Accurate Weed Detection with Semi-Supervised Learning

Yingxue Wang (yingxuew@umich.edu)
Supervisor: Prof. Dmitry Berenson
University of Michigan
April 30, 2021

***Abstract*—Accurate weed detection is essential in robotic weed control applications such as selective spraying and weed electrification. CNN-based deep learning methods have proven to be accurate in classifying weeds at pixel-level by producing vegetation masks. However, these methods are heavily reliant on the quantity of available training data and quality of annotations which cannot be used reliably for weed detection because there is no suitably large dataset that exists for this task. We propose a semi-supervised learning architecture that alleviates the data deficiency issue by introducing a self-supervised training stage that adapts the BYOL architecture to a standard R-CNN based detection pipeline. Through learning from a large number of unannotated images, the network is able to produce better feature representations of the plants, leading to more accurate vegetation masks and bounding box predictions. We present two mask-prediction methods: the first one directly generates instance-mask predictions from a semantic segmentation network, e.g. Mask R-CNN, and the second one computes the instance-masks using bounding box prediction outputs from an object detection network, e.g. Faster R-CNN. The geometric centroid of a mask is computed as the grasp-point and the information is passed to a pair of robotic arms for weed-pulling.**

***Index Terms*—Self-supervised Learning, weed detection, R-CNN, robotic weed control**

I. Introduction

Robotic weed control provides a promising future in agriculture by largely increasing productivity and reducing human-labor cost. The methods of robotic weed control can be grouped into two categories: airborne remote sensing and ground-based techniques. For example, [1] uses unmanned aerial vehicles (UAVs) to detect weeds in soybean fields and perform selected spraying, whereas ground-based techniques [2] commonly involve a ground-robot to perform the sensing and weed control tasks. As an on-going project at the Autonomous Robotic Manipulation Lab of the University of Michigan, a robot is designed to autonomously navigate in a roof-top garden, perform weed detection using captured image data, and direct a pair of robotic arms to pull the weeds. The Robot gardening project is aimed towards miniaturized version of the large-scale theme of weed control in farming.

Research in this application usually focuses on four major areas: detection, mapping, guidance, and control [3]. Despite various advancements being made, the biggest challenge remains on how to distinguish crop species from its natural growth environment during the detection stage. Recent learning-based approaches, especially those based on Convolutional Neural Network(CNN) show promising outcomes to accurately identify crop and weed pixels in a given image and produce a vegetation mask [3] [4]. However, these approaches are usually data-dependent and thus are not robust to varying lighting conditions. They were also incapable of distinguishing object instances due to the complex overlapping poses of crop and weed plants [5].

Moreover, training of deep-learning networks require large-scale datasets, and the performance of such models is reliant on the quality of annotations of the datasets [6]. The literature boasts many relatively large-scale weed and plant life image datasets [7] [5]; however, they are meant for classification tasks-only and neither provide instance-level plant bounding boxes nor contain segmentation masks. There does exist weed/crop datasets that provide the desired labels, such as the Crop Weed Field Image Dataset(CWFID) [8], but it only contains 60 images with merely a few hundred object instances thus cannot provide enough diversity of data to ensure generalizability of the trained model. Given that manual labeling of such images is very expensive, data deficiency brings a challenge in accurately detecting weeds. There is no prior work that deal with this issue in automatic weed detection applications, but recent advances in self-supervised learning techniques provide a possibly feasible solution to it.

Self-supervised Learning refers to learning methods that are explicitly trained with automatically generated labels, in other words, they are trained without ground-

truth annotations. Whereas semi-supervised Learning refers to learning methods that use labeled data in conjunction with unlabeled data [9]. Both methods are suitable to be applied to scenarios where sufficient labeled data is not available.

In this project, we develop a computer vision pipeline for grasp-center detection from image data to inform location for the robotic arms to pull the weeds, building on the preliminary work conducted by Zhu et al. [10]. The grasp-centers of the weeds are calculated from their vegetation masks. The vegetation masks can be obtained either directly from the outputs of a semantic segmentation model (e.g. Mask R-CNN), or through post-processing predicted bounding boxes that produced by an object detection model (e.g. Faster R-CNN). To improve the accuracy of detection and deal with the data deficiency challenge, we propose a semi-supervised learning architecture, where a deep-learning network is first pre-trained on a large-scale, unannotated image dataset via self-supervised learning and then fine-tuned using transfer learning on fully annotated dataset. By learning from more image data through self-supervised learning, the model is able to learn a better representation of the plant features, thus obtain better detection results.

The paper is structured as follows. It briefly introduces the background of semantic segmentation and object detection using deep learning methods in Section II. Section III first introduces the baselines for detection and segmentation task and then explains how the self-supervised learning method, BYOL [11] was trained and integrated into the detection pipeline so the overall architecture forms a semi-supervised learning problem. We propose two slightly different pipelines to obtain vegetation mask: 1) BYOL + Mask R-CNN, 2) BYOL + Faster R-CNN + Otsu-based thresholding. The performance of the semi-supervised learning pipelines are evaluated by performing an ablation study with different datasets and the results are discussed in Section IV. Finally, conclusion and possible future extensions to the work are mentioned in Sections V,VI.

II. Related Work

Traditional weed detection methods are built on hand-crafted features and shallow trainable architectures. Their performance easily stagnates by constructing complex ensembles which combine multiple low-level image features with high-level context from object detectors and scene classifiers [6]. One method is based on support vector machines (SVM) using histograms of oriented gradients (HOG) as feature descriptor [12], and the vegetation masks are generated with the NDVI [13] and the Otsu-based methods [14]. Such methods have lower detection accuracy compared to the recent CNN-based deep learning approaches, but provide a reliable mask-generation solution using image-processing techniques.

Modern learning-based approaches usually involve training a network architecture with many data-label pairs to allow the network directly produce plant categories, bounding boxes, and vegetation masks from images during the evaluation stage. A typical segmentation architecture for weed detection is to use a Region-CNN (R-CNN) to produce vegetation masks [10] [15] [16]. R-CNN [17] models involves a two-step process where an attention mechanism that mimic human brains to some extent produces a coarse scan of the whole image and produce a list of "Region of Interests" (RoI), which are then further processed by sub-networks to achieve downstream tasks such as classification and mask-generation. Further, the combination of R-CNN with deep residual learning [18], i.e. the ResNet backbone has became a standard practice in object detection tasks, an example of which is Faster R-CNN [19]. Mask R-CNN [20] extends this methodology further to efficiently detect objects while simultaneously generating high quality segmentation masks for each instance by adding a branch for mask prediction in parallel with the existing branch for bounding box recognition. Our weed detection pipeline builds on Faster R-CNN and Mask R-CNN.

Self-supervised learning approaches are being actively studied in the field computer vision. Among various proposed methods, contrastive learning approach [21] [22] [23] [24] outperforms others. Contrastive learning is a discriminative method which, in the image-related tasks, tries to bring the representation of different views of the same image closer ("positive pairs"), and separate the different views from different images ("negative pairs") apart [25]. However, such methods often require many negative examples, which is not suitable when there is already not sufficient training data in the weed detection task. Grill et al. [11] claimed that negative pairs were not necessary, and proposed an architecture, BYOL, that achieved new state-of-the-art without them. The success of BYOL relies on multiple factors, like predictor networks, stop-gradients, exponential moving averages, and weight decay to effectively avoid representational collapse [26] when missing negative data. We integrate the BYOL architecture to our weed-detection pipeline to allow the detection backbone model to learn a better representation of plant features.

III. Method

In this section, we start by giving an overview of the computer vision pipeline for determining grasp-centers

by presenting the baseline Mask R-CNN and Faster R-CNN architectures. Then, we introduce BYOL architecture and explain how it is incorporated into the training pipeline. The details for masks generation for object detection models and how grasp-centers are computed are also illustrated.

A. Baseline Architecture

The baseline prediction pipeline follows the standard implementation of Mask R-CNN, where instance segmentation masks were directly predicted as network outputs to provide pixel-level classification results. During the training stage, training data is fed into a deep neural backbone to generate feature maps, followed by a Feature Pyramid Network (FPN) to generate regional proposals. A branch of Fully Convolutional Network is used to perform mask-prediction and, in parallel, another branch of neural network is used to perform classification and bounding box regression tasks. Each of the sub-networks is referred as an "RoI head". The predictions were then compared with ground-truth labels by calculating a cross-entropy loss, the gradient of which is then backpropagated for network parameter update. During the test/evaluate stage, one or more images are fed into the network to obtain predictions. For each mask prediction, the grasp point for the robot arm is calculated.

Without the mask-prediction RoI head, the architecture is a standard Faster R-CNN architecture. The training stage and test stage are very similar to the Mask R-CNN architecture, except there are only class prediction and bounding box predictions available. Therefore, the mask has to be generated using one of the aforementioned traditional image processing techniques such as Otsu-based thresholding for each bounding box prediction. The method that we adopted will be described in detail in section III-C. The grasp point can then be calculated for each post-generated mask.

B. Semi-supervised Learning Architecture

We designed a semi-supervised learning pipeline to solve the weed detection task in a 2-stage process: a self-supervised learning phase, followed by a supervised learning phase.

In self-supervised learning, the task that we use for pre-training is known as the "pretext task", such as image colorization and relative position prediction, which is used to guide the network to learn a good representation of the image. The tasks that we then use for fine tuning are known as the "downstream tasks" [9], examples of which include object detection, semantic segmentation,

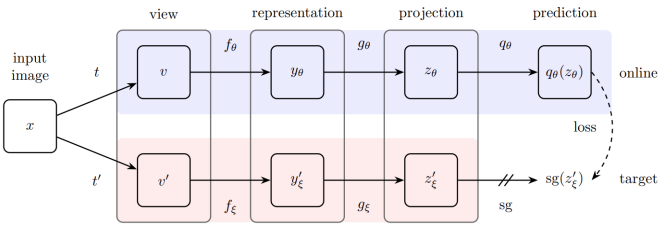


Fig. 1: BYOL's Architecture. t and t' represent different transformations applied on the same training image. BYOL minimizes a similarity loss between $q\theta(z\theta)$ and $sg(z'_\xi)$, where θ are weights updated via gradient descent, ξ are an exponential moving average of θ and sg means stop-gradient.

and image captioning. The pretext task for the self-supervised learning stage in the weed detection context is defined as having the network identify augmented views of the same plant image, as suggested by the authors. Intuitively, by performing various image augmentations on the same image, the network is taught to tolerate minor variances in data. Thus, the model becomes robust to irrelevant view changes and learns only the key features relevant to the downstream detection task.

The BYOL architecture is shown in Figure 1. Two differently transformed views v, v' of the same image are separately fed into an "online network" and a "target network", whose parameters are labeled by θ and ξ respectively. The online network is updated by gradient descent, thus "online", and the target network is updated as an exponential moving average of the online network's parameters. Each of the network encodes a view to obtain representations y_θ and y'_ξ, which were then put through a projection stage and a prediction stage. We define the similarity loss in our experiments as the mean squared error between the normalized predictions and target projections:

$$\mathcal{L}_{\theta,\xi} = ||\bar{q_\theta} - \bar{z'_\xi}||_2^2 = 2 - 2 \cdot \frac{\langle q\theta(z\theta), z'_\xi \rangle}{||q\theta(z\theta)||_2 \cdot ||z'_\xi||_2} \quad (1)$$

where $\bar{q_\theta} = q\theta(z\theta)/||q\theta(z\theta)||_2$ and $\bar{z'_\xi} = z'_\xi/||z'_\xi||_2$ are the $l2$-normalized predictions and projections. The BYOL network is trained for a number of epochs until the loss converges. At the end of training, only $f\theta$ is extracted and used as the backbone for downstream tasks, object detection and instance segmentation in our case, and everything else is discarded. The architecture for downstream task that incorporates BYOL-trained network is shown in Figure 2. This architecture is then fine-tuned in the same way as the baseline model with labeled training data.

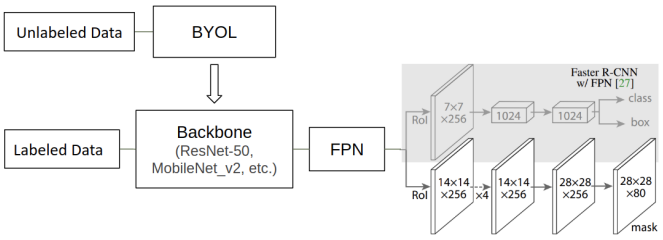


Fig. 2: Mask R-CNN detection architecture incorporating BYOL training step. The backbone trained by BYOL is transplanted into Mask R-CNN architecture to be further fine-tuned via supervised learning. The Faster R-CNN architecture is modified in the same manner.

C. Otsu-based Image Segmentation in L^*a^*b Color Space

For object detection models where only bounding box predictions are available, the post-training mask-generation process is achieved using Otsu-based thresholding [14] on images in L^*a^*b color space. Figure 3 presents the content of individual L^*a^*b color channel and provides an example of vegetation mask generated using the proposed method. The abbreviates L stands for luminances (lightness), a indicates color balance between green and magenta, and b indicates the color balance between blue and yellow. Based on the assumption that the plants usually have a significant color difference from its background, i.e. the soil, we are able to directly perform thresholding on one of the L^*a^*b channels, usually the a-channel, because plants are most-commonly seen as green.

Otsu's thresholding method involves finding a possible threshold value that separates the image into a foreground and a background so that the sum of pixel level spreads on each side is at its minimum. The spread can be measured by variances of pixel levels. Because Otsu-based thresholding works extremely robustly when generating vegetation mask, it is assumed that the mask generated from each of the predicted bounding boxes is accurate, and we therefore only focus on the performance of bounding box predictions when evaluating the Faster R-CNN model.

D. Finding Grasp Center

The ultimate the goal of the detection pipeline is to output a location on image space that serves as a grasp point for the robot arm to pull the weeds. The grasp point is selected as the barycenter of each generated weed mask, namely the geometric centroid. For non-convex shapes like vegetation masks, the centroid is not guaranteed to always lie within. However, as we are usually working with top-down views of the plants and there's underlying symmetry of plant leaves' distribution, the found grasp centers lie within the mask boundaries nearly all the time.

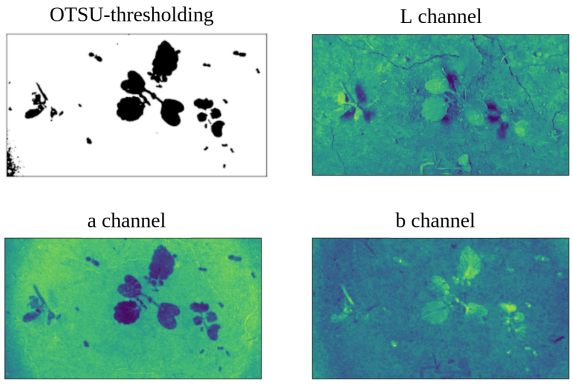


Fig. 3: Example of vegetation masks generated by Otsu-based thresholding method on a-channel of plant image in L^*a^*b color space. The a-channel is capable of discriminating the "greenness" from the background in the picture.

IV. Experiments and Results

A. Dataset

The backbone networks involved with the experiments were all pre-trained on the ImageNet dataset [27] and then fine-tuned and evaluated on the 3 smaller datasets whose detailed information is given below. The small-scaled datasets, CWFID and OPEN_DB, were used in supervised learning architectures, Mask R-CNN and Faster R-CNN. DeepWeeds is only used in the self-supervised training given its lack of box and mask annotations. How the fine-tuned models can be applied on real-world garden images captured from the robot garden is discusses in section IV-E.

1) Crop Weed Field Image Dataset (CWFID): contains 60 field images, vegetation segmentation masks, and crop/weed plant type annotations, and compromising of 162 crop plants, 332 weed plants in total.
2) The Food Crops and Weed Images Dataset (OPEN_DB): contains 1,118 raw images with around 7,853 manually-labeled bounding boxes and crop/weed plant type annotations.
3) DeepWeeds Dataset (DeepWeeds): contains 17,509 images (size 256 x 256) capturing eight different weed species with only species plant type annotations.

B. Baseline: Semantic Segmentation

The baseline semantic segmentation model, Mask R-CNN, were trained and tested on the CWFID dataset.

Method	BYOL Dataset	R-CNN Dataset	Box AP (IoU=0.75)	Box AP (IoU=0.5)	Box mAR (maxDet=100)	Mask AP (IoU=0.5)	Mask mAR (maxDet=100)
MaskRCNN	-	CWFID	0.298	0.637	0.433	0.508	0.365
FasterRCNN	-	OPEN_DB	0.498	0.704	0.554	-	-
BYOL+MaskRCNN	CWFID	CWFID	0.303	0.750	0.466	0.660	0.397
	OPEN_DB	CWFID	0.294	0.743	0.466	0.634	0.397
	DeepWeeds	CWFID	0.368	0.759	0.465	0.669	0.394
BYOL+FasterRCNN	CWFID	OPEN_DB	0.498	0.679	0.541	-	-
	OPEN_DB	OPEN_DB	0.513	0.695	0.553	-	-
	DeepWeeds	OPEN_DB	0.514	0.695	0.570	-	-

TABLE I. Object detection and segmentation results using different methods. All methods were fine-tuned on ResNet-50-FPN backbone pre-trained on ImageNet. For each case, the BYOL dataset was only used for training, and the R-CNN dataset were used for both training and testing. The CWFID data were trained with batch_size=2 and OPEN_DB data were trained with batch_size=4, considering the training speed-accuracy trade-offs.

Performance is evaluated by calculating Average Precision(AP) and Average Recall(AP) of predicted bounding boxes and masks at different Intersection over Union (IoU) thresholds. For bounding box prediction, AP at IoU=0.5, and IoU=0.75, and mean AR over a range of IoUs (IoU=0.5:0.05:0.95) are reported. For mask predictions, mean AP is calculated by averaging the pixel-coverage ratios of the predictions comparing to the ground truths (IoU=0.5:0.05:0.95), and similarly for mean AR.

We first examine the effect of varying batch size on performance of the Mask R-CNN architecture, shown in Table III. A batch size of 2 has significant better results than batch size of 16 in both AP and AR. It closely aligns with a general understanding that larger batch sizes are usually detrimental to the accuracies of deep learning models.

batchsize	Box AP (IoU=0.5)	Box mAR (maxDet=100)	Mask AP (IoU=0.5)	Mask mAR (maxDet=100)
2	0.750	0.466	0.660	0.397
4	0.586	0.420	0.498	0.356
8	0.586	0.414	0.482	0.321
16	0.584	0.397	0.478	0.314

TABLE II. MaskRCNN performance against batch sizes on CWFID dataset using ResNet-50-FPN backbone. There are in total 60 images in the dataset. Mean AR are calculated by averaging AR at IoU=0.5:0.05:0.95 where maximum detection over which it is calculated is 100 (maxDet=100).

A light-weighted backbone, MobileNet_v3 [28] is also experimented. Its small size and efficient resource-constraint implementation enables it to process on a mobile phone CPU, which may allow it to run on-board from the robot. The performance of the model is significantly worse than the ResNet-50 backbone, but the results are still reported here for reference.

C. Baseline: Object Detection

The baseline object detection task is trained and tested with the OPEN_DB dataset using a Faster R-CNN architecture. Since the mask-generation for individual plant is done with accurate Otsu-based thresholding, only bounding box prediction results were evaluated and presented in Table I. The bounding box predictions of the Faster R-CNN are better than Mask R-CNN despite being trained with a larger batch size: 4 for Faster R-CNN versus 2 for Mask R-CNN. This can be attributed to the merit that it is trained with a larger dataset, thus having a chance to learn a better representation of plant features. This also suggests that data deficiency is indeed a challenge to deep learning for accurate weed detection tasks.

Backbone	#param	FLOPs	Box AP (IoU=0.5)	Box mAR (maxDet=100)	Mask AP (IoU=0.5)	Mask mAR (maxDet=100)
ResNet-50	25.6M	3800M	0.750	0.466	0.660	0.397
MobileNet	6.9M	300M	0.654	0.331	0.161	0.073

TABLE III. Mask R-CNN results comparison using MobileNet_v3 and ResNet-50 Backbones. Both models were trained with batch_size=2.

D. Semi-Supervised Learning

The BYOL architecture was trained with large batch sizes of 96 to achieve a good performance as the paper suggested. We used a BYOL backbone pre-trained on ImageNet in our experiments, which is then updated in our self-supervised training stage. The extracted backbone network is finally fine-tuned by performing supervised learning. The BYOL+R-CNN architecture was tested with different dataset combinations for training and testing, and the results are shown in Table I.

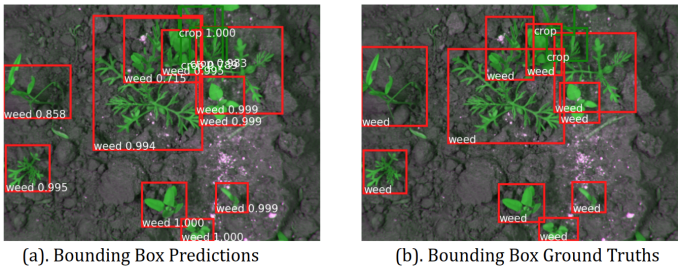
By performing both self-supervised learning and supervised learning using the same dataset, we observe an overall significant improvement of results on CWFID dataset but no significant change on performance of OPEN_DB. This suggests that the addition of self-supervised learning architecture is capable of directing the backbone network to capture extra information by comparing projected features output from the 2 networks

of BYOL directly, unlike in the supervised learning process where the network predicts and regress on the label data such as bounding boxes and masks. Expressly, it suggests that the learned information through self-supervised training and supervised training on the same dataset was not identical. This advantage capacitates the network to generate better feature representations for recognizing objects in a scene.

The results also suggest that performing the 2 stages of trainings on different datasets, i.e. train BYOL with CWFID and R-CNN with OPEN_DB or vice versa, does not necessarily improve the performance and might bias the network weights towards a wrong direction. This may be explained considering the sizes of CWFID and OPEN_DB were relatively small, thus the data might belong to two relatively further distributions in the latent space (i.e. encoded feature space).

Training the backbone network with BYOL on a large-scale dataset like DeepWeeds produces the best performance for both R-CNN models. This matches with the expectations because the network learned additional information during the self-supervised learning stage and is able to retain that knowledge in the future training stages. This inherits the reasoning of transfer learning where a model trained with one task, i.e. the pretext task of BYOL, is exploited to improve generalization in another setting, i.e. a better feature representation for recognizing and distinguishing plants.

An example of bounding box (Figure 4), mask prediction (Figure 5) and grasp point calculation results of a test image in CWFID dataset is provided. The best-performing network, i.e. Mask R-CNN model with pre-trained backbones on DeepWeeds, is shown in Figure 4. The model produce very accurate bounding boxes of correct categories. In addition, the predicted masks of each individual plants have very clear and distinct boundaries, indicating the model successfully learned to distinguish plant objects from soil background.



(a). Bounding Box Predictions (b). Bounding Box Ground Truths

Fig. 4: Example of bounding box predictions of Mask R-CNN (left) and the ground truth (right). The predictions are relatively accurate and have high beliefs for each prediction.

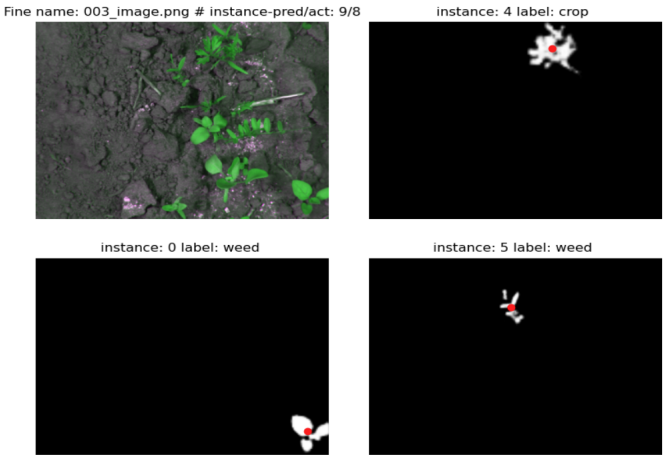


Fig. 5: Example of generated masks and calculated grasp points for 3 of the detected objects. The grasp-points are represented by the red dots plotted on the white vegetation mask. The top right mask has a blurry boundaries because the corresponding plant's leaves are thin and interleaving.

E. Adapting to Real Robot Garden Images

Because large amount of annotations for plants are often not available in real-world scenarios, the model should able to adapt to the new environments with only a few trials of teach-and-learn iterations. The validity of adapting the model to real-world scenarios can be mimicked by feeding a trained model with only a few labeled data points from an unseen dataset for training, and evaluate its performance on the test set of the unseen dataset. We refer it as the "model adaptation" to the new environment.

An example setup that we use for testing the adaptation ability of the model is to have a trained model, e.g. DeepWeeds for BYOL training, and OPEN_DB for R-CNN training, to learn 10 image-label pairs from the CWFID dataset, and then being tested on the test set of the CWFID dataset. Because OOPEN_DN dataset does not provide mask labels, we trained a Faster R-CNN model in this case. The performance the adapted model is shown in Table IV. The model have an acceptable performance in box AP. The masks and grasp-points are generated using the Otsu-based thresholding method (an example is shown in Figure 6), whose performance relies on the quality of bounding box detection.

BYOL Dataset	R-CNN Dataset	Adaptation Dataset	Box AP (IoU=0.75)	Box AP (IoU=0.5)	Box mAR (maxDet=100)
DeepWeeds	OPEN_DB	CWFID	0.230	0.527	0.384

TABLE IV. Performance of model adaptation to unseen dataset. Trained with batch_size=1.

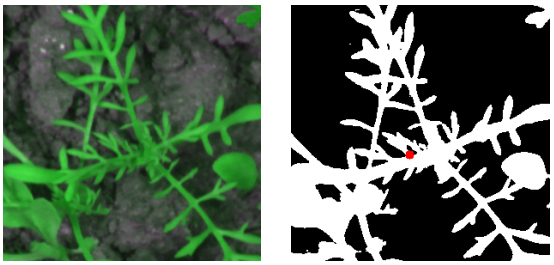


Fig. 6: Example of a bounding box prediction (left) cropped from the original test image and the mask generated via Otsu-based thresholding (right). The grasp-center is plotted on the vegetation mask in a red point. Notice that the mask is not capable of distinguishing different plant individuals.

V. Conclusion

The proposed semi-supervised detection architecture serves as an effective solution to the data deficiency problem currently existing in accurate weed detection tasks. The addition of self-supervised training process enables the network backbone to capture additional information especially when it is trained with a large-scale unseen data. In general, the network is able to produce a better representation for down-stream tasks like object detection and segmentation.

Otsu-based thresholding method is capable of accurately extracting plants from background however it is incapable of distinguishing different object instances, leading to inaccurate grasp-center results. This can be used as a naive solution when no mask annotation labels are available. Otherwise, Mask R-CNN model is always preferred. Since the thresholding method produces a very sharp and clear vegetation boundaries, it can alternatively be used to refine the mask-outputs from the Mask R-CNN network, which sometimes can be blurry.

VI. Future Work

One potential aspect of future work resides in the training of the self-supervised model. As contrastive learning generally produces excellent performance, it is worth exploiting it to help a network distinguish different categories (e.g. weeds/crops or different plant species). A large-scaled dataset such as DeepWeeds can be used to train such models, with the negative pairs being plants of different categories according to the labeled ground truth.

In this project, we opt for simplicity and adopt the commonly used ResNet backbone for our experiments. To aim for better accuracy, there are several cutting-edge backbone structures to be explored. Two promising better-performing backbones include ResNeSt [29], which is a derivative structure from ResNet, and Swin Transformer [30], a network structure that was previously used to capture time-context. Swin Transformer computes efficient local self-attention and is capable of dealing with large variations in the scale of visual entities. The transformer network can be used as a drop-in replacement of the current ResNet-50-FPN backbone in the detection pipeline. Using the pre-trained Swin transformer backbone, the model can be fine-tuned on small datasets such as CWFID or OPEN_DB to perform down-stream tasks such as object detection and segmentation.

REFERENCES

[1] A. dos Santos Ferreira, D. Matte Freitas, G. Gonçalves da Silva, H. Pistori, and M. Theophilo Folhes, "Weed detection in soybean crops using convnets," *Computers and Electronics in Agriculture*, vol. 143, pp. 314 – 324, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168169917301977

[2] A. Wendel and J. Underwood, "Self-supervised weed detection in vegetable crops using ground based hyperspectral imaging," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 5128–5135.

[3] D. C. Slaughter, D. K. Giles, and D. Downey, "Autonomous robotic weed control systems: A review," *Computers and Electronics in Agriculture*, vol. 61, no. 1, pp. 63 – 78, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168169907001688

[4] A. Wang, W. Zhang, and X. Wei, "A review on weed detection using ground-based machine vision and image processing techniques," *Computers and Electronics in Agriculture*, vol. 158, pp. 226 – 240, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168169918317150

[5] A. Olsen, D. A. Konovalov, B. Philippa, P. Ridd, J. C. Wood, J. Johns, W. Banks, B. Girgenti, O. Kenny, J. Whinney, B. Calvert, M. Rahimi Azghadi, and R. D. White, "DeepWeeds: A Multiclass Weed Species Image Dataset for Deep Learning," *Scientific Reports*, vol. 9, no. 2058, 2 2019. [Online]. Available: https://doi.org/10.1038/s41598-018-38343-3

[6] Z.-Q. Zhao, P. Zheng, S. tao Xu, and X. Wu, "Object detection with deep learning: A review," 2019.

[7] S. H. Lee, C. S. Chan, P. Wilkin, and P. Remagnino, "Deep-plant: Plant identification with convolutional neural networks," in *2015 IEEE International Conference on Image Processing (ICIP)*, 2015, pp. 452–456.

[8] S. Haug and J. Ostermann, "A crop/weed field image dataset for the evaluation of computer vision based precision agriculture tasks," in *Computer Vision - ECCV 2014 Workshops*, 2015, pp. 105–116. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-16220-1_8

[9] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," 2019.

[10] Y. Zhu and D. Berenson, "Plant Detection for Gardening Robot," *Independent Study Report, University of Michigan*, 12 2020.

[11] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko, "Bootstrap your own latent: A new approach to self-supervised learning," 2020.

[12] A. E. P. Lara, C. Pedraza, and D. A. Jamaica-Tenjo, "Weed estimation on lettuce crops using histograms of oriented gradients and multispectral images," 2020.

[13] G. M. Gandhi, S. Parthiban, N. Thummalu, and A. Christy, "Ndvi: Vegetation change detection using remote sensing and gis – a case study of vellore district," *Procedia Computer Science*, vol. 57, pp. 1199–1210, 2015, 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050915019444

[14] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[15] J. Champ, A. Mora-Fallas, H. Goëau, E. Mata-Montero, P. Bonnet, and A. Joly, "Instance segmentation for the fine detection of crop and weed plants by precision agricultural robots," *Applications in Plant Sciences*, vol. 8, no. 7, 2020. [Online]. Available: https://hal.inrae.fr/hal-02910844

[16] K. Osorio, A. Puerto, C. Pedraza, D. Jamaica, and L. Rodríguez, "A deep learning approach for weed detection in lettuce crops using multispectral images," *AgriEngineering*, vol. 2, no. 3, pp. 471–488, 2020. [Online]. Available: https://www.mdpi.com/2624-7402/2/3/32

[17] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," 2014.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[19] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2016.

[20] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," 2018.

[21] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2019.

[22] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," 2020.

[23] P. Bachman, R. D. Hjelm, and W. Buchwalter, "Learning representations by maximizing mutual information across views," 2019.

[24] O. J. Hénaff, A. Srinivas, J. D. Fauw, A. Razavi, C. Doersch, S. M. A. Eslami, and A. van den Oord, "Data-efficient image recognition with contrastive predictive coding," 2020.

[25] Z. Wu, Y. Xiong, S. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance-level discrimination," 2018.

[26] Y. Tian, X. Chen, and S. Ganguli, "Understanding self-supervised learning dynamics without contrastive pairs," 2021.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[28] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam, "Searching for mobilenetv3," 2019.

[29] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, H. Lin, Z. Zhang, Y. Sun, T. He, J. Mueller, R. Manmatha, M. Li, and A. Smola, "Resnest: Split-attention networks," 2020.

[30] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," 2021.