TOWARD ROBOTIC WEED CONTROL: DETECTION OF NUTSEDGE WEED IN BERMUDAGRASS TURF USING INACCURATE AND INSUFFICIENT TRAINING DATA

A Thesis

by

SHUANGYU XIE

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Dezhen Song
Committee Members,	Guni Sharon
	Ivan Damnjanovic
Head of Department,	Duncan M. Walker

May 2021

Major Subject: Computer Science and Engineering

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, it is expensive and error-prone to perform manual data labeling. 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 have implemented the proposed algorithm and compare it with Faster R-CNN, a typical object detection approach. The results show that our design can effectively reduce the impact of imprecise and insufficient training sample issues and significantly outperforms the counterpart with a false negative rate of 0.4%, a satisfying result for weed control applications.

ACKNOWLEDGMENTS

Throughout the writing of this thesis I have received a great deal of support and assistance.

First of all, I'm extremely grateful to my supervisor, Professor Dezhen Song, whose expertise was invaluable in formulating the research questions and methodology. Furthermore, I would like to extend my deepest gratitude to my collaborators Chengsong Hu and Muthukumar Bagavathiannan who contribute many valuable ideas. I would also like thank to my committee members, Ivan Damnjanovic and Sharon Guni for providing important feedbacks. Finally, I must express my sincere thanks to my team members Shu-hao Yeh, Di Wang, Aaron Angert, Aaron Kingery, Jasmine Cheng for the inspiring discussions and kindly supports.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis committee consisting of Professor Dezhen Song and Assistant Professor Guni Sharon of the Department of Computer Science and Engineering and Associate Professor Ivan Damnjanovic of the Department of Civil Environmental Engineering.

The algorithm design depicted in Chapter 4.1 were conducted in part by Chengsong Hu of the Department of Soil and Crop Sciences.

All other work conducted for the thesis was completed by the student independently.

Funding Sources

Graduate study was supported in part by Texas A&M University.

TABLE OF CONTENTS

Page

ABSTRACT ii

ACKNOWLEDGMENTS iii

CONTRIBUTORS AND FUNDING SOURCES iv

TABLE OF CONTENTS v

LIST OF FIGURES vii

LIST OF TABLES viii

1. INTRODUCTION 1

2. RELATED WORK 3
   - 2.1 Robotic Weed Control 3
   - 2.2 Image-Based Weed detection and Segmentation 4
   - 2.3 Using Synthesis Data 5
   - 2.4 Evaluation 6

3. PROBLEM DEFINITION 7

4. ALGORITHMS 8
   - 4.1 High Fidelity Data Augmentation 8
     - 4.1.1 Template Collection and Label Creation 9
     - 4.1.2 Nutsedge Annotation 9
     - 4.1.3 Background Synthesis 10
     - 4.1.4 Recombination of Nutsedge and Background 10
   - 4.2 Training and Network Design 11
     - 4.2.1 Probability Map and Loss Function Modification 11
     - 4.2.2 Modifying Loss Function 12
     - 4.2.3 Transfer Learning Using Data with Different Levels of Annotation 13
     - 4.2.4 Skeleton Decoder 14
   - 4.3 Semi-supervised Evaluation 14
     - 4.3.1 Consistent Metrics 14
     - 4.3.2 Inconsistent Metrics 15

5. EXPERIMENT 17

5.1 Nutsedge dataset 17
5.1.1 Raw Image Set 17
5.1.2 Synthetic Dataset 17
5.2 Component Tests 18
5.2.1 Loss Function Comparison 18
5.2.2 Improvement with Transfer Learning 19
5.2.3 Synthetic Data Generation Configuration 19
5.3 Overall Performance Comparison 21
5.3.1 Algorithms and Training Setup 21
5.3.2 Metrics and Results 22

6. CONCLUSION AND FUTURE WORK 24

REFERENCES 25

LIST OF FIGURES

FIGURE Page

1.1 An overview of nutsedge detection algorithms. 1

2.1 Dimension of annotation difficulty and localization level for different methods. 5

4.1 An overview of the image synthesis pipeline. 1,2,3 and 4 represents template selection, annotating, synthesising and recombination. 8

4.2 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. 11

5.1 Samples of synthetic data. 18

5.2 A comparison of detection result with cross entropy (in green) and KL-divergence (in red) models. The grey boxes are bounding boxes from manual labelling. It is clear that there are a lot more red pixels than green pixels which means using KL-divergence loss function miss less than using cross entropy loss function. Both models are use R101 as backbone. 18

5.3 Affect of different number of nutsedge and background templates in generating synthetic training data. 20

5.4 Nutsedge on bermudagrass. 23

LIST OF TABLES

TABLE Page

5.1 Detection comparison. 20

5.2 Overall performance comparison.. 21

1. 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 turf grass 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 in a turfgrass background is difficult. Hence the manual data labeling process is expensive and error-prone. The resulting imprecise and insufficient training data significantly impact the performance of common data-driven deep learning approaches.

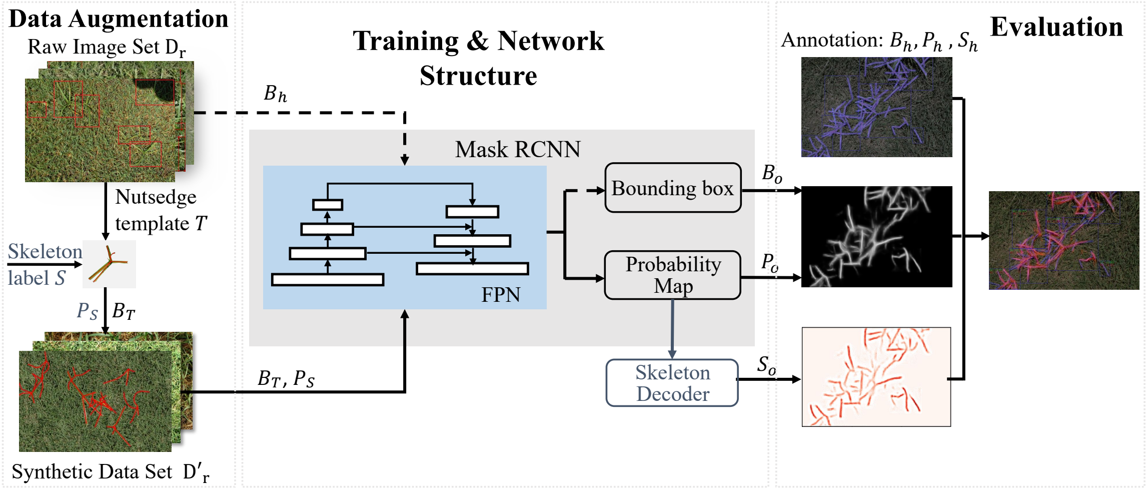

Figure 1.1: An overview of nutsedge detection algorithms.

Fig. 1.1 illustrates how we handle the challenge. First, we propose a data augmentation approach. This also 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 requirement. More over, we propose a nutsedge skeleton-based probabilistic map (NSPM) representation. NSPM (e.g. P_S in Fig. 1.1) gives more weight 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 compare it with with typical object detection method such as Faster R-CNN [2]. The experimental results have shown that our algorithm significantly outperforms the counterpart. 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%.

2. RELATED WORK

In this chapter, we discuss some recent works on weeding robot, weed detection, general models in detection and segmentation, data synthesis, and metrics designed for evaluation.

2.1 Robotic Weed Control

In recently years, applying autonomous robots in precision agriculture has raise people's attention because the nature of robot helps in reduce operating costs and dependency on labor ([3–6]).

The general architecture for weed control robot incorporates three basic components: a sensing system to detect weeds, a decision-making unit to process the information from the sensing system and to make manipulation decisions, and actuators to act accordingly [7]. Selection of the actuating (weed-killing) method is one critical area requiring progress. Actuating methods that have been extensively evaluated include cultivation tools [8–10], heat (i.e. application of flame and hot oil) [11], abrasion using a stream of particles [12], stamping [13], mowing [14] [15] , and precise herbicide application [16–20]. In addition to the actuator development, modular robotic platforms that are able to carry various weeding actuators are under active development [21–23].

In addition to the actuator development, weed sensing technology is another area requiring intensive development. Two weed sensing approaches, the first localizing and avoiding crop rows, and the other identifying individual weed plants, are under development. Crop row localization has achieved a high level of accuracy and some commercial application. English et al. [24] developed a computer vison method that tracks the direction and lateral offset of the dominant parallel texture, and achieved a standard deviation of 3.4 cm for wheat. Similar accuracy has been achieved by GPS guidance. Abidine et al. [25] demonstrated that inter-row cultivation guided by real-time kinematic (RTK) GPS at 11 km per hour could operate as close as 5 cm to the plant without causing damage. In contrast, methods for identifying individual weed plants are not yet ready for full-scale commercial application.

2.2 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 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 [7]. For example, Burks et al. [26] utilize the color co-occurrence method to discriminate textures between five common weed species. Herrera et al. [27] 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) outcompetes traditional computer vision methods in feature extraction and becomes more popular for weed detection nowadays. Many recent works employ CNNs to detect weeds in various crops, including soybean [28], cereal [29], ryegrass [30], canola [31] and rice [32]. These methods receive satisfying results in distinguishing the weed from highly color contrasted background. However, when the background is turf grass, the weed detection problem become more challenging and we are developing new methods here to improve detection performance.

With increasing capability of detection networks such as Faster R-CNN [2], YOLO [33], and SSD [34], object detection on highly-similar background can be achieved. However, these object detection networks only provide bounding box output which is not sufficient for further field operation, especially localization. The localization problem can be partially addressed by segmentation networks such as Mask R-CNN [1] and Deeplab [35], because these networks achieve finer segment result for objects of interest. The problem for such methods is the tremendously annotation cost, i.e. these networks often require pixel-wise precise ground truth for training, which is difficult and expensive in weed detection problem.

Considering the shape of nutsedge, 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 [36] and Hi-Fi [37]. Although these methods only target single object detection, which are not directly applicable in our scenarios, this inspires our development of nutsedge skeleton probably map to balance between the localization and annotation cost (Fig. 2.1).

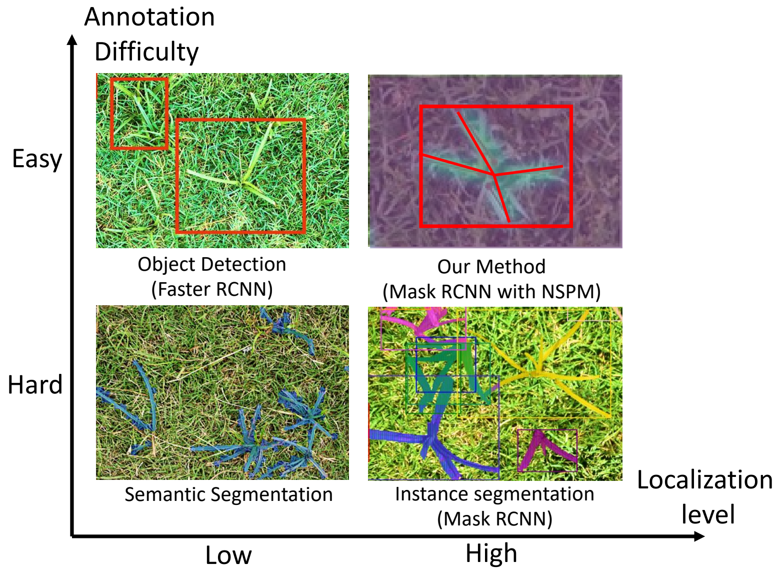

Figure 2.1: Dimension of annotation difficulty and localization level for different methods.

2.3 Using Synthesis Data

Researchers have explored different methods for data augmentation to enhance neural network training result, especially in domains where annotated data is difficult to obtain or expensive to annotate. Generative adversarial networks (GAN) is one of the method and is quite popular these days [38]. 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 need for nutsedge detection.

Image synthesis using real object segments is a straight forward but powerful way to augment image dataset. A common pipeline involves a segmentation stage where nutsedge objects are extracted from background either manually or automatically, and a synthesis stage where extracted foreground objects are pasted to the background. Using this approach, Gao [39] trained a YOLOv3 model for weed and crop detection, and achieved a mean average precision at 0.829. Toda [40] showed that a Mask R-CNN model for barley seed morphology phenotyping can be trained purely by a synthetically generated dataset. 96% recall and 95% average Precision against real test dataset was achieved.

In addition to the above direct method, image synthesis from 3D model has also been explored. Barth [41] proposed a method to procedurally generate renders of 3D plant models based on empirical measurements, using bell pepper as a running example. The method is able to generate a large number of images with per-pixel class and depth annotation. The authors showed high similarity between synthetic images and empirical images qualitatively and quantitatively. The advantage of 3D approach over 2D method is obvious as parameters can be easily change during rendering to generate datasets under a broad set of conditions, such as different light conditions, perspectives or camera types. However, 3D modelling can be time consuming itself.

2.4 Evaluation

For general image segmentation task, one of the popular evaluation metrics is calculate the Intersect over Union(IoU) for the segment result and the ground truth. This metric is proposed by [42]. Most of image segmentation algorithm in general task, for example, SegNet, Mask RCNN, DeepLab, are using IoU as evaluation metric. IoU is pretty straight forward and easy to compute because it only consider the pixel-level match between output and ground truth. Besides the IoU, there are some other metrics depict the contour alignment [43] and global match [44] between segment result and ground truth. However, these evaluation metrics only consider the human annotation as the well approximation of ground truth, ignoring the fact that human can produce wrong annotation. Besides, due to the eye fatigue, human will tend to ignore the region that they cannot be determined. In our task, the background and detection object share similar shape and color, which makes human error non-negligible. To better evaluate our own task, we are proposed our own plant skeleton metric. This metric designs for our specific task and considers the uncertainty exist in the segmentation and validate process.

3. PROBLEM DEFINITION

Our robot observes field through a downward facing camera to collect images. Therefore, all images are collected from a perspective that is perpendicularly facing the ground from the same distance (0.5m 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 $\mathbf{B}_h$ represents 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 $\mathbf{S}_h$ represents as human labeled skeleton set and $\mathbf{S}_o$ as 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.*

4. ALGORITHMS

Our algorithmic development consists of three major components: data augmentation, network design & training, and evaluation (Fig. 1.1). Data augmentation addresses the issue of insufficient training data by combining synthesizing data with manually-labelled data. Due to the non-negligible level of errors exist in manually annotated labeling, the network design & training revises existing neural networks to handle the inaccurately labelled training data. For the same reason, we cannot trust the labelled data as ground truth and have to design a new evaluation pipeline considering the labeling noise to validate our model. We begin with data augmentation.

4.1 High Fidelity Data Augmentation

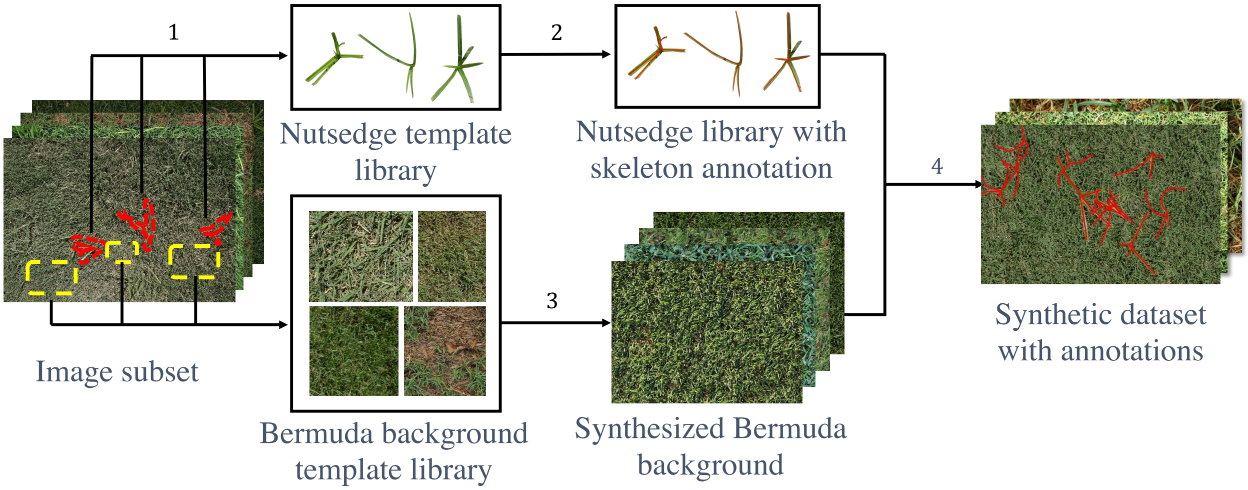
1
2
Nutsedge template library
Nutsedge library with skeleton annotation
4
Synthetic dataset with annotations
Image subset
3
Bermuda background template library
Synthesized Bermuda background

Figure 4.1: An overview of the image synthesis pipeline. 1,2,3 and 4 represents template selection, annotating, synthesising and recombination.

As detailed later, we employ deep neural networks for weed recognition which often require manually labeled data as the ground truth for training. To an untrained eye, nutsedge weeds look similar to background turf grass. Creating a large manually-labeled dataset is cost-prohibitive and time consuming. The process also inevitably contains non-negligible levels of error. The insufficient and inaccurate training data pose a big challenge to deep learning methods.

We develop an image synthesis algorithm to efficiently generate high-fidelity artificial dataset from existing dataset. This data augmentation by using image synthesis has three benefits: 1) it expands the size of training dataset 2) it provides precise pixel-level labels, and 3) it requires a minimal human labeling effort.

In the synthetic dataset, each image is composed by nutsedge foreground and Bermuda grass background. To generate realistic synthetic image, we ask human experts hand-select a small number of nutsedge templates and background patches from a raw image set as a material library. For training purpose, the nutsedge templates are annotated. The algorithm consists of the following four steps corresponding to steps 1-4 in Fig. 4.1.

4.1.1 Template Collection and Label Creation

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 [45] based on the lighting condition to build an image subset (5% of the training set) with images under different light conditions proportional to raw image set. Human experts segment out nutsedge template $T \subset I_r$ and nutsedge-free turf background pixel patches from the sampled image set.

4.1.2 Nutsedge Annotation

There are 3 different annotations: bounding box B_T, binary mask M_s, and plant skeleton S for each nutsedge template T. 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

$$B_T := \{(u,v)|u \in [u_{\text{left}}, u_{\text{right}}], v \in [v_{\text{bottom}}, v_{\text{top}}], (u,v) \in I_r\}, \tag{4.1}$$

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

We propose plant skeleton label S to better describe the structure attribute of nutsedges and use it in our network design. As illustrated in Fig. 4.2(a), S models each nutsedge plants as a cluster of line segments where each line segment depicts the center of a leave,

$$S := \{\mathbf{l}_k : k = 1, ..., k_{\text{max}}\}, \tag{4.2}$$

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 annotation process, one skeleton is corresponded to one bounding box. All labels can be generated automatically.

4.1.3 Background Synthesis

To generate realistic background image with appropriate size and scale, a natural texture synthesis algorithm [46] 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.

4.1.4 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 light condition inconsistency. The skeleton annotations of each nutsedge template are also inserted during the image synthesis process.

Comparing to the manual labeling effort for a large dataset with pixel level annotations using human experts (more than 5 minute per image) for supervised learning, our image synthesis and skeleton model method significantly reduces labeling cost.

4.2 Training and Network Design

With both synthesized data and human-annotated training data (i.e. all raw image training set comes with human-labelled bounding boxes), we employ Mask R-CNN [1] to develop our detector. However, we need to modify the neural network to handle the imprecision in labeling in the training samples.

4.2.1 Probability Map and Loss Function Modification

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.

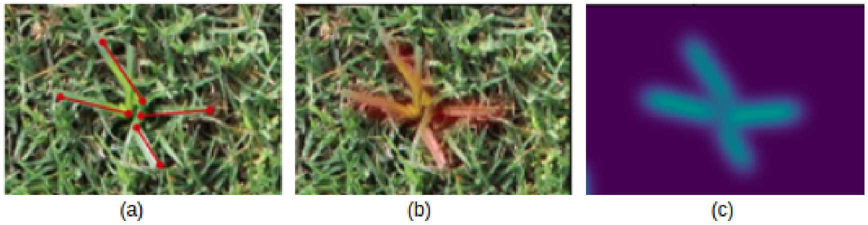

Figure 4.2: An example of NSPM. (a) skeleton from the data synthesis, (b) pixels masked as nutsedge in the synthesized image, and (c) the resulting nutsedge skeleton probability model.

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 vein of the nutsedge is more important than detecting its edges for weed detection applications. This motivates us to propose to use NSPM input. The purpose is to instruct Mask R-CNN to differentiate the central leaf vain part of the nutsedge while reducing the impact of imprecision in nutsedge boundary segmentation.

Fig. 4.2 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 (4.1) 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

$$p_S(\mathbf{x}_{uv}) \propto \begin{cases} \sum_{k=1}^{k_{\max}} \frac{1}{\sigma\sqrt{2\pi}} \exp\{-\frac{1}{2}[\frac{d((u,v),\mathbf{l}_k)}{\sigma}]^2\}, & \text{if}((u,v) \in B_h) \wedge (B_h \in \mathbf{B}_h) \\ 0, & \text{otherwise.} \end{cases} \tag{4.3}$$

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 to estimate the proper value of σ. By drawing the histogram for each nutsedge template with d 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

$$P_S(u,v) = \{p_S(\mathbf{x}_{uv}), \forall (u,v) \in I\} \tag{4.4}$$

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

4.2.2 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 $p_2(\mathbf{x}_{uv}) \in [0,1]$

$$L_H(p_1, p_2) = -\sum_{(u,v)\in B_o} p_1(\mathbf{x}_{uv}) \log(p_2(\mathbf{x}_{uv})). \tag{4.5}$$

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

$$L_{KL}(p_1, p_2) = -\sum_{(u,v)\in B_o} p_1(\mathbf{x}_{uv}) \log(\frac{p_1(\mathbf{x}_{uv})}{p_2(\mathbf{x}_{uv})}). \tag{4.6}$$

4.2.3 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 boxes 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. 1.1, 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. 1.1, solid line's flow).

4.2.4 Skeleton Decoder

When we train the Mask R-CNN, we adopt ResNet-FPN [**?**] backbone to obtain feature fusing map in the feature extraction stage. With the high-resolution and high-level semantic map embedded in same feature map, the model learns complex semantic information through training. The inference output probability map P_o has a higher probability in the center line 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 [37] algorithm to the pre-processed probability map to decode its skeleton structure.

4.3 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). However, human annotations contain non-negligible errors due to high similarity between nutsedge and background grass in weed detection problem. 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.

4.3.1 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 (4.4). We can threshold P_S to obtain region set I_S according to human labels,

$$I_S := \{(u,v)|p_S(\mathbf{x}_{uv}) > t\} \subseteq I_r, \tag{4.7}$$

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,

$$r_{\text{IoU}} = \frac{|I_S \cap I_o|}{|I_S \cup I_o|}, \tag{4.8}$$

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 $d_{Sh}(u, v)$ to its closest point in S_h,

  $$d_{Sh}(u, v) = \min_{(u_a, u_b) \in S_h} \sqrt{(u_a - u)^2 + (u_b - v)^2}. \tag{4.9}$$

  If $d_{Sh}(u, v)$ 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 pixels number in S_h,

  $$C_s = \frac{|\{(u, v) | (u, v) \in S_o, d_{Sh}(u, v) \leq d\}|}{|S_h|} \tag{4.10}$$

  as the skeleton similarity metric.

4.3.2 Inconsistent Metrics

Due to the difficulty in labeling and high similarity between nutsedge and Bermuda grasses, model output and human annotations are not always consistent. It is possible the model fails to recognize a nutsedge and it is also possible 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 $\mathbf{R}_a$,

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

where r_{IoU} and C_s are computed using (4.8) and (4.10), respectively. Then the inconsistent bounding box set $\mathbf{R}_c = \{(\mathbf{B}_h \cup \mathbf{B}_o) \setminus \mathbf{R}_a\}$. When the inconsistent case is detected, we manually reexamine the labels of these bounding box and classify $\mathbf{R}_c$ into four classes: false positive case set of algorithm output $\mathbf{B}^o_{\text{FP}}$; false negative case set of algorithm output $\mathbf{B}^o_{\text{FN}}$; false negative set of human annotation $\mathbf{B}^h_{\text{FN}}$.

5. EXPERIMENT

We have implemented our weed detection algorithm based on Detectron2 [47] 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 pre-trained model on MSCOCO dataset.

5.1 Nutsedge dataset

We have two types of data: the raw image set collected from the field with manual annotations and synthetic image set with ground truth synthetic label.

5.1.1 Raw Image Set

We build a TAMU nutsedge dataset which has been collected at ScottsMiracle-Gro Facility for Lawn and Garden Research using Nikon™ D3300 or Canon EOS Rebel T7™ mounted at fixed height 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 cost. To cover the appearance variation of nutsedge, data are collected at different lighting conditions, temperature, weather, and moisture levels. To cover a typical nutsedge growth season, data have been 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 sets D_t (10%) . All data are labeled with bounding boxes for both training and testing purposes. In addition, 25% of 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" [48] tool.

5.1.2 Synthetic Dataset

Generated using method in Section 4.1, our synthetic dataset contains 4750 images with bounding box labels which are to be used as training set. The density of nutsedge is set at 5 to 10 plants per one million pixels. 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 Fig. 5.1.

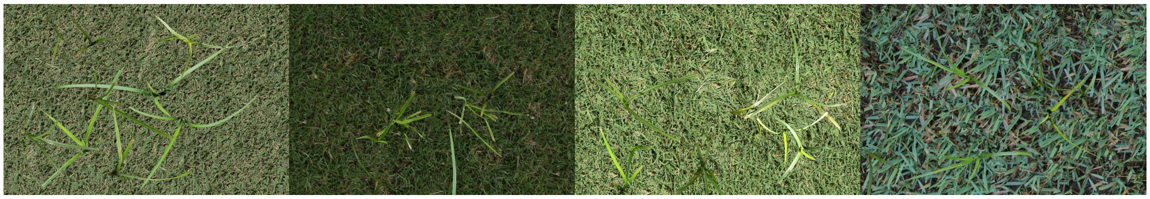

Figure 5.1: Samples of synthetic data.

5.2 Component Tests

5.2.1 Loss Function Comparison

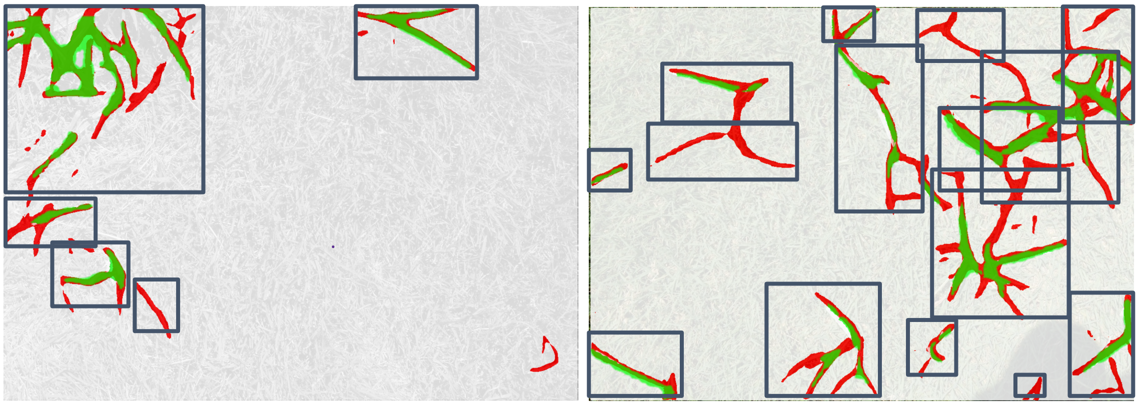

Figure 5.2: A comparison of detection result with cross entropy (in green) and KL-divergence (in red) models. The grey boxes are bounding boxes from manual labelling. It is clear that there are a lot more red pixels than green pixels which means using KL-divergence loss function miss less than using cross entropy loss function. Both models are use R101 as backbone.

We train 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

$$r_{\text{S-IoU}} = \frac{1}{n_b} \sum_{n_b} \frac{|S_h|}{c_{I_S}} r_{\text{IoU}} \text{ and } \overline{r}_{\text{IoU}} = \frac{1}{n_{t_S}} \sum_{n_{t_S}} r_{\text{S-IoU}}. \tag{5.1}$$

Similarly, we extend the skeleton similarity metric,

$$C_{Ss} = \frac{1}{n_b} \sum_{n_b} \frac{|S_h|}{c_{I_S}} C_s \text{ and } \overline{C}_s = \frac{1}{n_{t_S}} \sum_{n_{t_S}} C_{Ss}. \tag{5.2}$$

The overall result is show in Table. 5.1. We use R50, R101, CE and KL represents 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), model trained by KL loss function performs better than that of R101 using CE loss function by 3% in $\overline{r}_{\text{IoU}}$ and 4% in $\overline{C}_s$. When the backbone is identical, model with KL loss improves over the CE by over 10% in both $\overline{r}_{\text{IoU}}$ and $\overline{C}_s$. Sample results are shown in Fig. 5.2.

5.2.2 Improvement with Transfer Learning

We follow the basic rules of transfer learning by using pre-trained model to improve the performance. In general case, without the task-specialized pre-trained model, the common model such as model trained by MSCOCO is used as the pre-trained model. The first four lines in Table 5.1 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 5.1 which is highlighted in bold font as the best performance.

5.2.3 Synthetic Data Generation Configuration

Synthetic data provides accurate ground truth in pixel level mask which is expected to help improve the model. We study how the number of foreground nutsedge and background Bermuda grass templates (Section 4.1.1) in generating synthetic data affects the overall detection perfor-

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**

Table 5.1: Detection comparison.

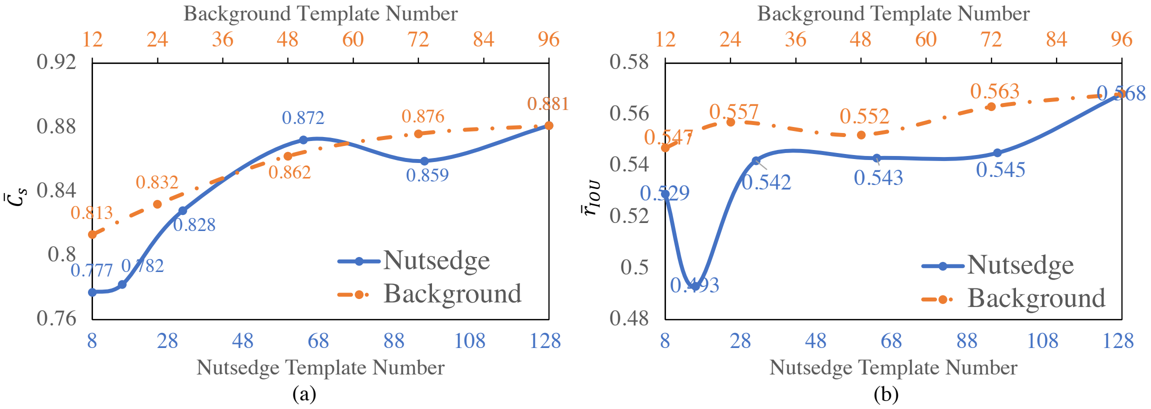

Figure 5.3: Affect of different number of nutsedge and background templates in generating synthetic training data.

mance. First, we variate nutsedge foreground template sizes while keeping the background template number to be 96. We increase the number of nutsedge template from 8 to 129. Again, $\overline{r}_{\text{IoU}}$ and $\overline{C}_s$ are used to evaluate the detection result (Fig. 5.3). With mere 8 nutsedge templates, the trained model achieves $\overline{r}_{\text{IoU}}$ of 52.9% and $\overline{C}_s$ of 77.7%. With 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 background template number from 12 to 96 while fixing number of nutsedge template to be 128. $\overline{r}_{\text{IoU}}$ and $\overline{C}_s$ are 54.7% and 81.3%. The curve in Fig. 5.3 also illustrates the positive correlation between the number of background templates and the model performance, but the trend is less significant if comparing to that of the nutsedge template number. Considering the fact that selecting templates is costly, we choose 129 nutsedge templates with 96 background templates as our setup in generating synthetic data.

5.3 Overall Performance Comparison

5.3.1 Algorithms and Training Setup

The overall evaluation compares below four algorithms (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. This vanilla algorithm serves as a baseline.

b. Mask R-CNN based model with R101 as backbone and trained by CE loss function: Here we use synthetic data with binary pixel-level mask label D_{s_b}. This algorithm test the power of synthetic data.

c. We change Alg. b settings by swapping the loss function from CE to KL divergence in (4.6). 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 model c with a pre-trained model described in Sec. 5.2.2. 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 component test.

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

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%

Table 5.2: Overall performance comparison

5.3.2 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 be 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 5.2 show 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 definitely help in training the network.

For Algs. b-c, r_d does not tell the complete story. We need to take a close look because not all nutsedge pixels are equal or error-free. Also, we are also interested if disagreement between algorithm and human label 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 (4.8) and skeleton similarity (4.10)). Cases outside $\mathbf{R}_a$ are gone through a manual re-examination to determine the which is 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.

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

Table 5.2 shows that Alg. d achieves the best overall results. This is due to high overall agreement between human and algorithm output and lowest false negative ratio. In lawn care applications, algorithms with low false negative help remove weeds more thoroughly. However, if one handles valuable horticultural crops, we may want to choose Alg. b due to its lowest false positive rate.

Fig. 5.4 shows some of the visualization of the results. First row is the original image. Second row is hand craft segmentation from the raw image with blur background. Third row is our proposed method's detection result. The nutsedge segmentation is represent by red mask and attached to the raw image.

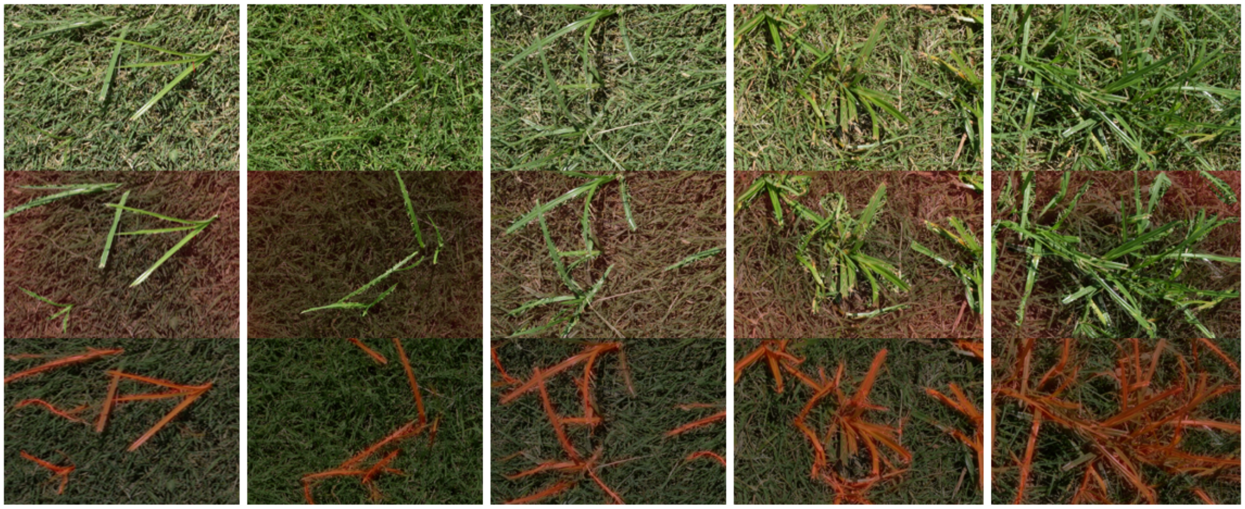

Figure 5.4: Nutsedge on bermudagrass.

6. CONCLUSION AND FUTURE WORK

We reported our detection algorithm development for robotic weed control. We focused on detecting nutsedge weed from Bermuda turf grass. Building on Mast R-CNN, a semantic segmentation framework, our new algorithm incorporated four new designs to handle the imprecise and insufficient training data issues. 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 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 result showed that our design was successful and significantly outperform Faster R-CNN approach.

In the future, we will extend our approach to more types of weeds and turfs. Building on the result, we will also develop robotic weed removal algorithms and systems.

REFERENCES

[1] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, 2017.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, pp. 1–10, 01 2016.

[3] T. C. Thayer, S. Vougioukas, K. Goldberg, and S. Carpin, "Multirobot routing algorithms for robots operating in vineyards," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 3, pp. 1184–1194, 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 *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3930–3936, 2020.

[5] W. McAllister, D. Osipychev, A. Davis, and G. Chowdhary, "Agbots: Weeding a field with a team of autonomous robots," *Computers and Electronics in Agriculture*, vol. 163, p. 104827, 08 2019.

[6] H. Williams, M. Nejati, S. Hussein, N. Penhall, J. Y. Lim, M. H. Jones, J. Bell, H. S. Ahn, S. Bradley, P. Schaare, P. Martinsen, M. Alomar, P. Patel, M. Seabright, M. Duke, A. Scarfe, and B. MacDonald, "Autonomous pollination of individual kiwifruit flowers: Toward a robotic kiwifruit pollinator," *Journal of Field Robotics*, Jan. 2019.

[7] D. Slaughter, D. Giles, and D. Downey, "Autonomous robotic weed control systems: A review," *Computers and Electronics in Agriculture*, vol. 61, no. 1, pp. 63 – 78, 2008.

[8] B. Melander, B. Lattanzi, and E. Pannacci, "Intelligent versus non-intelligent mechanical intra-row weed control in transplanted onion and cabbage," *Crop Protection*, vol. 72, pp. 1 – 8, 2015.

[9] S. A. Fennimore, R. F. Smith, L. Tourte, M. LeStrange, and J. S. Rachuy, “Evaluation and economics of a rotating cultivator in bok choy, celery, lettuce, and radicchio,” *Weed Technology*, vol. 28, no. 1, p. 176–188, 2014.

[10] C. McCool, J. Beattie, J. Firn, C. Lehnert, J. Kulk, O. Bawden, R. Russell, and T. Perez, “Efficacy of mechanical weeding tools: A study into alternative weed management strategies enabled by robotics,” *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1184–1190, 2018.

[11] Y. Zhang, E. S. Staab, D. C. Slaughter, D. K. Giles, and D. Downey, “Automated weed control in organic row crops using hyperspectral species identification and thermal micro-dosing,” *Crop Protection*, vol. 41, pp. 96 – 105, 2012.

[12] F. Forcella, “Air-propelled abrasive grit for postemergence in-row weed control in field corn,” *Weed Technology*, vol. 26, no. 1, p. 161–164, 2012.

[13] A. Michaels, S. Haug, and A. Albert, “Vision-based high-speed manipulation for robotic ultra-precise weed control,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5498–5505, 2015.

[14] C. Melita, G. Muscato, and M. Poncelet, “A simulation environment for an augmented global navigation satellite system assisted autonomous robotic lawn-mower,” *Journal of Intelligent Robotic Systems*, vol. 71, 08 2013.

[15] H. Have, J. D. Nielsen, B. Blackmore, and F. Theilby, “Development and test of an autonomous christmas tree weeder,” *Precision Agriculture 2005, ECPA 2005*, pp. 629–635, 01 2005.

[16] W. S. Lee, D. Slaughter, and D. Giles, “Robotic weed control system for tomatos,” *Precision Agriculture*, vol. 1, pp. 95–113, 01 1999.

[17] R. Lamm, D. Slaughter, and D. Giles, “Precision weed control system for cotton,” *Transaction of the American Society of Agricultural and Biological Engineers*, vol. 45, pp. 231–238, 01 2002.

[18] H. Søgaard, I. Lund, and E. Graglia, “Real-time application of herbicides in seed lines by computer vision and micro-spray system,” in *American Society of Agricultural and Biological Engineers(ASAE) Annual Meeting*, 2006.

[19] A. Nieuwenhuizen, J. Hofstee, and E. Van Henten, “Performance evaluation of an automated detection and control system for volunteer potatoes in sugar beet fields,” *Biosystems Engineering*, vol. 107, pp. 46–53, 2010.

[20] H. Midtiby, S. Mathiassen, K. Andersson, and R. Jørgensen, “Performance evaluation of a crop/weed discriminating microsprayer,” *Computers and Electronics in Agriculture*, vol. 77, pp. 35–40, 06 2011.

[21] O. Bawden, J. Kulk, R. Russell, C. Mccool, A. English, F. Dayoub, T. Perez, and C. Lehnert, “Robot for weed species plant-specific management,” *Journal of Field Robotics*, vol. 34, 06 2017.

[22] J. Underwood, M. Calleija, Z. Taylor, C. Hung, J. Nieto, R. Fitch, and S. Sukkarieh, “Real-time target detection and steerable spray for vegetable crops,” in *Proceedings of the International Conference on Robotics and Automation: Robotics in Agriculture Workshop, Seattle, WA, USA*, 2015.

[23] L. Grimstad and P. From, “The Thorvald II agricultural robotic system,” *Robotics*, vol. 6, 09 2017.

[24] A. English, P. Ross, D. Ball, and P. Corke, “Vision based guidance for robot navigation in agriculture,” in *IEEE International Conference on Robotics and Automation*, pp. 1693–1698, 05 2014.

[25] A. Abidine, B. Heidman, S. Upadhyaya, and D. Hills, “Autoguidance system operated at high speed causes almost no tomato damage,” *California Agriculture*, vol. 58, pp. 44–47, 01 2004.

[26] T. Burks, S. Shearer, and F. Payne, “Classification of weed species using color texture features and discriminant analysis,” *Transactions of the American Society of Agricultural Engineers*, vol. 43, pp. 441–448, 03 2000.

[27] P. Herrera, J. Dorado, and Ribeiro, “A novel approach for weed type classification based on shape descriptors and a fuzzy decision-making method,” *Sensors*, vol. 14, p. 15304–15324, Aug 2014.

[28] A. Ferreira, D. Freitas, G. Silva, H. Pistori, and M. Folhes, “Weed detection in soybean crops using convnets,” *Computers and Electronics in Agriculture*, vol. 143, pp. 314–324, 12 2017.

[29] 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 *International Conference on Precision Agriculture. International Society of Precision Agriculture*, 2018.

[30] J. Yu, A. Schumann, Z. Cao, S. Sharpe, and N. Boyd, “Weed detection in perennial ryegrass with deep learning convolutional neural network,” *Frontiers in Plant Science*, vol. 10, p. 1422, 10 2019.

[31] M. H. Asad and A. Bais, “Weed detection in canola fields using maximum likelihood classification and deep convolutional neural network,” *Information Processing in Agriculture*, vol. 7, 12 2019.

[32] O. Barrero, D. Rojas, C. Gonzalez, and S. Perdomo, “Weed detection in rice fields using aerial images and neural networks,” in *2016 XXI Symposium on Signal Processing, Images and Artificial Vision (STSIVA)*, pp. 1–4, 2016.

[33] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, real-time object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 06 2016.

[34] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “SSD: Single shot multibox detector,” in *European Conference on Computer Vision*, p. 21–37, Springer International Publishing, 2016.

[35] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in *European Conference on Computer Vision*, pp. 833–851, 2018.

[36] Y. Wang, Y. Xu, S. Tsogkas, X. Bai, S. J. Dickinson, and K. Siddiqi, "Deepflux for skeletons in the wild," *CoRR*, vol. abs/1811.12608, 2018.

[37] K. Zhao, W. Shen, S. Gao, D. Li, and M.-M. Cheng, "Hi-Fi: Hierarchical feature integration for skeleton detection," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pp. 1191–1197, 7 2018.

[38] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2242–2251, 07 2017.

[39] 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, 03 2020.

[40] Y. Toda, F. Okura, J. Ito, S. Okada, T. Kinoshita, H. Tsuji, and D. Saisho, "Training instance segmentation neural network with synthetic datasets for crop seed phenotyping," *Communications Biology*, vol. 3, p. 173, 04 2020.

[41] R. Barth, J. IJsselmuiden, J. Hemming, and E. van Henten, "Data synthesis methods for semantic segmentation in agriculture: A capsicum annuum dataset," *Computers and Electronics in Agriculture*, vol. 144, pp. 284–296, Jan. 2018.

[42] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, pp. 303–338, 06 2010.

[43] G. Csurka, D. Larlus, F. Perronnin, and F. Meylan, "What is a good evaluation measure for semantic segmentation?.," in *BMVC*, vol. 27, pp. 10–5244, 2013.

[44] D.-P. Fan, C. Gong, Y. Cao, B. Ren, M.-M. Cheng, and A. Borji, "Enhanced-alignment measure for binary foreground map evaluation," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pp. 698–704, 7 2018.

[45] J. R. Bergado, C. Persello, and C. Gevaert, "A deep learning approach to the classification of sub-decimetre resolution aerial images," in *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 1516–1519, 2016.

[46] M. Ashikhmin, "Synthesizing natural textures," in *Proceedings of the 2001 Symposium on Interactive 3D Graphics*, I3D '01, (New York, NY, USA), p. 217–226, Association for Computing Machinery, 2001.

[47] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2." `https://github.com/facebookresearch/detectron2`, 2019.

[48] K. Wada, "labelme: Image Polygonal Annotation with Python." `https://github.com/wkentaro/labelme`, 2016.