Toward Robotic Weed Control: Detection of Nutsedge Weed in Bermudagrass Turf Using Inaccurate and Insufficient Training Data

Shuangyu Xie, *Student Member, IEEE*, Chengsong Hu, Muthukumar Bagavathiannan, and Dezhen Song, *Senior Member, IEEE*

***Abstract*—To enable robotic weed control, we develop algorithms to detect nutsedge weed from bermudagrass turf. Due to the similarity between the weed and the background turf, manual data labeling is expensive and error-prone. Consequently, directly applying deep learning methods for object detection cannot generate satisfactory results. Building on an instance detection approach (i.e. Mask R-CNN), we combine synthetic data with raw data to train the network. We propose an algorithm to generate high fidelity synthetic data, adopting different levels of annotations to reduce labeling cost. Moreover, we construct a nutsedge skeleton-based probabilistic map (NSPM) as the neural network input to reduce the reliance on pixel-wise precise labeling. We also modify loss function from cross entropy to Kullback-Leibler divergence which accommodates uncertainty in the labeling process. We implement the proposed algorithm and compare it with both Faster R-CNN and Mask R-CNN. The results show that our design can effectively overcome the impact of imprecise and insufficient training sample issues and significantly outperform the Faster R-CNN counterpart with a false negative rate of only 0.4%. In particular, our approach also reduces labeling time by 95% while achieving better performance if comparing with the original Mask R-CNN approach.**

***Index Terms*—Weed detection, deep Learning, robotic weed control, precision agriculture.**

I. Introduction

We are interested in developing robotic weed removal solutions for environmentally-friendly lawn care. One key issue is to be able to recognize weeds from background turfgrass using a low-cost camera on-board a robot. In this paper, we start with a particular instance: detection of nutsedge weed (*Cyperus spp*.; mix of yellow and purple nutsedges) in bermudagrass (*Cynodon dactylon*) turf.

However, weed detection is nontrivial. To an untrained eye, distinguishing a nutsedge plant from a turfgrass background is difficult especially in a recently mown lawn. Hence the manual data labeling process is expensive and error-prone. The resulting imprecise and insufficient training data is expected to significantly reduce the performance of common data-driven deep learning approaches.

Fig. 1 illustrates how we handle the challenge. First, we propose a data augmentation approach to allow us to combine synthetic data with raw data for neural network training. This significantly reduces the labeling requirement. We propose a data synthesis algorithm to generate high fidelity synthetic data, which also provides accurate labeling. Second, instead of relying on precise pixel-wise labeling, we employ annotations at different levels including bounding box and skeleton model to reduce labeling rigor requirement. Additionally, we propose a nutsedge skeleton-based probabilistic map (NSPM) representation. NSPM (e.g. P_S in Fig. 1) gives more weightage to the structure of nutsedge instead of equal treatment of individual pixels. Third, we modify our neural network loss function from cross entropy, which assumes accurate training samples, to Kullback-Leibler (KL) divergence, which measures the similarity between two probability functions that can take uncertainty in labeling into consideration. At last, we also propose new evaluation metrics to handle imprecise human labeling by extending existing intersection over union (IoU) metric and proposing a new skeleton similarity metrics using NSPM. We incorporate these new designs in a Mask R-CNN framework [1] to complete our detection algorithm.

We have implemented the proposed algorithm and compared it with state-of-the-art methods such as Faster R-CNN [2] and Mask R-CNN. The experimental results have shown that our algorithm significantly outperforms the counterparts. More specifically, the combination of using synthetic data with fine grain labels and raw image data with noisy bounding box labels under KL-divergence loss function leads to the lowest false negative rate of 0.4%. In particular, our approach also reduces labeling time by 95% while achieving better performance if comparing with the original Mask R-CNN approach.

Manuscript received February 21, 2021; accepted July 10, 2021. Date of publication July 20, 2021; date of current version August 2, 2021. This letter was recommended for publication by Associate Editors H. Ryu and S. Manzoor and Editor Y. Choi upon evaluation of the reviewers' comments. This work was supported in part by a T3 grant at TAMU and in part by NSF under NRI-1925037. *(S. Xie and C. Hu are co-first authors of this paper.) (Corresponding author: Dezhen Song.)*

Shuangyu Xie and Dezhen Song are with the Computer Science and Engineering Department, Texas A&M University, College Station, TX 77843-3112, USA (e-mail: sy.xie@tamu.edu; dzsong@cs.tamu.edu).

Chengsong Hu and Muthukumar Bagavathiannan are with the Department of Soil and Crop Sciences, Texas A&M University, College Station, TX 77843, USA (e-mail: huchengsong@tamu.edu; muthu@tamu.edu).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2021.3098012, provided by the authors.

Digital Object Identifier 10.1109/LRA.2021.3098012

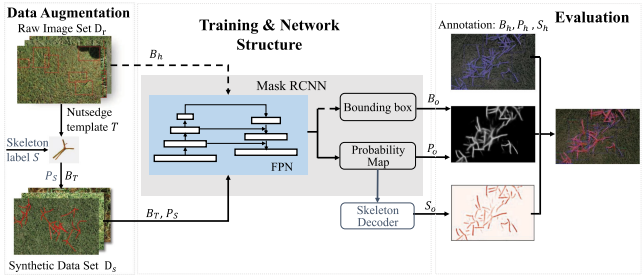

Fig. 1. An overview of nutsedge detection algorithms.

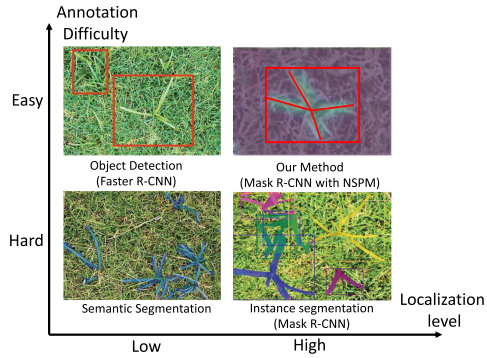

Fig. 2. Dimension of annotation difficulty and localization level for different methods.

II. Related Work

Our work relates to robotic weed control, weed detection, image-based detection and segmentation, and data synthesis.

Robotic weed control: Recently, autonomous robots have seen many applications in precision agriculture because they have enormous potential to reduce operating costs and dependency on labor [3], [4]. A robotic weed control system often includes three components: a sensing system to detect weeds, a decision-making unit to process the information from the sensing system and make manipulation decisions, and actuators to act accordingly [5]. Our work belongs to the first component [6]. For robot decision making, it is important to localize individual plants, target the weeds and avoid crop plants [7]. For actuation, selection of the actuation (weed-killing) mechanism is under fast development. Common actuation methods include cultivation tools [8], stamping [9], mowing [10], precise herbicide application [11], etc. In addition to the actuator development, modular robotic platforms that are able to carry various weeding actuators are also under active development [12].

Image-based weed detection and segmentation: In this area, methods can be categorized into two types: traditional computer vision methods and learning-based methods. Our weed detection algorithm developed here belongs to the latter. Before learning-based methods are widely adopted in solving weed detection problems, traditional computer vision methods that extract hand-craft plant visual characteristics have been commonly used. These characteristics can be classified into two major groups: visual texture and biological morphology [5]. For example, Burks *et al.* [13] utilize the color co-occurrence method to discriminate textures between five common weed species. Herrera *et al.* [14] propose a strategy utilizing a set of shape descriptors to discriminate grasses from broad-leaf weeds, which works when weeds are at an early stage of growth.

Convolution neural network (CNN)-based methods outcompete traditional computer vision-based methods in feature extraction and have become more popular for weed detection nowadays. Many previous works employ CNNs to detect weeds in various crops, including soybean [15], cereal crops [16], ryegrass [17], canola [18] and rice [19]. These methods produced satisfactory results in distinguishing the weed from highly color contrasted soil background. However, with the turfgrass background, the weed detection problem becomes more challenging, and we are developing new methods here to improve detection performance.

With the increasing capability of detection networks such as Faster R-CNN [2], YOLO (You Only Look Once) [20], and SSD (Single Shot Detector) [21], object detection against a highly-similar background can be achieved effectively. However, these object detection networks only provide bounding box output, which is not adequate for further field operation, especially localization. The localization problem can be partially addressed by segmentation networks such as Mask R-CNN [1] and Deeplab [22], because these networks achieve finer image segmentation results for objects of interest. The problem with such methods is the tremendously high annotation cost, i.e. these networks often require pixel-wise precise ground truth for training, which is difficult and expensive for weed detection problems.

Considering the unique shape of nutsedge leaves and plant architecture, extracting plant skeleton of nutsedge is a good approximation of semantic structure. In fact, the skeleton detection is also widely explored with end-to-end deep learning methods such as DeepFlux [23] and Hi-Fi [24]. Although these methods only target single object detection, which are not directly applicable in our scenarios, this inspires our development of nutsedge skeleton probability map to balance between the robustness of localization and annotation cost (Fig. 2).

Using synthetic data: Researchers have explored different methods for data augmentation to enhance neural network training results, especially in domains where annotated data is difficult to obtain or expensive. Generative adversarial networks (GAN) is one of the methods that has gained popularity [25]. However, training a GAN model to converge in specific tasks is often complicated and time consuming due to its adversarial nature. Thus, an easily accessible method for data augmentation is needed for nutsedge detection. Therefore, we synthesize images using real object segments. This approach involves a segmentation stage where nutsedge templates are extracted from real images either manually or automatically, and a synthesis stage where the extracted foreground nutsedge templates are pasted to the background of interest (i.e. turfgrass). Using a similar approach, Gao *et al.* [26] train a YOLOv3 model for weed and crop detection, and achieve a mean average precision of 0.829. Toda *et al.* [27] show that a Mask R-CNN model for barley seed morphology phenotyping can be trained purely by a synthetically generated dataset where 96% recall and 95%

average precision against real test dataset were achieved. Inspired by these results, we are developing a data-synthesis-based approach for weed detection problems.

III. Problem Definition

Our robot observes field through a downward facing camera to collect images (see video attachment for more details). Therefore, all images are collected from a perspective that is perpendicularly facing the ground from the same distance (0.5 m in our set-up).

Common notations are defined as follows:

- binary random variable $\mathbf{x}_{uv} = 1$ indicates event that pixel (u, v) is a nutsedge pixel on the image where u and v are pixel indexes in horizontal and vertical directions, respectively.
- $p(\mathbf{x}_{uv})$, probability of pixel at (u, v) is a nutsedge pixel.
- $I_r := \{(u, v) : \forall(u, v)\}$, pixel set of a raw image collected from the field.
- $P_o := \{p(\mathbf{x}_{uv}) : \forall(u, v) \in I_r)\}$, a probability map set describing spatial probability distribution of $\mathbf{x}_{uv}$. It is the part of the output of the neural networks characterizing the confidence of the prediction.
- $\mathbf{B} = \{B\}$ is a set of bounding boxes with each $B = \{(u, v)|u \in [u_{\text{left}}, u_{\text{right}}], v \in [v_{\text{bottom}}, v_{\text{top}}], (u, v) \in I_r\}$ where $(u_{\text{left}}, v_{\text{bottom}}) \in I_r$ and $(u_{\text{right}}, v_{\text{top}}) \in I_r$ is the bottom-left and top-right corners of the output bounding box, respectively. We use $\mathbf{B}_h$ representing human labeled bounding box set and $\mathbf{B}_o$ as algorithm output bounding box set.
- $\mathbf{S} = \{S\}$ is a set of plant skeleton S which will be defined later. We use $\mathbf{S}_h$ representing human labeled skeleton set and $\mathbf{S}_o$ as the algorithm output skeleton set.

The weed detection problem can be defined as follows,

Definition 1: Given the image collected by robot I_r, compute $\mathbf{B}_o$, $\mathbf{S}_o$ and P_o.

IV. Algorithms

Our algorithm development consists of three major components: data augmentation, network design & training, and evaluation (Fig. 1). Our data augmentation algorithm addresses the issue of insufficient training data by combining synthesised data with manually-labeled data. Due to the non-negligible level of errors existing in manually annotated labeling, we revised the network design & training to handle the inaccurately labeled training data. For the same reason, we cannot entirely trust the manually-labeled data as the ground truth and have to design a new evaluation pipeline considering the labeling noise to validate our model. We begin with the data augmentation.

A. High Fidelity Data Augmentation

As detailed later, we employ deep neural networks for weed recognition which often require massive manually labeled data as the ground truth for training. We develop an image synthesis algorithm to efficiently generate high-fidelity artificial dataset from images collected from the field with different granularity

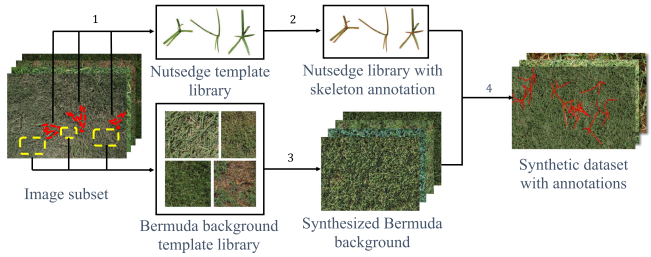

Fig. 3. An overview of the image synthesis pipeline. 1,2,3 and 4 represents template selection, annotation, background synthesis and recombination. Step 4 is the combined stage of image synthesis algorithm whose pseudocode is attached in the attached video file.

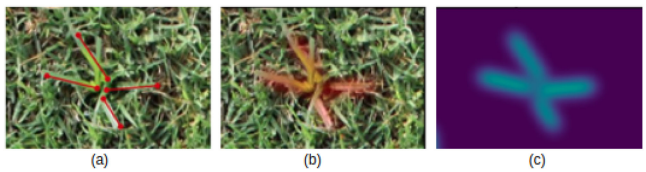

Fig. 4. An example of NSPM: (a) skeleton from the data synthesis, (b) pixels masked as nutsedge in the synthesized image, and (c) the resulting nutsedge skeleton probability model.

labels. Using data augmentation with image synthesis algorithm instead of directly labeling the raw images has three specific advantages: 1) it requires a minimal human labeling effort, 2) it expands the size of training dataset, and 3) it provides precise pixel-level labels.

In the synthetic dataset, each image is composed by nutsedge foreground and bermudagrass background. To generate realistic synthetic images with label, we ask human experts hand-select a small number of nutsedge templates and background patches from a raw image set as a material library (red and yellow dash shapes in the leftmost image in Fig. 3). Then, image synthesis algorithm creates complete label sets for nutsedge template based on human expert's partial label for training purpose. The complete image synthesis algorithm consists of the following four steps corresponding to steps 1-4 in Fig. 3.

1) Template Selection: There are two libraries needed: a nutsedge template library (with skeleton and masking label) and a turf background library. To reduce the work load of human experts, we first employ the stratified random sampling [28] based on the lighting condition to build an image subset (5% of the training set) with images under different lighting conditions proportional to raw image set. Human experts segment out nutsedge template $T \subset I_r$ where T is a polygon covering nutsedge pixels, and nutsedge-free turf background pixel patches from the sampled image set.

2) Nutsedge Annotation: For each nutsedge template T, there are 3 different types of annotation: plant skeleton S, binary mask M_s, and bounding box B_T.

To simplify the labeling process and better describe the structure attribute of nutsedges, we use plant skeleton labeling S in our network design. As illustrated in Fig. 4(a), S models each nutsedge plants as a cluster of line segments where each line segment depicts the center of a leaf,

$$S := \{\mathbf{l}_k : k = 1, \ldots, k_{\max}\}, \tag{1}$$

where $k_{\max}$ is the total number of the line segments, and line segment $\mathbf{l}_k = \{(u,v),(p,q)\}$, $(u,v) \in I_r$ and $(p,q) \in I_r$ are endpoints of the line segment. In the annotation process, one skeleton corresponds to one bounding box. The line segments forming skeleton are annotated by human expert.

The mask labels and bounding boxes are generated automatically from nutsedge templates T. Following the manner of instance segmentation dataset creation [29], M_s labels are created by setting all template pixel as 1 for foreground nutsedge pixels and 0 otherwise. The bounding box computed from T is defined as

$$B_T := \{(u,v) | u \in [u_{\text{left}}, u_{\text{right}}], v \in [v_{\text{bottom}}, v_{\text{top}}], (u,v) \in I_r\}, \tag{2}$$

where $u_{\text{left}} = \min\{u\}$, $v_{\text{bottom}} = \min\{v\}$, $u_{\text{right}} = \max\{u\}$, $v_{\text{top}} = \max\{v\}$, and $(u,v) \in T$.

3) Background Synthesis: To generate realistic background images with appropriate size and scale, a natural texture synthesis algorithm [30] is employed. The advantage of using this algorithm over directly tiling with background templates is that it adds randomness to the synthesized background so as to prevent the neural network from picking up the unique patterns of each background template.

4) Recombination of Nutsedge and Background: After background synthesis, the foreground of randomly selected subsets of the nutsedge template library are pasted onto the synthesized background images. The size of the subsets follows a uniform distribution within a desired range (this range is determined by experiment settings). While pasting each nutsedge template, the pixel locations in homogeneous coordinate are transformed by 2D coordinate transformation matrix $\begin{bmatrix} \cos(\theta) & \sin(\theta) & t_x \\ -\sin(\theta) & \cos(\theta) & t_y \\ 0 & 0 & 1 \end{bmatrix}$ where θ is a random rotation angle within $[0, 2\pi)$, and t_x & t_y are horizontal and vertical random translations, respectively. They have uniformly distributed value within the image boundary. The resulting images are then augmented in hue, saturation, value (HSV) color space by randomly varying brightness value from 80% to 120% so that the trained models are more robust to the color variation in the testing dataset as a result of inconsistency for light conditions. The skeleton annotations of each nutsedge template are also inserted during the image synthesis process.

The overall time complexity of the proposed image synthesis algorithm is $\mathcal{O}(u_{\max} v_{\max} s^2)$ where $(u_{\max} v_{\max})$ is the maximum image size in pixel count, and s is the neighbourhood size for pixel candidate searching. In our implementation, the neighbourhood size s is 24 pixels. The detailed pseudocode and anlysis is in the attached video file.

B. Network Design and Training

With both synthesized data and human-annotated training data (i.e. all raw image training set comes with human-labeled bounding boxes), we employ Mask R-CNN [1] to develop our detector. In the original Mask R-CNN structure, the binary mask branch segments the image by assigning each pixel to a class. To better capture the feature of nutsedge while considering the imprecision in training dataset, we design a skeleton probability map representation of mask and modify the loss function of Mask R-CNN's mask branch correspondingly.

1) Nutsedge Skeleton-Based Probabilistic Map Generation: For nutsedge segmentation problem, the difficulty of distinguishing the boundary of nutsedge's class increases as the distance from the center of nutsedge grows. Meanwhile, detecting the center and leaf midrib of the nutsedge is more important than detecting its edges for plant recognition. This motivates us to propose the use of NSPM input. The purpose is to instruct the network to differentiate the central leaf midrib part of the nutsedge, while reducing the impact of imprecision in nutsedge boundary segmentation.

Fig. 4 illustrates NSPM computation. The bounding box for a skeleton S is defined as $B_h := \{(u,v) | u \in [u_{\text{left}}, u_{\text{right}}], v \in [v_{\text{bottom}}, v_{\text{top}}], (u,v) \in I\}$ in similar format of B_T in (2) with u_{left}, v_{bottom}, u_{right}, v_{top} determined by human labeling instead of T. For image I, we define the bounding box set as $\mathbf{B}_h = \{B_h\}$. For pixel $(u,v) \in B_h$ which contains the plant skeleton S, the probability of (u,v)'s class is nutsedge

$$p_S(\mathbf{x}_{uv}) \propto \begin{cases} \sum_{k=1}^{k_{\max}} \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\left[\frac{d((u,v),\mathbf{l}_k)}{\sigma}\right]^2\right\} \\ \qquad \text{if } ((u,v) \in B_h) \wedge (B_h \in \mathbf{B}_h) \\ 0, \qquad \text{otherwise.} \end{cases}$$

where $d((u,v),\mathbf{l}_k)$ is the point (u,v) to line segment $\mathbf{l}_k$'s nearest point's distance, and we use the nutsedge template and the skeleton labeled by human expert to estimate the proper value of σ. By drawing the histogram for each nutsedge template in the template library with $d((u,v),\mathbf{l}_k)$ as x-axis and count of pixel number as y-axis, we can use half-normal distribution to approximate the histogram and estimate σ.

The NSPM P_S of the image is defined as

$$P_S(u,v) = \{p_S(\mathbf{x}_{uv}), \forall (u,v) \in I\} \tag{3}$$

P_S is used as the annotation input for the training image I.

2) Modifying Loss Function: At the same time, we need to modify the original loss function (cross entropy) in mask branch to accommodate labeling imprecision. In original loss function, it maps origin binary annotation (ground-truth) value to discrete distribution for binary mask M_s as $p_1(\mathbf{x}_{uv}) \in \{0,1\}$ and represents mask branch output as probability density function $p_2(\mathbf{x}_{uv}) \in [0,1]$

$$L_H(p_1,p_2) = -\sum_{(u,v)\in B_o} p_1(\mathbf{x}_{uv}) \log(p_2(\mathbf{x}_{uv})). \tag{4}$$

The problem of cross entropy loss function is that it is designed for deterministic annotation without considering the uncertainty introduced by the imprecision in labeling. To address this problem, we introduce KL-Divergence as the loss function for mask branch that perceives the uncertainty in human annotation and model it as a probability distribution using NSPM, where the

annotation's probability distribution is $p_1(\mathbf{x}_{uv}) = P_S(u, v)$.

$$L_{KL}(p_1, p_2) = -\sum_{(u,v)\in B_o} p_1(\mathbf{x}_{uv}) \log\left(\frac{p_1(\mathbf{x}_{uv})}{p_2(\mathbf{x}_{uv})}\right). \quad (5)$$

It is worth noting that (4) and (5) share the same time complexity in computation. When we replace the cost function (4) with (5), our revised Mask R-CNN share the same time complexity with the original version.

3) Transfer Learning Using Data With Different Levels of Annotation: A worth-mentioning design of our training dataset is that images have labels at different levels of granularity. Human labeled raw image set only contains the bounding box annotation, while the synthesized data generated by nutsedge template have higher precision level labels: binary mask and plant skeleton label.

To efficiently train our model with different annotation levels, we develop a new training strategy for Mask R-CNN. As an instance segmentation network, Mask R-CNN outputs the bounding box, the class of bounding box, and the binary mask of nutsedge. All the three branches share the same backbone feature extraction and Region Proposed Network (RPN) [2]. Our training strategy fully exploits the structure's potential. First, we employ raw image I_r with human labeled bounding box B_h to train the model's classification and bounding box detection branch to ensure that the feature extraction network has been mostly trained from real data's distribution and human observation (Fig. 1, dash line's flow). Second, we fine-tune the feature extractor and train the original mask branch using synthesized data I_s with its label M_s (Fig. 1, solid line's flow).

4) Skeleton Decoder: When we train the Mask R-CNN, we adopt ResNet-FPN [31] backbone to obtain feature fusing map in the feature extraction stage. With the high-resolution and high-level semantic map embedded in the same feature map, the model learns complex semantic information through training. The inference output probability map P_o has a higher probability in the midrib of leaves. This attribute of the probability map enables us to extract nutsedge skeleton from it. After receiving the probability map P_o, we adopt pre-processing morphology dilation and erosion with the Gaussian blur to make the probability map distribution more smooth. Then, we apply a non-maximum suppression skeleton selection [24] algorithm to the pre-processed probability map to decode its skeleton structure.

C. Semi-Supervised Evaluation

Standard evaluation methods for detection and segmentation problem often compare the region similarity using intersection over union (IoU) metric between the model output and label of the bounding box (ground-truth). As we described early, due to the labeling imprecision, human annotation cannot be treated as ground truth. Thus, a new evaluation method is needed. Here we design evaluation methods targeting situations when human annotations and model are consistent or inconsistent, respectively.

1) Consistent Metrics: In this step, we evaluate how model outputs compare to bounding box set labeled by human ($\mathbf{B}_h$) when they are consistent. For this purpose, we compare both pixel-wise region overlap and skeleton similarity.

- **Region overlap:** With human labeled bound box $\mathbf{B}_h$ set and skeleton S_h set, we can obtain probability map P_S using (3). We can threshold P_S to obtain region set I_S according to human labels,

$$I_S := \{(u, v) | p_S(\mathbf{x}_{uv}) > t\} \subseteq I_r, \quad (6)$$

  where t is probability threshold. Similarly, we can obtain region set I_o according to the model output probability map P_o using the same threshold. The region overlap between I_S and I_o can be measured by IoU metric,

$$r_{\text{IoU}} = \frac{|I_S \cap I_o|}{|I_S \cup I_o|}, \quad (7)$$

  where $|\cdot|$ is set cardinality.
- **Skeleton similarity:** We use the skeleton similarity between S_o and S_h to evaluate how well the model capture main structure of the nutsedge. First, for each pixel (u, v) in S_o, if we can find the distance $d_{S_h}(u, v)$ to its closest point in S_h,

$$d_{S_h}(u, v) = \min_{(u_a,u_b)\in S_h} \sqrt{(u_a - u)^2 + (u_b - v)^2}). \quad (8)$$

  If $d_{S_h}(u, v)$ is less than a given threshold d, we believe that the pixel (u, v) has a corresponding point in S_h. We obtain the ratio between the corresponding pixel counts in S_h and the total pixel number in S_h,

$$C_s = \frac{|\{(u, v) | (u, v) \in S_o, d_{S_h}(u, v) \le d\}|}{|S_h|} \quad (9)$$

  as the skeleton similarity metric.

2) Inconsistent Metrics: For our problem, it is possible that the model fails to recognize a nutsedge and it is also possible that human may make mistakes in annotation. We want to catch these inconsistent cases and further analyze them.

First, we identify the consistent bounding box set $\mathbf{R}_a$,

$$\mathbf{R}_a = \{B \mid B \in \mathbf{B}_h \cap \mathbf{B}_o, (r_{\text{IoU}} \ge 0.5) \vee (C_s \ge 0.7)\},$$

where r_{IoU} and C_s are computed using (7) and (9) respectively. Then we obtain the inconsistent bounding box set $\mathbf{R}_c = \{(\mathbf{B}_h \cup \mathbf{B}_o) \setminus \mathbf{R}_a\}$. When inconsistent cases are detected, we manually reexamine the labels of those bounding boxes and classify $\mathbf{R}_c$ into three groups: 1) false positive case set of algorithm output $\mathbf{B}^o_{\text{FP}}$, 2) false negative case set of algorithm output $\mathbf{B}^o_{\text{FN}}$, and false negative set of human annotation $\mathbf{B}^h_{\text{FN}}$.

V. Experiment

We have implemented our weed detection algorithm based on Detectron2 [32] system on Pytorch platform. We choose ResNet-50 with Feature Pyramid Networks (FPN) and ResNet-101 with FPN as our backbone network. The initial network parameters

of Faster R-CNN and Mask R-CNN are both from a pre-trained model on MSCOCO dataset [29].

A. Nutsedge Dataset

We have built the a shared TAMU nutsedge dataset [33] which contains two types of data: the raw image set collected from the field with manual annotations and synthetic image set with ground truth synthetic label.

1) Raw Image Set: The raw images were collected at the Scotts Miracle-Gro Facility for Lawn and Garden Research, Texas A&M University using Nikon$^{\mathrm{TM}}$ D3300 or Canon EOS Rebel T7$^{\mathrm{TM}}$ mounted at a height of 0.5 m on a data collection cart. See attached video file for more details. The original image resolution is 6000×4000 but downsized to 1200×800 to adapt the model and reduce training costs. To cover the appearance variation of nutsedge, data are collected at different lighting conditions, temperature, weather, and moisture levels. To cover the majority of nutsedge growth season, data were collected from June to August at different times of day. The raw dataset contains 6000 images which is split into a training set D_r (90%) and a testing set D_t (10%). All data are labeled with bounding boxes for both training and testing purposes. In addition, 25% of the testing images contain skeleton label. We denote the testing set with skeleton label as $D_{t_S} \subseteq D_t$. The size of D_{t_S} is $n_{t_S} = |D_{t_S}|$. All the labels are created by human annotation using "labelme"[34] tool.

2) Synthetic Dataset: Generated using the method in Section IV-A, our synthetic dataset contains 4750 images with bounding box labels, which are used as the training set. The density of nutsedges is set at 5 to 10 plants per one million pixels. When we generated the NSPM, we set $\sigma = 12$ pixel based on statistical analysis of existing data. Moreover, the dataset contains both binary mask label and skeleton label. When only the binary pixel-level mask label is used with the synthetic dataset, we name it as D_{s_b}. When only skeleton label is used with the synthetic dataset, we name it as D_{s_p}. $|D_{s_b}| = |D_{s_p}| = 4750$. The sample images of synthesized dataset is shown in attached multimedia file.

3) Reduction of Labeling Time: The data synthesis algorithm significantly reduces manual labeling effort. The average density of nutsedge in raw dataset is 10 plants per image. It takes about 30 seconds to label for each plant. To label all 800 raw images with mask label, it would cost 66 hours. With the help of image synthesis algorithm, we only need to select and create mask label for 129 nutsedge template. The labeling time is reduced to less than 3 hours which is a 95% reduction in labeling time. Also, the synthetic data contains ground truth that is not attainable in noisy manually labeled data.

B. Component Tests

1) Loss Function Comparison: We train a Mask R-CNN model using cross entropy loss function with dataset D_{s_b} and using KL-divergence with dataset D_{s_p}. The $r_{\text{S-IoU}}$ is an average B_h's r_{IoU} in an image weighted by skeleton size. We calculate the $\overline{r}_{\text{IoU}}$ by averaging all image's $r_{\text{S-IoU}}$. Let $n_b = |\mathbf{B}_h|$ in one image and the total pixel count of skeleton in the image be $c_{I_S} = \sum_{n_b} |S_h|$. We have

$$r_{\text{S-IoU}} = \frac{1}{n_b} \sum_{n_b} \frac{|S_h|}{c_{I_S}} r_{\text{IoU}} \quad \text{and} \quad \overline{r}_{\text{IoU}} = \frac{1}{n_{t_S}} \sum_{n_{t_S}} r_{\text{S-IoU}}. \tag{10}$$

Similarly, we extend the skeleton similarity metric,

$$C_{Ss} = \frac{1}{n_b} \sum_{n_b} \frac{|S_h|}{c_{I_S}} C_s \quad \text{and} \quad \overline{C}_s = \frac{1}{n_{t_S}} \sum_{n_{t_S}} C_{Ss}. \tag{11}$$

TABLE I
DETECTION COMPARISON

Training	Testing	Backbone	Loss	$\overline{r}_{\text{IoU}}$	$\overline{C}_s$
D_{s_b}	D_{t_S}	R50	CE	0.42	0.75
D_{s_b}	D_{t_S}	R101	CE	0.45	0.77
D_{s_p}	D_{t_S}	R50	KL	0.48	0.81
D_{s_p}	D_{t_S}	R101	KL	0.57	0.88
$D_{s_p} \cup D_r$	D_{t_S}	R101	KL	**0.61**	**0.88**

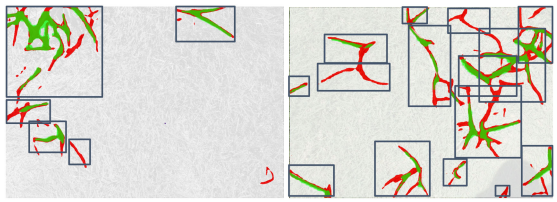

Fig. 5. A comparison of the detection results with cross entropy (in green) and KL-divergence (in red) models. The grey boxes are bounding boxes from manual labeling. It is clear that there are a lot more red pixels than green pixels, which means that the KL-divergence loss function misses fewer than the cross entropy loss function. Both models use R101 as the backbone.

The overall result is shown in Table. I. We use R50, R101, CE and KL representing the ResNet-50-FPN, ResNet-101-FPN, cross entropy and KL divergence, respectively. It is clear that changing the loss function from CE to KL achieves higher $\overline{r}_{\text{IoU}}$ and $\overline{C}_s$. Even with a smaller backbone network (R50), the model trained by KL loss function performs better than that by R101 using CE loss function by 3% in $\overline{r}_{\text{IoU}}$ and 4% in $\overline{C}_s$. When the backbone is identical, the model with KL loss improves over the CE by more than 10% in both $\overline{r}_{\text{IoU}}$ and $\overline{C}_s$. Sample results are shown in Fig. 5.

2) Improvement With Transfer Learning: We follow the basic rules of transfer learning by using the pre-trained model to improve the performance. In general case, without the task-specialized pre-trained model, the common models such as the one trained by MSCOCO is used as the pre-trained model. The first four lines in Table I use MSCOCO pre-trained model as initial parameters. To get further improvement, we use D_r to pre-train the backbone and bounding box branch. The performance of model with D_r pre-trained and R101 as backbone lists in the line 5 of Table I which is highlighted in bold font as the best performer.

3) Synthetic Data Generation Configuration: Synthetic data provides accurate ground truth with pixel-level mask, which is expected to substantially improve the model. We study how the number of foreground nutsedge and background bermudagrass templates (Section IV-A1) in generating synthetic data affects

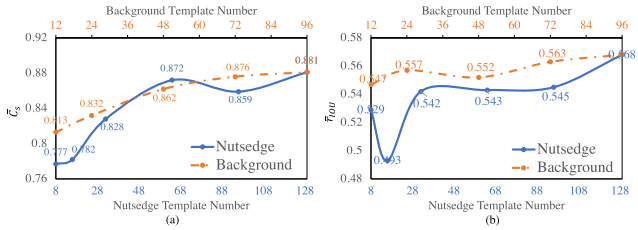

Fig. 6. Affect of different number of nutsedge and background templates in generating synthetic training data.

the overall detection performance. First, we vary nutsedge foreground template sizes while keeping the background template number at 96. We increase the number of nutsedge templates from 8 to 129. Again, $\overline{r}_{\text{IoU}}$ and $\overline{C}_s$ are used to evaluate the detection result (Fig. 6). With mere 8 nutsedge templates, the trained model achieves $\overline{r}_{\text{IoU}}$ of 52.9% and $\overline{C}_s$ of 77.7%. As the nutsedge templates increase, the $\overline{r}_{\text{IoU}}$ gradually grows to 56.8% and $\overline{C}_s$ reaches 88.1%. Similarly, we test our algorithm by changing the background template number from 12 to 96 while fixing the number of nutsedge templates at 129. $\overline{r}_{\text{IoU}}$ and $\overline{C}_s$ are 54.7% and 81.3%. The curve in Fig. 6 also illustrates the positive correlation between the number of background templates and the model performance, but the trend is relatively less significant compared to that of the nutsedge template number. Considering the fact that selecting templates is expensive, we choose 129 nutsedge templates with 96 background templates as our setup in generating synthetic data.

C. Overall Performance Comparison

1) Algorithms and Training Setup: The overall evaluation compares the four algorithms indicated below (Algs. a-d) under their required training setup. In fact, Algs. c-d are our algorithms with different configuration.

a) Faster R-CNN based model with R101 backbone: this setup only uses bounding boxes as training set input and algorithm output, and it does not require pixel-level labeling A sample input is shown in the top left image in Fig. 2. This algorithm serves as a baseline for Faster R-CNN.

b) Mask R-CNN based model with R101 as the backbone and trained by CE loss function: Here we use synthetic data with binary pixel-level mask label D_{s_b}. This algorithm tests the power of synthetic data and can also be viewed as an approximate baseline for Mask R-CNN with precise labeling. A sample input is shown in bottom right image in Fig. 2. The typical application of the original Mask R-CNN would require fully manually labeled pixel-wise training data. In fact, the synthetic data remove labeling noise which may make the algorithm performs better than the actual case. Also, the synthetic data may not be as representative as the precise real data which are not available. That is reason for us to call it an approximate baseline for Mask R-CNN. The comparison is not exact but still meaningful.

c) We change Alg. b settings by swapping the loss function from CE to KL divergence in (5). The swapping also allows us to use skeleton-labeled synthetic set D_{s_p}. This algorithm examines if the change of loss function improves the performance.

d) We further extend the model c with a pre-trained model described in Sec. V-B2. Also, real training set D_r is used in combination with D_{s_p}. This algorithm is presumed to be the best overall according to the component test.

All models are tested on the raw image set D_{t_S}.

TABLE II
OVERALL PERFORMANCE COMPARISON

Alg.	Loss	Training set	r_d	r_a	r_{FN}	r_{FP}
a	CE	D_r	3.01	-	-	-
b	CE	D_{s_b}	**22.71**	94.3%	5.0%	**0.2%**
c	KL	D_{s_p}	21.14	96.8%	0.7%	1.7%
d	KL	$D_{s_p} \cup D_r$	18.91	**97.1%**	**0.4%**	4.4%

2) Metrics and Results: To compare the detection ability of algorithm with only bounding box output (a) and Algs. with precise pixel-level output (b-d), we define the density ratio r_d as the ratio between nutsedge density of detection region and density of the entire image:

$$r_d = \frac{c_a/c_o}{c_s/c_I},$$

where c_s is the total number of nutsedge pixels, c_I is the total pixel count of the testing image, c_a is the total number of nutsedge pixels covered by output bounding boxes, and c_o is the pixel count for the union area of the output bounding boxes. c_s and c_a are based on human labeling results since Alg. a's input and output are just bounding boxes. High values of r_d indicate better detection because the algorithm is able to identify focused regions with more nutsedges. Table II shows the result. It is clear that Algs. b-c perform much better than Alg. a. This is expected because raw image with human label contains high error in training samples, which negatively affect detection results. For Algs. b-c, the use of synthetic data greatly improves network training.

For Algs. b-c, r_d does not tell the complete story. We need to take a closer look because not all nutsedge pixels are equal or error-free. Further, we are also interested if disagreements between algorithm and human labeling can reveal more insights. To focus on this, we need new metrics that do not simply treat human label as ground truth. Let $\mathbf{N}_d$ be the total detected nutsedge bounding box set based on both algorithm output and human labeling. It is a union of consistent case $\mathbf{R}_a$, cases missed by model output $\mathbf{B}^o_{\text{FN}}$, and cases missed by human label $\mathbf{B}^h_{\text{FN}}$: $\mathbf{N}_d = \{\mathbf{R}_a \cup \mathbf{B}^o_{\text{FN}} \cup \mathbf{B}^h_{\text{FN}}\}$. It is worth noting that these metrics build on segmented nutsedge pixels (i.e. region overlap in (7) and skeleton similarity (9)). Cases outside $\mathbf{R}_a$ are subjected to a manual re-examination step to determine which ones are correct. These metrics do not apply to Alg. a due to its lack of segmentation capability. For the rest, these sets allow us to define the agreement rate r_a, false positive rate of model r_{FP} and false negative rate of model r_{FN} as model comparison metrics.

$$r_a = \frac{|\mathbf{R}_a|}{|\mathbf{N}_d|}, \quad r_{\text{FP}} = \frac{|\mathbf{B}^o_{\text{FP}}|}{|\mathbf{N}_d \cup \mathbf{B}^o_{\text{FP}}|}, \quad \text{and } r_{\text{FN}} = \frac{|\mathbf{B}^o_{\text{FN}}|}{|\mathbf{N}_d|}.$$

Table II shows that Alg. d achieves the best overall results. This is due to high overall agreement between human and algorithm output and the lowest false negative ratio. Algorithms with low false negative detection help remove weeds more thoroughly. However, in situations where herbicide use reduction is much more important than thorough weed control, we may want to choose Alg. b due to its lowest false positive rate.

VI. Conclusion and Future Work

We reported our weed detection algorithm development for robotic weed control. We focused on detecting nutsedge weed in bermudagrass turf. Building on the Mask R-CNN, an instance segmentation framework, our new algorithm incorporated four new designs to handle the imprecision and insufficiency of training datasets. First, we proposed a data synthesis method to generate high fidelity synthetic data. We combined the precise labeling from the synthetic data and noisy labeling from the raw data to train our network. We also proposed new data representation to allow the network to focus on the skeleton of the nutsedge instead of individual pixels. We modified the loss function to enable Mask R-CNN to handle training data with high uncertainty. We also proposed new evaluation metrics to facilitate comparison under imprecise ground truth. The experimental results showed that our design was successful and significantly better than the Faster R-CNN approach.

In the future, we will extend our approach to more types of weeds and turf species. Building on these results, we will also develop robotic weed removal algorithms and systems, and test them under field conditions.

Acknowledgment

The authors would like to thank H. Cheng, S. Yeh, A. Kingery, A. Angert, and D. Wang for their early feedback and contributions to the NetBot Lab at Texas A&M University. We acknowledge Z. Mansour and B. Wherley for assisting with field image collection.

References

[1] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proc. IEEE Conf. Comput. Vis*., 2017, pp. 2961–2969.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[3] T. C. Thayer, S. Vougioukas, K. Goldberg, and S. Carpin, "Multirobot routing algorithms for robots operating in vineyards," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 3, pp. 1184–1194, Jul. 2020.

[4] A. You, F. Sukkar, R. Fitch, M. Karkee, and J. R. Davidson, "An efficient planning and control framework for pruning fruit trees," in *Proc. IEEE Int. Conf. Robot. Autom*., 2020, pp. 3930–3936.

[5] D. Slaughter, D. Giles, and D. Downey, "Autonomous robotic weed control systems: A review," *Comput. Electron. Agric.*, vol. 61, no. 1, pp. 63–78, Apr. 2008.

[6] A. Abidine, B. Heidman, S. Upadhyaya, and D. Hills, "Autoguidance system operated at high speed causes almost no tomato damage," *Calif. Agric.*, vol. 58, no. 1, pp. 44–47, Jan. 2004.

[7] A. English, P. Ross, D. Ball, and P. Corke, "Vision based guidance for robot navigation in agriculture," in *Proc. IEEE Int. Conf. Robot. Autom*., May 2014, pp. 1693–1698.

[8] C. McCool *et al.*, "Efficacy of mechanical weeding tools: A study into alternative weed management strategies enabled by robotics," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 1184–1190, Feb. 2018.

[9] A. Michaels, S. Haug, and A. Albert, "Vision-based high-speed manipulation for robotic ultra-precise weed control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst*., 2015, pp. 5498–5505.

[10] C. Melita, G. Muscato, and M. Poncelet, "A simulation environment for an augmented global navigation satellite system assisted autonomous robotic lawn-mower," *J. Intell. Robot. Syst.*, vol. 71, pp. 127–142, Aug. 2013.

[11] W. S. Lee, D. Slaughter, and D. Giles, "Robotic weed control system for tomatos," *Precis. Agric.*, vol. 1, pp. 95–113, Jan. 1999.

[12] O. Bawden *et al.*, "Robot for weed species plant-specific management," *J. Field Robot.*, vol. 34, pp. 1179–1199, Jun. 2017.

[13] T. Burks, S. Shearer, and F. Payne, "Classification of weed species using color texture features and discriminant analysis," *Trans. Am Soc. Agric. Eng.*, vol. 43, no. 2, pp. 441–448, Mar. 2000.

[14] P. Herrera, J. Dorado, and A. Ribeiro, "A novel approach for weed type classification based on shape descriptors and a fuzzy decision-making method," *IEEE Sensors J.*, vol. 14, no. 8, pp. 15304–15324, Aug. 2014.

[15] A. Ferreira, D. Freitas, G. Silva, H. Pistori, and M. Folhes, "Weed detection in soybean crops using convnets," *Comput. Electron. Agric.*, vol. 143, pp. 314–324, Dec. 2017.

[16] M. Dyrmann, S. Skovsen, M. Laursen, and R. Jørgensen, "Using a fully convolutional neural network for detecting locations of weeds in images from cereal fields," in *Proc. Int. Conf. Precis. Agric*., 2018, pp. 1–7.

[17] J. Yu, A. Schumann, Z. Cao, S. Sharpe, and N. Boyd, "Weed detection in perennial ryegrass with deep learning convolutional neural network,"*front. Plant Sci.*, vol. 10, Art. no. 1422, pp. 1–9, Oct. 2019.

[18] M. H. Asad and A. Bais, "Weed detection in canola fields using maximum likelihood classification and deep convolutional neural network," *Inf. Process. Agric.*, vol. 7, no. 4, pp. 535–545, Dec. 2019.

[19] O. Barrero, D. Rojas, C. Gonzalez, and S. Perdomo, "Weed detection in rice fields using aerial images and neural networks," in *Proc. Symp. Signal Process., Image Artif. Vis.*, Aug. 2016, pp. 1–4.

[20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*., 2016, pp. 779–788.

[21] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2016, vol. 9905, pp. 21–37.

[22] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. conf. Comput. Vis*., 2018, pp. 801–818.

[23] Y. Wang, Y. Xu, S. Tsogkas, X. Bai, S. J. Dickinson, and K. Siddiqi, "Deepflux for skeletons in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*., 2019, pp. 5287–5296.

[24] K. Zhao, W. Shen, S. Gao, D. Li, and M.-M. Cheng, "Hi-fi: Hierarchical feature integration for skeleton detection," in *Proc. Int. Joint Conf. Artif. Intell.,* 2018, pp. 1191–1197.

[25] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*., 2017, pp. 2107–2116.

[26] J. Gao, A. French, M. Pound, L. He, T. Pridmore, and J. Pieters, "Deep convolutional neural networks for image-based convolvulus sepium detection in sugar beet fields," *Plant Methods*, vol. 16, no. 29, pp. 1–12 Mar. 2020.

[27] Y. Toda *et al.*, "Training instance segmentation neural network with synthetic datasets for crop seed phenotyping," *Commun. Biol.*, vol. 3, Art. no. 173, pp. 1–12, Apr. 2020.

[28] J. R. Bergado, C. Persello, and C. Gevaert, "A deep learning approach to the classification of sub-decimetre resolution aerial images," in *Proc. IEEE Int. Geosci. Remote Sens. Symp*., Jul. 2016, pp. 1516–1519.

[29] H. Caesar, J. Uijlings, and V. Ferrari, "Coco-stuff: Thing and stuff classes in context," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*., 2018, pp. 1209–1218.

[30] M. Ashikhmin, "Synthesizing natural textures," in *Proc. ACM Symp. Interact. 3D Graph*., 2001, pp. 217–226.

[31] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*., 2017, pp. 2117–2125.

[32] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron 2," https://github.com/facebookresearch/detectron2, 2019.

[33] S. Xie, C. Hu, M. Bagavathiannan, and D. Song, "Toward robotic weed control: Detection ofnutsedge weed in bermudagrass turf usinginaccurate and insufficient training data: Nutsedge dataset," 2021. [Online]. Available: http://telerobot.cs.tamu.edu/weed-detection/

[34] K. Wada, "labelme: Image polygonal annotation with python," 2016. [Online]. Available: https://github.com/wkentaro/labelme