0.05 \text{ m/s}$ based on experimental analysis. Therefore, the following weighting matrices $H_k$ and $H_N$ are used in the NMHE:

$$
H_k = \text{diag}(\sigma_x^2, \sigma_y^2, \sigma_y^2, \sigma_\theta^2) \tag{5}
$$

$$
= \text{diag}(0.03^2, 0.03^2, 0.5^2, 0.35^2)^{-1}
$$

$$
H_N = \text{diag}(x^2, y^2, \theta^2, v^2, \mu^2, \kappa^2)^{-1}
$$

$$
= \text{diag}(10.0^2, 10.0^2, 0.1^2, 1.0^2, 0.25^2, 0.25^2)^{-1}
$$

The inputs to the NMHE algorithm are the position values coming from the global navigation satellite system, the velocity values coming from the encoders mounted on the DC motors and the yaw rate values coming from the gyro. The outputs of NMHE are the position in x- and y-coordinate system, the yaw angle, the wheel velocity and the traction coefficients. The estimated values are then fed to the NMPC.

C. Nonlinear Model Predictive Control

Nonlinear Model Predictive Control (NMPC) approach is the most well known control method for system with fast dynamics. The reason is that NMPC has the capability of dealing with hard constraints on state and input, and online optimization allows updating cost, model parameters, constraints [21], [22]. The following finite horizon optimal control formulation for the ultra-compact 3D printed robot is solved to obtain the current control action by using current samples and parameters of the system as initial state:

$$
\min_{\xi(t),u(t)} \quad \frac{1}{2} \left\{ \sum_{i=k+1}^{k+N_c} \| \dot{\xi}(t_i) - \dot{\xi}(t_i) \|_{Q_k}^2 + \| u(t_i) \|_{R}^2 \right\}
$$

$$
+ \| \dot{\xi}(t_{k+N_c}) - \dot{\xi}(t_{k+N_c}) \|_{Q_N}^2
$$

s. t. $\dot{\xi}(t_k) = \dot{\xi}(t_k)$

$$
u = \dot{\xi}(t_k-N_c+1)
$$

$$
z(t) = f(\xi(t),u(t),p)
$$

$$
0.1 \text{ rad/s} \leq \omega(t) \leq 0.1 \text{ rad/s}, \quad t \in [t_k,t_{k+N_c-1}] \tag{6}
$$

where $Q_k \in \mathbb{R}^{n_x \times n_x}$, $R \in \mathbb{R}^{n_u \times n_u}$ and $Q_N \in \mathbb{R}^{n_x \times n_x}$ are symmetric and positive semi-definite weighting matrices. $\dot{\xi}$ is the state reference, $\dot{\xi}$ and $u$ are the states and inputs, $t_k$ is the current time, $N_c$ is the prediction horizon, $\dot{\xi}(t_k)$ is the current state estimates and $\dot{\xi}(t_k-N_c+1)$ is the current parameter estimates. The first term in the cost function is the stage cost and it is the cost throughout the prediction horizon. The second term in the cost function is the terminal penalty and it is the cost at the end of the prediction horizon. The terminal penalty is stated for stability reasons [23]. The weighting matrices $Q_k$, $Q_N$ and $R$ are selected based on experimental analyses.
$R$ and $Q_N$ are selected as follows:

$$Q_k = \text{diag}(1,1,1), \quad R = 1 \quad \text{and} \quad Q_N = 10 \times Q_k \quad (7)$$

The first element of the optimal control sequence is applied to the system:

$$u(t_{k+1}, \xi_{k+1}) = u^*(t_{k+1}) \quad (8)$$

and then the procedure is repeated for future sampling times by shifting prediction horizon for the subsequent time instant. It is important to remark that the control input $u^*(t_{k+1})$ is precisely the same as it would be if all immeasurable states and parameters acquire values equal to their estimates based on the estimation up to current time $t_k$ due to the certainty equivalence principle.

The state reference for the 3D printed robot is dynamically changing online and defined as follows:

$$\xi_r = [x_r, y_r, \theta_r]^T \quad (9)$$

where $x_r$ and $y_r$ are the position references, $\omega_r$ is the yaw rate reference calculated from the position references as follows:

$$\theta_r = \text{atan}2(y_r, x_r) + \lambda \pi \quad (10)$$

where $\lambda$ describes the desired direction of the 3D printed field robot ($\lambda = 0$ for forward and $\lambda = 1$ for backward).

### D. Solution Methods

NMHE and NMPC methods for systems require online solutions of nonlinear least square optimization problems at each sampling time [24]. In this study, a single solution method, consisting of a fusion between multiple shooting and generalized Gauss-Newton methods, has been used to solve both optimization problems [25]. This approach is valid because the formulation of NMPC problem is akin to that NMHE problem. The generalized Gauss-Newton method derived from the classical Newton method was developed for least-squared problems. This method is advantageous because it does not require difficult computations of the second derivatives; however, it is challenging to foreknow the number of iterations to reach a solution of the desired accuracy. To overcome this challenge, the solution proposed in [26] has been used where the number of Gauss-Newton iterations is restricted to 1, and the initial value of each optimization problem takes on the value of the previous one intelligently. Hence, this improves the convergence of the Gauss-Newton method.

In this paper, the Gauss-Newton iteration is divided into two parts: preparation and feedback parts. The preparation parts are executed before the feedback parts, and the feedback part is executed after measurements for NMHE and estimates for NMPC are available. In the preparation part, the system dynamics are integrated with the previous solution, and objectives, constraints, and corresponding sensitivities are evaluated. In the feedback part, a single quadratic program is solved with the current measurements for the NMHE and the current estimates for the NMPC. Thus, new estimates for the NMHE and new control signals for the NMPC are obtained. Compared to the classical method, this method minimizes feedback delay and produces similar results with higher computational efficiency.

### III. Corn Stand Counting Algorithm

Conventional machine learning techniques require considerable domain-specific expertise to carefully design a feature extractor to transform raw data (e.g., pixel values of an image) into appropriate feature space where classifiers can detect patterns in the input. In contrast, Deep learning is a set of methods that allow end-to-end training and prediction. Deep learning models take raw data input and automatically learn the representations from the data’s internal structure. Typically a deep learning model consists of multiple modules each of which slightly increases the level of abstraction from the previous representation. When enough such modules are used, very intricate structures can be learned such that relevant patterns are recognized whereas irrelevant variations are suppressed. Deep learning has made tremendous progress in areas that have confounded traditional machine learning for many years including image recognition [9], [11]. However, deep learning requires a lot of human-labeled images, unfortunately, this is something that is rather expensive to obtain. In this paper, we combine convolutional neural network (Section III-A) and support vector machine (Section III-B) via transfer learning to recognize corn plants using very few labeled images.

Recognition alone is not sufficient to count plants on a moving robot since it is difficult to distinguish different corn plants to avoid double-counting. To address this issue, we utilize motion estimation techniques to determine the number of plants that have passed through a fixed Region Of Interest (ROI). Details of this algorithm are given in Section III-D.

#### A. Convolutional Neural Network

It is important to maintain a good balance between performance and efficiency on algorithm to be implemented on mobile platforms, such as robots. MobileNets [27] is a family of networks specially developed for mobile and embedded applications. Due to the use of depthwise separable convolution that requires between 8 to 9 times less computation than standard convolution, the model runs significantly faster than its more complicated counterparts with only slight accuracy compromise. The network parameters are pre-trained on the ImageNet Large Scale Visual Recognition Competition (ILSVRC) dataset [28] with fully connected and softmax layer in the original model replaced by SVM for classification.

##### B. Support Vector Machine

Support Vector Machine (SVM) is a widely used model for classification [29]–[34]. The model constructs a hyperplane that maximizes the distance between the decision boundaries and any data point. The convexity of its loss function gives SVM desirable advantage over other methods such as feedforward neural network that suffers from the existence of multiple local minima and the need for a large dataset to ensure convergence [35], [36]. Therefore, we chose to replace the fully connected layer in the MobileNet with SVM.
for better performance on limited data. The kernel functions of the SVM transform input data from the latent feature space of the deep network to a higher dimensional smooth Reproducing Kernel Hilbert Space. This enables the SVM to classify non-linearly separable features despite being a linear-in-the-parameter classifier.

C. Training

Images are captured by the side facing camera on the robot throughout the growing season. The camera slightly points downward so that only the closest row is visible. Patches are cropped from the images and labeled according to the presence or absence of corn as positive and negative samples, respectively. The number of training and testing samples listed in Table I. We also employ data augmentation to increase the presence or absence of corn as positive and negative samples.

Algorithm 1: Counting algorithm

<table>
<thead>
<tr>
<th>Input</th>
<th>$I_1, \ldots, I_n :=$ image frames</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>$k :=$ average window size; $w :=$ ROI width in pixels; $d :=$ image translation in pixels</td>
</tr>
<tr>
<td>Output</td>
<td>$C :=$ Corn stand count</td>
</tr>
<tr>
<td>Initialize</td>
<td>$C = 0$; $d = 0$</td>
</tr>
<tr>
<td>foreach $I_i$ do</td>
<td></td>
</tr>
<tr>
<td>Extract features by ConvNet $\rightarrow f_i \in \mathbb{R}^{1024}$; Classify by SVM $\rightarrow y_i \in {-1, 1}$;</td>
<td></td>
</tr>
<tr>
<td>Average with neighboring images $\hat{y}<em>i \leftarrow \frac{\sum</em>{i=1}^{N} y_i}{2N+1}$;</td>
<td></td>
</tr>
<tr>
<td>if $\hat{y}<em>i &gt; 0$ then Estimate translation from $I</em>{i-1}$ to $I_i \rightarrow d_i$;  $d \leftarrow d + d_i$;</td>
<td></td>
</tr>
<tr>
<td>end</td>
<td></td>
</tr>
<tr>
<td>else if $\hat{y}_{i-1} &gt; 0$ then multiplicity $M \leftarrow \frac{d_i}{w}$;</td>
<td></td>
</tr>
<tr>
<td>$C \leftarrow C + M$;  $d \leftarrow 0$;</td>
<td></td>
</tr>
<tr>
<td>end</td>
<td></td>
</tr>
<tr>
<td>end</td>
<td></td>
</tr>
</tbody>
</table>

IV. Field Results

A. Real-time Results for Navigation

Linear controllers cannot achieve good trajectory tracking performance on uneven terrain if the system starts off-track [38]. As shown in Fig. 3(a), after the 3D printed field robot is started off-track, it is capable of reaching to the reference trajectory and staying on-track. One of the major problems of such GNSS-based navigated outdoor robots is the missing data values from the satellites. This drawback is inevitable for Real-Time Kinematic (RTK) GNSSs, and the control algorithm has
to be robust to cope with missing data points. As can be observed from Fig. 3(a), although there were 35 missing data points within 1850 data points throughout the experiment, high accurate trajectory tracking performance has been achieved by the NMHE-NMPC framework.

The Euclidean error is shown in Fig. 3(b). The mean value of the Euclidean error after staying on-track is 0.0423 m. The NMPC benefits from traction parameters estimated by the NMHE and results in highly accurate tracking performance so that the Euclidean error is less than 10 cm. The number of violations is shown in Fig. 3(c). The results of multiple experiments indicate that the NMHE-NMPC framework does not violate this error constraint. This demonstrates the capability of the NMHE-NMPC framework.

The traction parameters estimates are shown in 3(d) and within the upper and lower bounds specified in (6). It is clearly seen that they consistently stay close the upper bound and stabilize at values which ensure stable trajectory. It is to be noted that if the soil conditions are unchanging, then hard-coding might result in more accurate trajectory tracking performance. However, the soil conditions are dynamically changing in practice.

The measured and estimated speeds of the ultra-compact 3D printed field robot are shown in Fig. 3(e). As can be seen, the robot has a constant speed, and the NMHE is capable of dealing with noise on the measurements. It is to be noted that some estimated speed values between 300 and 350 seconds are larger than measurements. The reason is that there are missing GNSS measurements as shown in Fig. 3(a). If GNSS measurement is invalid, previous valid measurement is fed to the NMHE instead of an outlier. There are 14 invalid points in the GNSS measurements between 318.8 and 321.6 seconds and these result in the peak values in the speed estimates between 322.8 and 325.6 seconds. The effect on the speed estimates arises seen subsequently because the speed in the system model (2) is considered as a parameter and the parameter estimation is achieved in the arrival cost of the NMHE while the parameters in the estimation horizon are assumed to be time-invariant. Moreover, the longitudinal traction parameter estimate in Fig. 3(d) has nadir values between 322.8 and 325.6 seconds similar to the speed estimate. Consequently, the effective speed $\mu v$ is equal to the speed of the robot despite unavailable GNSS points. The same explanation is the case for the two peak points the speed estimate between after around 340 seconds.

The output of the NMPC, i.e., the desired yaw rate, and yaw rate measurements are shown in Fig. 3(f). The generated control signal by the NMPC is within the lower and upper bounds specified in (6). The performance of the low-level controller is sufficient to track the reference signals despite the high noisy measurements.

**B. Real-Time Results for Corn Stand Counting**

We collected data from various corn fields. The dates and growth stages when data were collected span the entire growing season. In each field, we randomly selected a row and drove the robot at a constant velocity for an arbitrary number of plots. The true populations were counted manually. For data collections prior to October, a small portion of the images were annotated. We collected two additional data-sets in October to test the generalization ability of the model on unseen data from different fields conditions. Table III summarizes the conditions of the data collection. We follow the widely used staging method by Abendroth et al. [39]. The method divides the growth of corn into vegetative phase and reproductive phase. The vegetative phase covers stages from emergence (VE) to tasseling (VT), where stages in-between are identified by the number of collar leaves (e.g. V4). The reproductive phase includes silking (R1), blister (R2), milk (R3), dough (R4), dent (R5), and maturity (R6).

1) **SVM training:** The best hyper-parameters for the SVM found in the grid search are listed in Table IV with their performance metrics on the test data. The hyper-parameters are almost the same for the four growth stages except that the C-value is different for V4, largely due to the difference in appearance of corn plants at V4 from VT, R2, and R6. The RBF kernel consistently works better than the linear kernel, and the optimal $γ$-value for the kernel is identical at four stages.

The performance metrics are also consistently high for different growth stages. It is noted that the training samples exclude leaves and only focus on stems and stalks. The metrics demonstrate that the deep learning model is capable of distinguishing subtle difference in images and thus reduces the inference of leaves and weeds. The mean accuracy for all stages is 91.67%.

2) **Validation: In-Field Corn counting:** Figure 4 shows the corn plant population per plot by robot vs human for each dataset. The robot predictions agree well with the ground truth. The least-square fitted line through all data is given as count$_{robot} = 0.96 \times$ count$_{human} + 0.85$ and a correlation coefficient $R = 0.96$.

Figure 5 shows the box-and-whisker plot of the relative accuracy as in Equation (12) for each dataset. The algorithm achieves consistently high accuracy for all locations and growth stages. It is noted that data from Martinsville and Lebanon were processed without using the new data.
Fig. 3. a) Reference and actual trajectories, and unavailable Global Navigation Satellite System (GNSS) signals on the trajectory. b) Euclidean error. The mean value of Euclidean error is less than 5 cm after being stayed on-track. c) The number of violations. The available space on either side of the field robot is restricted to 0.12 m. It does not violate this accuracy limit. d) Reference and estimated yaw angles, and estimated traction parameters \( \mu, \kappa \). e) Measured and estimated speeds. f) Reference and measured yaw rates.

<table>
<thead>
<tr>
<th>Date</th>
<th>Location</th>
<th>Training</th>
<th>Growth stage</th>
<th>Plots</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jun 6th, 2017</td>
<td>Assumption, IL</td>
<td>Yes</td>
<td>V4</td>
<td>16</td>
</tr>
<tr>
<td>Jul 6th, 2017</td>
<td>Champaign, IL</td>
<td>Yes</td>
<td>VT</td>
<td>10</td>
</tr>
<tr>
<td>Aug 2nd, 2017</td>
<td>Champaign, IL</td>
<td>Yes</td>
<td>R2</td>
<td>28</td>
</tr>
<tr>
<td>Sep 21st, 2017</td>
<td>Ivesdale, IL</td>
<td>Yes</td>
<td>R6</td>
<td>10</td>
</tr>
<tr>
<td>Oct 25th, 2017</td>
<td>Lebanon, ID</td>
<td>No</td>
<td>V6</td>
<td>15</td>
</tr>
<tr>
<td>Oct 26th, 2017</td>
<td>Martinsville, IL</td>
<td>No</td>
<td>R6</td>
<td>8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>V4</th>
<th>VT</th>
<th>R2</th>
<th>R6</th>
</tr>
</thead>
<tbody>
<tr>
<td>kernel</td>
<td>rbf</td>
<td>rbf</td>
<td>rbf</td>
</tr>
<tr>
<td>C</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>( \gamma )</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>accuracy</td>
<td>91.75%</td>
<td>94.38%</td>
<td>92.79%</td>
</tr>
<tr>
<td>precision</td>
<td>0.91</td>
<td>0.91</td>
<td>0.91</td>
</tr>
<tr>
<td>recall</td>
<td>0.95</td>
<td>0.94</td>
<td>0.95</td>
</tr>
<tr>
<td>F1-score</td>
<td>0.95</td>
<td>0.93</td>
<td>0.94</td>
</tr>
</tbody>
</table>

The embedded high precision control and corn stand counting algorithms for a low-cost, ultra-compact 3D printed, autonomous field robot for off-road terrain have been elaborated in this paper.

An NMHE that identifies key terrain parameters using onboard robot sensors and a learning-based NMPC that enables highly accurate tracking have been developed to be capable of autonomously and safely navigating through row based crops. The developed framework for high precision control that enables reliable field robot path tracking in off-road terrain provides less than 5 cm accuracy in the presence of slip and does not violate the 12 cm accuracy limit to avoid crop damage.

The developed vision algorithm based on color imagery recognizes corn stalks robustly in the presence of interference from leaves and weeds, yet only requires relatively small amount of data for training to estimate corn plant population. This was achieved by fusing a leading deep learning model to train SVM. The results demonstrate that the recognition model generalizes well to unseen data and handles real-world variations effectively.

\[
\text{accuracy} = 1 - \frac{\text{count}_{\text{robot}} - \text{count}_{\text{human}}}{\text{count}_{\text{human}}} \tag{12}\]
with Support Vector Machine Classifiers. Our method was tested at five different growth stages and locations. The robot predictions agreed well with the ground truth with the correlation coefficient $R = 0.96$. Least-square regression shows strong correlation ($R = 0.96$) between populations given by robot and human. In addition, the accuracy stays consistently high for all growth stages and locations. These results establish that ultra-compact, lightweight, autonomous agricultural robots equipped with exteroceptive sensors and artificial intelligence could be utilized to relieve the phenotyping bottleneck.

ACKNOWLEDGMENTS

The information, data, or work presented herein was funded in part by the Advanced Research Projects Agency-Energy (ARPA-E), U.S. Department of Energy, under Award Number DE-AR0000598. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Dr. Erkan Kayacan is a postdoctoral researcher at Massachusetts Institute of Technology since February 2018. Any questions or comments should be directed to Girish Chowdhary. We thank EarthSense Inc. for valuable suggestions.

The authors would like to thank Benjamin Thompson and others at Distributed Autonomous Systems Laboratory for valuable contributions to the design and construction of the TerraSentia robot.

REFERENCES


