Semantic-aware plant traversability estimation in plant-rich environments for agricultural mobile robots

Shigemichi Matsuzaki, Hiroaki Masuzawa and Jun Miura
Toyohashi University of Technology

Abstract—This paper describes a method of estimating the traversability of plant parts covering a path and navigating through them in greenhouses for agricultural mobile robots. Conventional mobile robots rely on scene recognition methods that consider only the presence of objects. Those methods, therefore, cannot recognize paths covered by flexible plants as traversable. In this paper, we present a novel framework of the scene recognition based on image-based semantic segmentation for robot navigation that takes into account the traversable plants covering the paths. In addition, for easily creating training data of the traversability estimation model, we propose a method of generating labels of traversable regions in the images, which we call traversability masks, based on the robot’s traversal experience during the data acquisition phase. It is often difficult for humans to distinguish the traversable plant parts on the images. Our method enables consistent and automatic labeling of those image regions based on the fact of the traversals. We conducted a real-world experiment and confirmed that the robot with the proposed recognition method successfully navigated in plant-rich environments.

I. INTRODUCTION

Robot technologies are attracting attention as a means of improving the product efficiency and reducing the labor burden in agricultural industry, where the aging population of the labors and harsh working environments are severe problems. The mobile robot technologies in general have come nearly to the level of practical realization and commercialization in some situations. For example, self-driving cars are already being tested in public areas [1] [2]. Another example is the service robots operating in public places such as airports and stores [3] [4]. Those applications mainly target well-structured environments such as urban areas and the inside of buildings. In plant-rich environments such as agricultural fields, however, it is more difficult for robots to autonomously navigate themselves than in structured environments due to the uncertainty that stems from the presence of plants and the difference of the individuals of the plants etc. Furthermore, the paths are possibly covered by plant parts such as branches and leaves, which, though, can be driven through by the robots. Most of conventional mobile robots only consider the presence of objects and do not consider such traversable objects.

In this paper, we present a novel framework of the scene recognition and robot navigation that takes into account the traversable plants covering the paths. We introduce a deep neural network (DNN) architecture consisting of Semantic Segmentation Module (SSM) for pixel-wise object label assignment and Traversability Estimation Module (TEM) for estimating how likely that each pixel can be traversed by robots. SSM is trained by an unsupervised domain adaptation method proposed by the authors [5], without a manually labeled greenhouse dataset. TEM is then trained using the features from SSM as inputs and binary labels that indicates the regions traversed by the robot, coined traversability masks. The traversability masks are automatically generated based on the traversal experience of the robot during the data acquisition. The overall network is therefore trained without manual annotation of the training images. Since the traversability masks are generated based on actual traverses of the robot, there may exist unlabeled traversable regions in the images. Directly treating the labeled part as positive and the rest as negative is inappropriate. Therefore, along with the labeling method, we introduce a positive-unlabeled (PU) learning-based model in order to take full advantage of the information given by the traversability masks.

The contributions of this paper are as follows:

1) A method of automatically labeling traversable regions in the images based on robots’ experience of traversals.
2) A method of learning features of traversable plants by
fully exploiting the sparse labels of traversable regions in the images based on PU learning.

3) Real-world experiment of robot operation with the proposed recognition method.

II. RELATED WORK

A. Traversability estimation

Traversability estimation is a crucial task in robot navigation. Traditional methods solely rely on geometric structure of the environments such as presence of rigid obstacles and slopes. In unstructured outdoor scenes, many studies have focused on estimating the traversability of unpaved terrains by analyzing the structure of the environments. A major approach is to train the traversability within the measurement range of stereo cameras using geometric and image features, and propagate the learned traversability to farther areas based on only image features [6, 7, 8].

Using information of robot’s traversal is a preferred approach for identifying traversable regions in recent studies. Barnes et al. [9] used trajectories of an autonomous car for automatically labeling images for semantic segmentation. Similarly, Wellhausen et al. [10] proposed to use footprints of a legged robot to label images and showed that sufficient performance can be achieved even with such sparse labels. Some studies consider the traversability of objects as an affordance depending on both the environment and the robots. Kim et al. [11] trained a classifier with simple hand-designed features by exploiting a robot’s experience of successful and unsuccessful traversals acquired by its interaction with objects. Ugur et al. [12, 13] sought to make a robot learn objects’ affordance of traversability depending on their shapes and the robot’s experience. Our method shares with those methods a similar idea to use a robot’s experience of traversals. Unlike those methods, our method targets plants hanging over and does not limit the object shapes to primitive ones such as spheres, rectangles, and ground surfaces. In addition, by exploiting deep features and a convolutional neural network, our method provides better generalization than those based on simpler classifiers. Specifically, we exploit the image features trained on a semantic segmentation task, which are expected to be informative for traversability estimation as the object class is a part of necessary information of objects’ traversability.

B. Agricultural mobile robots

Many of currently used agricultural robots move on rails or heating pipes equipped on the paths [14, 15]. Robotic systems with such locomotion are able to move stably, but limits the area of operation. More flexible robot operation is beneficial as it can broaden the field of tasks that the robots can automate and thus reduce workers’ burden.

In outdoor, precise localization by GNSS is adopted in various works [16, 17]. In indoor fields such as greenhouses, however, such technology is not available due to noise imposed by surrounding obstacles. Therefore, a popular approach is to localize itself and recognizing a traversable path using sensors equipped on the robot. The sensors used in path recognition include ultrasonic sensors [18, 19], LIDARs [20, 21] and monocular / stereo cameras [22, 23].

In one of pioneering work by Mandow et al. [18], it has been reported that while they succeeded to autonomously operate a mobile robot using an ultrasonic sensor, some failure cases observed where the robot recognized plant parts overhanging the path as obstacles and stopped the navigation. To the best of our knowledge, the problem of recognizing traversable plant parts covering the paths has not been explicitly addressed.

C. Semantic segmentation

Semantic segmentation is a task to assign an object class to each pixel of an image. Long et al. [24] first proposed to convert classification networks into fully convolutional network to produce a probability map for each pixel. Based on ResNet proposed by He et al. [25] as a backbone network, very deep networks have been shown to achieve high accuracy [26, 27]. While it is proven to be effective to stack layers in depth to get a high accuracy, those models impose high computational cost. For real-time visual tasks, computationally efficient architectures have also been studied, targeting the applications such as robotics and autonomous driving systems [28, 29, 30]. Recent approaches further improved the accuracy while maintaining the efficiency by introducing more efficient convolution approaches such as factorized convolution [31], depth-wise separable convolution [32], efficient spatial pyramid [33, 34] etc.

In our previous work, we have shown that a semantic segmentation network for greenhouses can be trained without requiring hand-labeled images, by utilizing multiple publicly available datasets of urban scenes and employing an unsupervised domain adaptation (UDA) method [5].

D. PU learning

PU (positive and unlabeled) learning is a problem where part of the data is labeled as positive, and the rest is unlabeled and could be either positive or negative. Unlike the setting where positive and negative data are assumed, PU setting cannot be solved by just applying a simple classification of positive and unlabeled data since the unlabeled data include positive ones. Elkan and Noto [35] showed that it is possible to estimate the label probabilities by modeling the probability that given samples are labeled. As a practical application, Yang et al. [36] applied PU learning framework in object detection task where some of objects are not labeled.

III. PROPOSED METHOD

Fig. 2 shows the overview of the proposed framework. We employ two network modules for the scene recognition: Semantic Segmentation Module (SSM) for semantic segmentation, and Traversability Estimation Module (TEM) for traversability estimation. SSM is trained by an unsupervised domain adaptation method. For the training of TEM, we generate traversability masks based on the robot’s traversal experience. The network is then trained with the generated traversability masks (Fig. 2(a)). Utilizing the trained network,
we build semantic local 3D map that indicates the traversability of the regions around the robot and the navigation is operated based on the map (Fig. 2(b)).

A. Traversability mask

Traversability masks are label data that indicate the image regions traversed by the robot during the data acquisition phase. Using the traversability masks, we automatically label the regions of images dominated by traversable plant parts to train a network so that it can distinguish traversable and non-traversable plants based on the robot’s experience.

We first acquire RGB-D data by manually operating a robot inside the target greenhouse moving through the plants. We then build a 3D voxel map using RTAB-Map [37] from the RGB-D data. After that, by utilizing the robot’s path estimated in the mapping process, the voxels traversed by the robot are labeled. Finally, traversability masks are generated by projecting the traversability labels of the voxels on an image plane from each sensor pose on the map.

Examples of the traversability masks are shown in Fig. 3. One may notice that while much of traversable plant parts such as leaves and branches are labeled, there is also quite a lot of unlabeled regions of traversable plants. This is because the traversability masks are based on the limited number of traverses during the data acquisition phase. This nature of the traversability masks makes it inappropriate to simply treat them as positive and negative examples. Therefore we introduce a PU learning-based training described in III-B.2.

B. Network architecture

In this section, we describe a novel network architecture for pixel-wise traversability estimation and semantic segmentation. The network architecture is shown in Fig. 4. The network consists of Semantic Segmentation Module (SSM) for pixel-wise object label assignment and Traversability Estimation Module (TEM) for estimating how likely that each pixel can be traversed by robots. SSM is first trained individually. TEM is then trained with intermediate features of SSM as inputs and traversability masks as training labels. The reason for not training them simultaneously is to avoid overfitting of the entire network to the traversability masks.

1) Semantic Segmentation Module: Semantic Segmentation Module (SSM) is responsible for pixel-wise general object classification. Here we use 3 general object classes: Plants, artificial objects, and the ground. For training SSM, we introduce a training method in our previous work [5]. In
the method, we employ multiple publicly available datasets with pixel-wise labels (source datasets) and an unlabeled greenhouse dataset (target dataset). First, segmentation models are trained with each source dataset. Images from the target dataset are then fed into the models, and corresponding outputs are yielded. If a pixel is assigned the same object label from all the source models, the label is treated as a pseudo-label of the pixel. For training a model with the pseudo-labels, we adopt a method of adaptively weighting a loss value on each pixel based on the uncertainty of estimation on each pixel and weight the loss values based on the uncertainty.

2) Traversability Estimation Module: Traversability Estimation Module (TEM) estimates the probabilities that the pixels are traversable. In the training of TEM with the traversability masks, we introduce the PU (positive and unlabeled) learning framework. Elkan and Noto [35] state that under the “selected completely at random” assumption meaning that the labels are given completely at random, PU learning can be solved by modeling a classifier \( g(x) \) such that \( g(x) = p(s = 1|x) \), where \( x \) denotes an input data, \( s \) denotes a random variable indicating that data \( x \) is labeled if \( s = 1 \). The equation can then be transformed as follows, given the “selected completely at random” assumption, i.e.,

\[
g(x) = p(s = 1|x) = p(y = 1 \land s = 1|x) = p(y = 1|x)p(s = 1|y = 1,x) = p(y = 1|x)p(s = 1|y = 1). \tag{1}
\]

Therefore, the traversable probability of a pixel \( p(y = 1|x) \) can be calculated as follows:

\[
p(y = 1|x) = \frac{g(x)}{c}, \tag{2}
\]

where \( c = p(s = 1|y = 1) \) which denotes the conditional probability that a positive data is labeled. \( c \) is approximated by feeding the training data to the estimated \( g(x) \) and average the probabilities of the labeled features, i.e.,

\[
c = p(s = 1|y = 1) \approx \frac{1}{n} \sum_{x \in P} g(x), \tag{3}
\]

where \( P \) denotes a set of all labeled pixels in the training dataset.

In TEM, we first model the label probability \( g(x) = p(s = 1|x) \) by a convolution and a sigmoid activation. The output is then divided by \( c \) calculated by eq. (3) to produce \( p(y = 1|x) \). The feature map from the SSM is fed in TEM. The size of the feature map is \( (B,C,H,W) \) where \( B \) denotes the batch size, \( C \) denotes the channel size, \( H \) and \( C \) denote the height and the width, respectively. Here we consider a vector of size \( C \) as an input \( x \) for the corresponding pixel. 3 \( \times \) 3 convolution is applied to the features, followed by a sigmoid function to scale the output in the range of \([0, 1]\). We use 3 \( \times \) 3 convolution instead of 1 \( \times \) 1 convolution to take information of adjacent pixels into consideration in the estimation.

Note that the “selected completely at random” assumption does not strictly hold in our task. In practice, however, we confirm that this formulation is effective to estimate pixel-wise traversability.

C. 3D semantic voxel map

Using the results of the recognition explained in section II-B we build 3D semantic voxel map around the robot. At first, an RGB image acquired from the RGB-D sensor is passed to the segmentation network. The outputs from SSM and TEM are then fused with the depth image and mapped into the 3D space. To build a semantic voxel map, the 3D space around the robot is separated into voxels with a side of 0.1 [m], and each object point and probability point are assigned to a corresponding voxel. For the object class information, a histogram of the object classes is constructed in each voxel. For the traversability information, the traversability values of the points within a voxel are averaged and treated as an observation of the voxel.

Those observations are fused by Bayesian update. The observation of the object class is fused by the following equation:

\[
P(l|z_o) = \eta P(z_o|l)P(l), \tag{4}
\]

where \( l \) denotes an object label, \( z_o \) denotes an observation, which is an object class with highest frequency within the voxel. \( \eta \) is a normalization term and \( P(l) \) is the prior of label \( l \). \( P(z_o|l) \) is the likelihood of label \( l \) with observation \( z_o \), and defined as follows:

\[
P(z_o|l) = \begin{cases} L_{correct}, & \text{if } z_o = l \\ (1 - L_{correct})/N_l, & \text{otherwise} \end{cases} \tag{5}
\]

\( L_{correct} \) indicates how likely that \( l \), which is the label currently assigned to the voxel, matches with the new observation \( z_o \). In our experiment, we fix \( L_{correct} \) to 0.7. \( N_l \) denotes the number of object classes. The object class with the highest probability is assigned to the voxel as its label.

Similarly, the observation of traversability is fused by the following equation:

\[
P(\tau|z) = \eta P(z|\tau)P(\tau), \tag{6}
\]
TABLE I: Greenhouse datasets used in the training. The type of each set is shown in the brackets.

| Train  | Test     | Date              |
|--------|----------|-------------------|
| A      | 6684 (unlabeled) | May 25, 2018     |
| B      | 899 (w/trav. masks) | July 12, 2019   |

where \( \tau \in \{0, 1\} \) denotes an event that the voxel is traversable (1) or not traversable (0). The likelihood \( P(z_t|\tau) \) is calculated from the estimation of TEM and the traversability masks.

After the integration of observed points to the semantic voxel map, obstacle point cloud is generated by treating the voxels with “plant” class and the traversability higher than the threshold as free spaces and the others as obstacles. By feeding the obstacle point cloud to a conventional navigation module, the navigation in plant-rich environments is realized. For mitigating the degradation of the speed as the number of voxels increases, the voxels where no point has been observed for a certain number of consecutive frames are removed from the map. In the experiment below, the number of frames is set to 10.

IV. EXPERIMENTS

A. Evaluation of TEM

1) Used datasets: For the training of SSM by the UDA method, we used three source datasets: CamVid [39], Cityscapes [40], and Freiburg Forest [41], for generating pseudo-labels. As target data, we used 6684 unlabeled greenhouse images from Greenhouse A dataset, taken in a greenhouse growing tomatoes.

For the training of TEM, we used 899 pairs of an RGB image and the corresponding traversability mask from Greenhouse B dataset, taken in the same greenhouse as Greenhouse A on a different date. Greenhouse A and B have a different appearance due to a different level of growth of the plants. For the evaluation, we manually gave true labels of traversable regions on 33 images from Greenhouse A. Overview of the greenhouse datasets is shown in Table I.

2) Training conditions and hyperparameters: The training epoch is set to 200 with a fixed learning rate of \( 5 \times 10^{-5} \), and the batch size of 64. TEM is then trained with weights of SSM fixed. The number of epoch is set to 200 and the batch size of 64. We use cyclical learning rate scheduling [42]. The initial learning rate is \( 5 \times 10^{-5} \), and linearly increases by factor of 10 in 10 epochs and then decreases to the original value in 20 epochs.

3) Results: We evaluate the binary traversability images generated from the predicted traversability with different thresholds. Here we compare two types of binary images: raw and refined. The raw binary images are generated by just binarizing the predicted traversability with a threshold. The refined binary images are generated by additionally setting all the pixels predicted as an artificial object or the ground to non-traversable. This refinement provides the filtering effect of false positive traversability predictions even in the case of high predicted traversability, which is the same as the calculation in the voxel described in III-C where the final decision of traversability is given considering both the object class and the predicted traversability. Fig. 5 shows a Precision-Recall curve as well as an IoU-Recall curve to analyze the performance of TEM. We found that 0.75 is the best threshold for pixel-wise binary traversability in terms of IoU. Table II shows the accuracy, precision, and recall of the model on the test set with the threshold of 0.75. From Fig. 5 and Table II we could see high recall and relatively low precision. The refined predictions resulted in better precision and IoU. This result indicates that fusing the predictions of object classes and traversability decreased the false positive rate and thus is suitable for safe robot operation compared to only predicting the traversability.

Fig. 6 is the visualization of the prediction by TEM. Much of the traversable regions are classified as positive. Besides, some non-traversable plants are predicted correctly. This result shows that our model can learn the differences between traversable and non-traversable plants. In terms of the refined results, we cannot qualitatively see obvious improvements, but we confirmed that small prediction noise was suppressed. Note that these segmentation results are yielded from automatically generated traversability masks, which are sparse and noisy. Even from such data, our PU learning-based training framework effectively learns the features of traversable objects, and provides reasonable estimation.

B. Navigation

Finally, we conducted a real-world navigation experiment in a greenhouse. All the software is processed on a laptop with NVIDIA GeForce 960M and 32 GB RAM. Revast Mercury is used as a robot platform. The robot used in the experiment is shown in Fig. 7(a).

Image of the greenhouse is shown in Fig. 7(b). We assumed that the path is straight and constant control signals.
Fig. 6: Prediction results of TEM. From the top: Camera images, Ground truth, predicted binary images with threshold 0.75, and refined binary images. Note that the traversable regions are not identical to the plant regions, but plant parts such as stems are not traversable. TEM is capable of distinguishing the traversable plants from the non-traversable ones (highlighted in yellow ellipses). The refinement by predicted object classes mainly worked as noise suppression (highlighted in red rectangles).

Fig. 7: Experiment environment

(a) Robot configuration  
(b) Inside the greenhouse

Fig. 7: Experiment environment

of linear velocity is given to the robot while the robot is recognizing the traversability of objects in front. The linear velocity was set to 0.1 [m/s] When non-traversable object is detected, the control signal is stopped. As a baseline, we also tested a system that considers all the voxels as obstacles.

We carried out 5 trials on the same path and confirmed that our system was able to recognize the traversable plants and navigate through the path between the plant rows in all the trials, while a baseline method failed the task. However, there were some cases where the robot stops for a while and then restarts even though there was no obstacle. This was due to misclassification of voxels in front of the robots as obstacles. The misclassification of voxels stems either from the error in the image-based semantic segmentation or from the registration error between the predicted segmentation images and the corresponding depth images, which RGB-D sensors inevitably have.

Fig. 8 is the visualization of the semantic 3D maps generated during the experiment at different times in chronological order. There was a lot of plant parts partially covering the path. Our proposed method was able to recognize the traversable plants and to traverse them. Fig. 8(b) shows the case where the system wrongly recognized the regions in a far front of the robot as obstacles shown in the red voxels. When the robot stopped moving in response to this situation, the voxels were already out of the sensor’s vertical field of view. The voxels were then removed from the map after the pre-defined number of frames and the robot restarted. For more reliable navigation, the accuracy of the recognition model needs to be improved. In addition, to deal with the prediction noise in DNN models, a better filtering method should be implemented.

V. CONCLUSIONS

We proposed a method of estimating the traversability of plants covering a path and navigating through them in greenhouses for agricultural mobile robots. For image-based traversability estimation as well as semantic segmentation, we developed a novel network architecture and its training method. We proposed a method to automatically generate pixel-wise labels of traversable regions in the images, called Traversability mask, based on the robot’s experience of traversing the regions during the data acquisition phase. For effectively train the model with the sparse traversability masks including unlabeled traversable regions, we introduced a PU learning in the network training. We conducted navigation experiment in a greenhouse and confirmed that our method was capable of distinguishing traversable plants.
from non-traversable ones and thus able to realize navigation through paths covered by plants.

As future work, we are going to work on the improvement of the navigation system. In particular, the accuracy of the scene recognition and the filtering of misclassification should be improved. In addition, we are also going to implement a path planning considering the traversable plants as well as normal obstacles and free spaces in order to realize high-level navigation with arbitrary paths inside greenhouses.

ACKNOWLEDGMENT

This work is supported by Knowledge Hub Aichi Priority Research Project (3rd term). The first author is also supported in part by the Leading Graduate School Program, “Innovative program for training brain-science-information-architects by analysis of massive quantities of highly technical information about the brain,” by the Ministry of Education, Culture, Sports, Science and Technology, Japan.

REFERENCES

[1] Waymo, “Waymo Safety Report,” Tech. Rep., 2020. [Online]. Available: https://storage.googleapis.com/sdc-prod/v1/safety-report/2020-09-waymo-safety-report.pdf.

[2] Tier IV, “Tier IV Safety Report,” Tech. Rep., 2020. [Online]. Available: https://assets.ctfassets.net/rfcf1c9f6x4w1/3ywp7p9xHoVM7dje77hxqX/f9468a13518bee9876ed7270c8ba76de/Safety_Report_Eng_9_29_master_compressed.pdf.

[3] A. K. Pandey and R. Gelin, “A Mass-Produced Sociable Humanoid Robot: Pepper: the First Machine of Its Kind,” IEEE Robotics and Automation Magazine, vol. 25, no. 3, pp. 40–48, 2018.

[4] R. Triebel et al., SPENCER: A socially aware service robot for passenger guidance and help in busy airports. 2016, pp. 607–622.

[5] S. Matsuzaki et al., “Multi-source Pseudo-label Learning of Semantic Segmentation for the Scene Recognition of Agricultural Mobile Robots,” 2021, pp. 1–10. arXiv: 2102.06386 [Online]. Available: https://arxiv.org/pdf/2102.06386.pdf.

[6] M. Happold et al., “Enhancing Supervised Terrain Classification with Predictive Unsupervised Learning.” In Proc. of the Robotics: Science and Systems, 2006.

[7] R. Hadsell et al., “Deep Belief Net Learning in a Long-Range Vision System for Autonomous Off-Road Driving.” In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008, pp. 628–633.

[8] M. Wang et al., “Learning Long-range Terrain Perception for Autonomous Mobile Robots,” International Journal of Advanced Robotic Systems, vol. 7, no. 1, pp. 55–66, 2010.

[9] D. Barnes et al., “Find your own way: Weakly-supervised segmentation of path proposals for urban autonomy.” In Proc. of the IEEE International Conference on Robotics and Automation, 2017, pp. 203–210.

[10] L. Wellhausen et al., “Where should i walk(Predicting terrain properties from images via self-supervised learning,” IEEE Robotics and Automation Letters, vol. 4, no. 2, pp. 1509–1516, 2019.

[11] D. Kim et al., “Traversability Classification using Unsupervised On-line Visual Learning for Outdoor Robot Navigation.” In Proc. of the IEEE International Conference on Robotics and Automation, 2006, pp. 518–525.

[12] E. Uğur et al., “The learning and use of traversability affordance using range images on a mobile robot.” In Proc. of the IEEE International Conference on Robotics and Automation, 2007, pp. 1721–1726.
[13] E. Uğur et al., “Curiosity-driven learning of traversability affordance on a mobile robot.” In Proc. of the IEEE International Conference on Development and Learning, 2007, pp. 13–18.

[14] L. Grimstad et al., A Novel Autonomous Robot for Greenhouse Applications, 2018.

[15] The SweeperProject, Homepage Sweeper, 2018. [Online]. Available: [http://www.sweeper-robot.eu/](http://www.sweeper-robot.eu/) (visited on 01/12/2020).

[16] Y. Nagasaka et al., “Autonomous guidance for rice transplanting using global positioning and gyroscopes,” Computers and Electronics in Agriculture, vol. 43, no. 3, pp. 223–234, 2004.

[17] L. Cantelli et al., “A Small Versatile Electrical Robot for Autonomous Spraying in Agriculture,” AgriEngineering, vol. 1, no. 3, pp. 391–402, 2019.

[18] A. Mandow et al., “The Autonomous Mobile Robot Aurora for Greenhouse Operation,” IEEE Robotics and Automation Magazine, vol. 3, no. 4, pp. 18–28, 1996.

[19] F. Dong et al., “Development of a row guidance system for an autonomous robot for white asparagus harvesting,” Computers and Electronics in Agriculture, vol. 79, no. 2, pp. 216–225, 2011.

[20] F. B. Malavazi et al., “LiDAR-only based navigation algorithm for an autonomous agricultural robot,” Computers and Electronics in Agriculture, vol. 154, pp. 71–79, 2018.

[21] E. H. Harik et al., “Combining Hector SLAM and Artificial Potential Field for Autonomous Navigation Inside a Greenhouse,” Robotics, vol. 7, no. 2, p. 22, 2018.

[22] P. J. Younse and T. F. Burks, “Greenhouse Robot Navigation Using KLT Feature Tracking for Visual Odometry,” Agricultural Engineering International: CIGR Journal, vol. 9, 2007.

[23] J. Xue et al., “Variable field-of-view machine vision based row guidance of an agricultural robot,” Computers and Electronics in Agriculture, vol. 84, pp. 85–91, 2012.

[24] J. Long et al., “Fully Convolutional Networks for Semantic Segmentation.” In Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.

[25] K. He et al., “Deep Residual Learning for Image Recognition.” In Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[26] H. Zhao et al., “Pyramid scene parsing network.” In Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2017, pp. 6230–6239.

[27] L.-C. Chen et al., “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 4, pp. 834–848, 2018.

[28] V. Badrinarayanan et al., “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 12, pp. 2481–2495, 2017.

[29] O. Ronneberger et al., “U-Net: Convolutional Networks for Biomedical Image Segmentation,” Medical Image Computing and Computer-Assisted Intervention, vol. 9351, pp. 234–241, 2015.

[30] A. Paszke et al., “ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation,” 2016. arXiv: [1606.02147v1] [Online]. Available: [https://arxiv.org/pdf/1606.02147.pdf](https://arxiv.org/pdf/1606.02147.pdf).

[31] E. Romera et al., “ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation,” IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 1, pp. 263–272, 2018.

[32] M. Sandler et al., “MobileNetV2: Inverted Residuals and Linear Bottlenecks.” In Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510–4520.

[33] S. Mehta et al., “ESPNet: Efficient Spatial Pyramid of Dilated Convolutions for Semantic Segmentation.” In Proc. of the European Conference on Computer Vision, 2018, pp. 561–580.

[34] S. Mehta et al., “ESPNetv2: A Light-weight, Power Efficient, and General Purpose Convolutional Neural Network.” In Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 9190–9200.

[35] C. Elkan and K. Noto, “Learning Classifiers from Only Positive and Unlabeled Data.” In Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2008, pp. 213–220.

[36] Y. Yang et al., “Object detection as a positive-unlabeled problem,” 2020, pp. 1–17. arXiv: [2002.04672](https://arxiv.org/pdf/2002.04672.pdf).

[37] M. Labbé and F. Michaud, “RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation,” Journal of Field Robotics, vol. 36, no. 2, pp. 416–446, 2018.

[38] Z. Zheng and Y. Yang, “Rectifying Pseudo Label Learning via Uncertainty Estimation for Domain Adaptive Semantic Segmentation,” International Journal of Computer Vision, 2021.

[39] G. J. Brostow et al., “Semantic object classes in video: A high-definition ground truth database,” Pattern Recognition Letters, vol. 30, no. 2, pp. 88–97, 2009.

[40] M. Cordts et al., “The Cityscapes Dataset for Semantic Urban Scene Understanding.” In Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2016, pp. 3213–3223.

[41] A. Valada et al., “Deep Multispectral Semantic Scene Understanding of Forested Environments Using Mul-
timodal Fusion.” In *Proc. of the International Symposium for Experimental Robotics*, 2017, pp. 465–477.

[42] L. N. Smith, “Cyclical learning rates for training neural networks.” In *Proc. of the IEEE Winter Conference on Applications of Computer Vision*, 2017, pp. 464–472.