Online Refinement of a Scene Recognition Model for Mobile Robots by Observing Human’s Interaction with Environments

Shigemichi Matsuzaki\textsuperscript{1}, Hiroaki Masuzawa\textsuperscript{1}, and Jun Miura\textsuperscript{1}

Abstract—This paper describes a method of online refinement of a scene recognition model for robot navigation considering \textit{traversable plants}, flexible plant parts which a robot can push aside while moving. In scene recognition systems that consider traversable plants growing out to the paths, misclassification may lead the robot to getting stuck due to the traversable plants recognized as obstacles. Yet, misclassification is inevitable in any estimation methods. In this work, we propose a framework that allows for refining a semantic segmentation model on the fly during the robot’s operation. We introduce a few-shot segmentation based on weight imprinting for online model refinement without fine-tuning. Training data are collected via observation of a human’s interaction with the plant parts. We propose novel robust weight imprinting to mitigate the effect of noise included in the masks generated by the interaction. The proposed method was evaluated through experiments using real-world data and shown to outperform an ordinary weight imprinting and provide competitive results to fine-tuning with model distillation while requiring less computational cost.

I. INTRODUCTION

Navigation in plant-rich environments is a challenging problem due to the presence of plant parts covering the paths. In scene recognition using range sensors, which are widely adopted in conventional mobile robots, those plant parts are recognized as obstacles blocking the paths. Some of such plant parts are, however, flexible and a robot can move through them by pushing them aside. We call such plants \textit{traversable plants} in this work.

Presence of traversable plants in the context of robot navigation has not been dealt in many studies. A pioneering work by Kim et al. [1] estimated the traversability of the regions in an image using color and geometric features. The authors’ previous work [2] proposed a novel deep neural network (DNN) based on semantic segmentation for object traversability estimation with a manual annotation-free training method. Although we demonstrated the effectiveness of the method through navigation experiments in a plant-rich greenhouse, misclassifications were a critical problem. Fig. 1 shows an example of a case where some voxels of traversable plants are misclassified as other obstacles and the path is considered blocked. When such a misclassification often occurs, the robot easily gets stuck. Such misclassifications are inevitable in scene recognition methods including DNNs.

A method to deal with them on the fly is, therefore, needed for practical uses of such a navigation system.

To this end, we are developing a method for online refinement of the recognition model exploiting a human’s interaction with the misclassified traversable plant regions during the actual operation of mobile robots. Ideally, the robot should indicate possibly misclassified regions to the human with some interfaces to acquire data efficiently. In this work, however, we consider a situation where the human randomly touches the plant parts. We propose a framework for refining the semantic segmentation model making use of the information acquired through such interaction. The proposed method will be a building block of more advanced system for traversable plant recognition with a function for online refinement of the recognition model.

In the proposed method, a human first touches regions of plants in the 3D space to provide the robot with information to correct the recognition of plants. The robot observes the interaction using a human pose detector such as OpenPose [3] to identify plant regions indicated by the human. Masks of the interacted regions in images are generated for the model refinement. The scene recognition model is then refined using the mask so that next time the robot encounters similar scenes, the robot recognizes the scenes correctly. More specifically, we introduce a few-shot learning method based on weight imprinting [4]. Since the masks of interacted regions are generated using depth images from an RGB-D sensor, we add a modification to the process of weight imprinting to deal with noise included in the masks.

The contributions of this work are as follows:
i) A novel framework of online refinement of a scene recognition model exploiting data acquired during the robot’s operation using a human’s interaction with the environment.

ii) A novel weight imprinting-based few-shot segmentation that is robust to noise in the masks.

II. RELATED WORK

A. Traversability estimation

Traversability estimation is an essential task in navigation of mobile robots. Navigation in unstructured environments requires traversability estimation based on the terrain condition such as steepness and roughness. Major approaches include terrain analysis using range sensors [5, 6, 7] and image-based estimation of terrain semantics [8, 9, 10].

Some methods have been introduced for analyzing the traversability of flexible objects such as plants. Kim et al. [1] trained a classifier with hand-crafted color and geometric features using a robot’s experience of traversals. We presented an image-based deep neural network for object traversability estimation and its manual annotation-free training method [2]. In [2], we reported some failure cases of navigation due to misclassifications. Such machine learning-based methods inevitably suffer from misclassifications. To the best of our knowledge, there has been no method proposed to deal with the problem on the fly during the operation in the context of robot navigation.

B. Image segmentation via interaction with objects

In the field of robotics, many studies have proposed to leverage the robots’ active interaction with objects to achieve information, which is not available in passive observation [11, 12, 13]. Some other studies have also tackled problems of learning an object segmentation model using a human’s interaction with the target objects as supervisory signals. Arsenio et al. [14] proposed an object segmentation method through robot’s own actions, actions of a person with a wearable system, or perceiving demonstrations of a human. Kim et al. [15] proposed a method for identification of objects to segment through a human’s gesture. Azagra et al. [16] also proposed a method to indicate objects to learn to the robot via a human’s gesture.

Existing work is focused on segmentation of rigid body objects. In contrast, our purpose is to identify plant regions especially flexible parts such as leaves and branches and to refine misclassifications of the recognition model on the fly.

C. Semantic segmentation with unseen classes

Adapting a model to unseen classes of objects is an actively studied task. Tasks that involves such objective include incremental learning, few-shot learning, continual learning, lifelong learning etc. [17].

Incremental learning is a task to adapt a pre-trained model to newly seen data while maintaining previously learned knowledge avoiding catastrophic forgetting. Knowledge distillation [18], or model distillation, is a widely used approach to the task [17, 19, 20]. Although the task tackled in the aforementioned studies is relevant to ours, those methods require heavy computation such as backpropagation, which is not suitable given limitation of computational resources available on mobile robots.

Few-shot learning is one of machine learning paradigms where limited new samples with ground truth labels (support set) are given for reference to make inference on data with object classes unseen in pre-training (query set). In few-shot segmentation, a support set with ground truth mask is exploited to aggregate features and the aggregated features are used to produce classification weights for segmentation of a query image [21, 22, 23].

Qi et al. [4] proposed a simple method for few-shot image classification coined weight imprinting, where normalized features of a target object are directly used as weights of the classifier. They claim that weight imprinting performs well even without fine-tuning. Based on the weight imprinting approach, Siam et al. [24] proposed the Adaptive Masked Proxies (AMP) for effective feature aggregation specifically designed for few-shot image segmentation. Our online learning is inspired by [4] and [24] for their simplicity.

III. REVIEW OF THE BASE METHODS

A. Traversability estimation

Our previously proposed scene recognition model considering traversable plants [2] consists of two modules; Semantic Segmentation Module (SSM) and Traversability Estimation Module (TEM). SSM provides information of general object classes, namely plant, artificial object, and ground. TEM is a pixel-wise class-agnostic binary classifier that estimates the traversability of each pixel taking a feature map from SSM as input.

The estimation of the model is projected to the 3D space using a depth image registered to the RGB image. The 3D space is then divided into voxels with a fixed size and the object probability and the traversability are calculated for each voxel. The estimation in each frame temporally fused via Bayesian update.

In the method, misclassification of voxels led the robot to getting stuck (see Fig. 1). We reported multiple failures of navigation due to misclassifications in [2]. The purpose of the present work is to amend the problem by enabling online refinement of the recognition model during the robot operation. Because the main cause of voxel misclassification observed in [2] was incorrect prediction of object classes by SSM, we focus on refinement of SSM in this work.

B. Weight imprinting

Weight imprinting [4] was first proposed as a method for few-shot image classification. Given an embedding extractor $\phi : \mathbb{R}^N \rightarrow \mathbb{R}^D$ which maps an input $x \in \mathbb{R}^N$ to a $D$-dimensional vector $\phi(x) \in \mathbb{R}^D$, and a softmax classifier $f : \mathbb{R}^D \rightarrow \mathbb{R}^C$
which produces probability values for $C$ classes, a probability for class $j$ is given as follows:

$$f_j(\phi(x)) = \frac{\exp(W_j^T \phi(x))}{\sum_{j=1}^{C} \exp(w_j^T \phi(x))},$$  \hspace{1cm} (1)$$

where $W_j$ denotes the weight vector for class $j$. Here, both the embedding $\phi(x)$ and the weight vector $w_j$ are assumed to be L2-normalized. Then, the main computation in eq. (1) is taking the inner product, or equivalently the cosine similarity between $\phi(x)$ and $w_j$. Viewing each weight as a template vector of the corresponding class, the authors argued that the prediction is equivalent to finding the template closest to the embedding $\phi(x)$ based on the cosine similarity in the embedding space. Based on the observation, they proposed to use the embedding for input data of a novel class as the classification weight of the class.

Siam et al. [24] applied weight imprinting to few-shot semantic segmentation. Given the intermediate feature map $x_i \in \mathbb{R}^{H \times W \times D}$, with the height $H$ and the width $W$, and a corresponding binary mask $M_i \in \{0,1\}^{H \times W}$, they aggregate embeddings over a given mask via simple averaging followed by L2-normalization (masked average pooling: MAP):

$$\bar{x}_{MAP} = \frac{\sum_{i=1}^{N} M_i(h,w)x_i(h,w)}{\sum_{i=1}^{N} \sum_{h=1}^{H} \sum_{w=1}^{W} M_i(h,w)}, \hspace{1cm} \bar{x}_{MAP} = \frac{\bar{x}_{MAP}}{\|\bar{x}_{MAP}\|_2},$$  \hspace{1cm} (2)$$

where $(h,w)$ denotes the coordinate of the image pixel. The normalized average embedding $\bar{x}_{MAP}$ is used as classification weights of the novel class. In [24], they proposed to apply it to multiple layers to exploit multi-resolution information, which is, in general, considered important for semantic segmentation. In our method, however, we apply weight imprinting only on the last classification layer for simplicity of implementation.

IV. ONLINE MODEL REFINEMENT

Our aim is to refine the scene recognition model on the fly with data collected during the robot’s operation. General fine-tuning is not suitable due to limitations of computational resources and of the variety of image data acquired during the operation. We, therefore, adopt a few-shot segmentation method based on weight imprinting [4], which does not require fine-tuning with costly backpropagation.

A. Problem setting

We assume that the scene recognition model $f(I)$ is pre-trained with a large amount of data before deploying the robot in a target environment following [2] with $C$ classes. We then assume to acquire a few pairs of an RGB image and a corresponding binary mask $S = \{(I_i, M_i)\}_{i=1}^{N}$, where $N$ denotes the number of the pairs. The value 1 in the binary mask $M_i \in \{0,1\}^{H \times W}$ indicates a pixel that should be learned as plant and the mask is acquired through interaction by the human with the environment during the robot operation.

The process of generating the mask is described in the next subsection.

Our purpose is to update the model on the fly with the image-mask pairs $S$ so that the model predicts the plant parts more accurately. As mentioned above, we formulate the problem of the online learning as a few-shot learning problem where the objects of the masked regions are learned as a novel $C + 1$th class unseen in the pre-training.

B. Data collection

For few-shot learning, a pair of an input image $I$ and a corresponding mask $M$ is generated based on the human’s interaction. The interaction mask indicates image regions that should be classified as plant.

1) Labeling plant regions via a human’s interaction: First, a human interacts with the regions of plant parts. A robot observes the human’s interaction using an RGB-D camera. The human joints are recognized by OpenPose [3]. The coordinate of the right hand is projected into the 3D space using the depth. Voxels with which the human interacted are identified by searching for voxels within a sphere with a radius of 5 [cm] centered at the projected hand coordinate.

2) Generating a training mask: After the map labeling, a mask $M$ for training is generated by the process as follows.

i) Get a pair of RGB and depth images.

ii) For each pixel of the depth image, project it in the 3D space and search for a voxel that the pixel belongs to.

iii) Label the pixel as 1 if the corresponding voxel has been interacted, and otherwise 0.

iv) Repeat i) to iii) for five consecutive frames and take pixel-wise OR of them to deal with the depth noise. We call the resulting mask an interaction mask denoted as $M'$.

v) Predict object labels on the RGB image $I$ using the network and set pixels of $M'$ to 0 if the predicted object label of the corresponding pixels are other than plant (see Fig. 3).

We call the resulting mask $M$ a training mask. Here we assume that the interacted regions in $M'$ are dominated
by plants. The training mask $M$, therefore, indicates false negative plant regions.

C. Network architecture and the loss function

We adopt the network architecture proposed in [2] as a base network. For the entire architecture of the original network, readers are referred to [2], Fig. 8. Although we adopt the architecture in this work, note that the proposed method is not dependent on a specific network architecture. To apply weight imprinting-based few-shot learning, we make two modifications on the network architecture for semantic segmentation and the loss function.

First, we apply L2 normalization on the weight vectors of the final classification layer $\{W_i\}_{i=1}^N$, which is a $1 \times 1$ convolution, and the features $\{x_i^{(h,w)}\}_{i=1}^N$, which is a feature vector in pixel $(h,w)$ of an intermediate feature map yielded by forwarding an image $I_i$ through the model $f$. The score of the features for each class is then evaluated as a cosine similarity between the features and the weight vector of the class [4].

Second, to formulate the softmax loss on the normalized data in a more natural way and also to encourage the model to learn discriminative feature representations, we introduce the Additive Angular Margin Loss (ArcFace) [25] defined as follows:

$$L_{arc} = -\sum_{i=1}^{N} \log \frac{e^{s\cos(\theta_i + m)}}{\sum_{j=1, j\neq y_i}^{C} e^{s\cos\theta_j}},$$

(3)

where $\cos \theta_i = W^T \hat{x}_i$, $m$ denotes an angular margin parameter, $y_i$ denotes an object label of $\hat{x}_i$, and $s$ denotes a scaling factor that controls smoothness of the predicted probability distribution. By setting the angular margin $m$ for the score of the correct class, the network is trained so that the intra-class distances of the feature distributions become large and thus discriminative features are learned.

D. Online learning by the robust weight imprinting

For online learning of segmentation, we adopt a few-shot learning based on weight imprinting [4][24]. Suppose we have $N$ pairs of an input image and a training mask $\{(I_i, M_i)\}_{i=1}^N$ and an intermediate feature map $\hat{x}_i$ is yielded by passing the image $I_i$ through the segmentation network.

Here, we point out a problem of the training mask generated by the procedure described in the previous subsection. Because of the depth noise and registration error of the RGB-D sensor, the training masks include regions set as 1 and not part of the plant regions. An example of erroneous mask is shown in Fig. 4. MAP described in Sec. III-B is prone to those outliers since features are equally averaged. In our method, we replace it with weighted averaging based on the distance from the center of the feature distribution within the mask, which we call robust average pooling (RAP):

$$\hat{x}_{\text{RAP}} = \frac{\sum_{d=1}^{N} \sum_{h=1}^{H} \sum_{w=1}^{W} v_{i}^{(h,w)} M_{i}^{(h,w)} x_{i}^{(h,w)}}{\sum_{d=1}^{N} \sum_{h=1}^{H} \sum_{w=1}^{W} M_{i}^{(h,w)}},$$

(4)

where $v^{(h,w)}$ denotes the weight on the corresponding feature $x^{(h,w)}$ calculated as follows:

$$v_{i}^{(h,w)} = \begin{cases} x_{i}^{(h,w)} \cdot \mathbf{MAP} & \text{if } x_{i}^{(h,w)} \cdot \mathbf{MAP} \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

(5)

The averaged feature is then L2-normalized to form a convolution weight vector for the novel class:

$$\hat{x}_{\text{RAP}} = \frac{\hat{x}_{\text{RAP}}}{\|\hat{x}_{\text{RAP}}\|_2}.$$  

(6)

We directly use $\hat{x}_{\text{RAP}}$ as a weight vector of the classification layer for the new class representing false negative plant regions in the model before the training.

V. EXPERIMENTS

A. Experimental setup

The DNN model is implemented with PyTorch [26]. The pre-training and inference are performed on one NVIDIA GeForce GTX 1080Ti with 11GB of memory. Before the online learning, we pre-train the scene recognition model in a method proposed in [2], with modification of the model.
TABLE I
PER-CLASS AND MEAN IOU BEFORE AND AFTER THE TRAINING

|                | Plant | Artificial obj. | Ground | mIoU |
|----------------|-------|-----------------|--------|------|
| Before         | 80.89 | 84.06           | 60.04  | 75.00|
| MD             | 80.88 | 85.90           | 63.23  | 76.67|
| WI-MAP         | 79.71 | 85.04           | 60.04  | 74.93|
| WI-RAP (proposed) | 81.19 | 85.49           | 60.05  | 75.57|

Bold denotes the best and underline denotes the second best results.

and the loss function described in Sec. IV-C. The angular margin $m$ is set to 0.1.

We use five pairs of an image and a mask taken in different scenes for few-shot learning. For testing, we exploit 15 manually labeled images of scenes where influence of misclassification can be significant, such as the end of the plant rows, as reported in [2].

B. Baselines

As baselines, we adopt weight imprinting with the ordinary masked average pooling and also a fine-tuning approach via model distillation. In the former, the averaged feature over the mask (eq. (2)) is used to calculate the classification weight instead of eq. (4).

In model distillation, we adopt output-level distillation described in [17]. We set a weight for distillation loss to 0.5 and temperature parameter $T$ for the softmax function to 1.0. The entire network is optimized. We utilize the full mask of interacted regions since the task is fine-tuning with the existing plant class, instead of learning a novel class. The batch size is set to 5 and a constant learning rate of $1 \times 10^{-4}$ is used. The model is trained for 15 epochs and we report the results on the epoch with the best mean IoU.

We hereafter denote the proposed robust weight imprinting as WI-RAP, the weight imprinting with the ordinary masked average pooling as WI-MAP, and the model distillation method as MD.

C. Online model refinement

Table I shows the comparison of IoU before and after the online learning as well as the results of model distillation. While WI-MAP resulted in degrading the performance, the proposed robust pooling led to better IoU. This qualitatively shows the advantage of our method. MD resulted in better mean IoU and per-class IoU on the artificial object and ground classes than the proposed method. The proposed method, however, resulted in better per-class IoU on the plant class than MD. Since our main purpose is to improve the accuracy on plant regions, this result shows better suitability of the proposed method to the online refinement.

We further look into the precision and the recall metrics. Table II shows the precision and the recall of each method. Our primary purpose is to refine the false negative plant regions which is relevant to the recall metric. The both few-shot learning approaches resulted in better recall of plant regions. The improvement of the recall, however, came at a cost of degradation of the precision. Compared to WI-MAP, the proposed robust imprinting, WI-RAP, yielded a comparative gain of recall (-0.92 from WI-MAP) with less degradation of precision (+2.44 from WI-MAP).

Fig. 5 shows the qualitative results. After the online training by the proposed method, false negative plant regions are corrected to the plant class. Especially, the regions in black circles show obvious improvement of segmentation. In those regions, detailed shapes of the leaves and the branches are better captured compared to the segmentation before training. Compared to WI-MAP, there are less false positives of prediction as plant in the predictions of the proposed WI-RAP. This is because while WI-MAP aggressively involves outliers in feature aggregation, WI-RAP robustly calculate the imprinted weights by the weighted averaging.

One may notice that MD resulted in better performance. It, however, required around 20 [sec] of training (excluding in-training validation) and 5 [GB] of GPU memory consumption. Although the batch size of 1 decreased the memory consumption to about 1.5 [GB], the results were worse. In contrast, the proposed method took 0.38 [sec] and consumed approx. 700 [MB] because it requires only one forward calculation for each training image. It is, therefore, more suitable to on-the-fly model refinement on modern onboard computers with GPU-acceleration such as NVIDIA Jetson.

D. Parameter evaluation

Next, we evaluate the effect of the angular margin $m$ in the ArcFace loss. Table III shows the relationship between $m$ used in model pre-training and the performance of few-shot learning. When $m = 0.0$, i.e., no angular margin constraint is imposed, few-shot learning resulted in significant degradation of mean IoU. This may be because when the features are learned without the angle margin, inter-class distances of the features are not encouraged to be larger, and thus the learned features are not discriminative enough with large overlaps. It will lead to confusion of features to be imprinted and

The results are shown in the order of recall / precision. Bold denotes the best and underline denotes the second best results.

| MD | WI-MAP | WI-RAP (proposed) |
|----|--------|-------------------|
| 92.21 | 86.80 | 92.03 |
| 93.77 | 84.16 | 94.23 |
| 92.85 | 86.60 | 93.34 |

TABLE II
RECALL AND PRECISION BEFORE AND AFTER THE TRAINING

|                | Plant | Artificial obj. | Ground |
|----------------|-------|-----------------|--------|
| Before         | 89.02 | 89.86           | 90.41  |
| MD             | 92.21 | 86.80           | 92.03  |
| WI-MAP         | 93.77 | 84.16           | 94.23  |
| WI-RAP (proposed) | 92.85 | 86.60           | 93.34  |

The results are shown in the order of recall / precision. Bold denotes the best and underline denotes the second best results.

| m | Before | 0.0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
|---|--------|-----|-----|-----|-----|-----|-----|
|   | 74.89  | 75.00 | 74.46 | 72.77 | 74.25 | 66.61 |
| WI-MAP | 41.47  | 74.93 | 68.29 | 72.97 | 74.55 | 66.99 |
| WI-RAP | 46.80  | 75.57 | 74.96 | 72.76 | 74.09 | 67.00 |
Fig. 5. Qualitative evaluation of the online learning. After the training, the shapes of plant parts are better captured especially in the regions in the black circles. Compared to WI-MAP, there are less false positive predictions as plant class in the predictions of the proposed WI-RAP. MD provided similar or better predictions. The proposed method, however, resulted in similar results with only forward pass unlike MD which requires backpropagation.

VI. CONCLUSIONS AND FUTURE WORK

We proposed a novel framework of online model refinement to deal with misclassification which may lead the robot to be stuck during navigation. We introduced a few-shot segmentation method based on weight imprinting, which allows for model refinement on the fly. Masks for the few-shot training are generated through observation of a human operator interacting with plant parts in the environment. To mitigate the effect of inaccurate masks due to depth noise, we proposed robust weight imprinting and showed that the proposed method outperformed the ordinary weight imprinting with masked average pooling in terms of IoU.

As future work, we are going to work on more detailed analysis of the relationship of intermediate feature representation and the performance of the online learning. We also consider adopting the proposed method in the navigation system and conduct real-world experiments. In addition, designing a better interface for human-robot interaction to indicate misclassified plant regions more efficiently is also under consideration.

REFERENCES

[1] 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.

[2] S. Matsuzaki et al., “Image-Based Scene Recognition for Robot Navigation Considering Traversable Plants and Its Manual Annotation-Free Training,” IEEE Access, vol. 10, pp. 5115–5128, 2022.

[3] Z. Cao et al., “OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 1, pp. 172–186, 2021.

[4] H. Qi et al., “Low-Shot Learning with Imprinted Weights.” In Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 5822–5830.
I. Bogoslavskyi et al., “Efficient traversability analysis for mobile robots using the Kinect sensor.” In Proc. of the European Conference on Mobile Robots, 2013, pp. 158–163.

J. L. Martínez et al., “Reactive navigation on natural environments by continuous classification of ground traversability,” Sensors, vol. 20, no. 22, 2020.

Y. Zhou et al., “3D Traversability Map Generation for Mobile Robots Based on Point Cloud.” In Proc. of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics, IEEE, 2021, pp. 836–841.

S. Matsuzaki et al., “Traversable Region Estimation for Mobile Robots in an Outdoor Image,” Journal of Intelligent & Robotic Systems, vol. 92, no. 3-4, pp. 453–463, 2018.

D. Maturana et al., “Real-time Semantic Mapping for Autonomous Off-Road Navigation,” Field and Service Robotics, vol. 5, pp. 335–350, 2018.

S. Hosseinpoor et al., “Traversability Analysis by Semantic Terrain Segmentation for Mobile Robots.” In Proc. of the IEEE International Conference on Automation Science and Engineering, IEEE, 2021, pp. 1407–1413.

W. H. Li and L. Kleeman, “Segmentation and modeling of visually symmetric objects by robot actions,” International Journal of Robotics Research, vol. 30, no. 9, pp. 1124–1142, 2011.

D. Katz et al., “Interactive segmentation, tracking, and kinematic modeling of unknown 3D articulated objects.” In Proc. of the IEEE International Conference on Robotics and Automation, 2013, pp. 5003–5010.

T. Novkovic et al., “Object Finding in Cluttered Scenes Using Interactive Perception,” Proceedings - IEEE International Conference on Robotics and Automation, pp. 8338–8344, 2020.

A. Arsenio et al., “The Whole World in Your Hand: Active and Interactive Segmentation.” In Proc. of the International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems, 2003, pp. 49–56.

S. Kim et al., “On-line object segmentation through human-robot interaction.” In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010, pp. 1734–1739.

P. Azagra et al., “Incremental Learning of Object Models from Natural Human-Robot Interactions,” IEEE Transactions on Automation Science and Engineering, vol. 17, no. 4, pp. 1883–1900, 2020.

U. Michieli and P. Zanuttigh, “Knowledge distillation for incremental learning in semantic segmentation,” Computer Vision and Image Understanding, vol. 205, p. 103 167, 2021.

G. Hinton et al., “Distilling the Knowledge in a Neural Network.” In Proc. of the Deep Learning and Representation Learning Workshop, 2015.

F. Cermelli et al., “Modeling the background for incremental learning in semantic segmentation.” In Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 9230–9239.

M. Klingner et al., “Class-Incremental Learning for Semantic Segmentation Re-Using Neither Old Data Nor Old Labels.” In Proc. of the IEEE International Conference on Intelligent Transportation Systems, 2020.

A. Shaban et al., “One-shot learning for semantic segmentation.” In Proc. of the British Machine Vision Conference, 2017, pp. 167.1–167.13.

K. Nguyen and S. Todorovic, “Feature weighting and boosting for few-shot segmentation.” In Proc. of the IEEE International Conference on Computer Vision, 2019, pp. 622–631.

G. Li et al., “Adaptive Prototype Learning and Allocation for Few-Shot Segmentation.” In Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, 2021, pp. 8330–8339.

M. Siam et al., “AMP: Adaptive masked proxies for few-shot segmentation.” In Proc. of the IEEE International Conference on Computer Vision, 2019, pp. 5248–5257.

J. Deng et al., “ArcFace: Additive Angular Margin Loss for Deep Face Recognition.” In Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, 2019, pp. 4685–4694.

A. Paszke et al., “PyTorch: An Imperative Style, High-Performance Deep Learning Library.” In Proc. of the Conference on Neural Information Processing Systems, 2019, pp. 8024–8035.