2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Real-Time Semantic Segmentation in Natural Environments with SAM-assisted Sim-to-Real Domain Transfer

Han Wang, Ruben Mascaro, Margarita Chli, Lucas Teixeira
Vision For Robotics Lab, ETH Zürich and University of Cyprus

***Abstract*— Semantic segmentation plays a pivotal role in many robotic applications requiring high-level scene understanding, such as smart farming, where the precise identification of trees or plants can aid navigation and crop monitoring tasks. While deep-learning-based semantic segmentation approaches have reached outstanding performance in recent years, they demand large amounts of labeled data for training. Inspired by modern Unsupervised Domain Adaptation (UDA) techniques, in this paper, we introduce a two-step training pipeline specifically tailored to challenging natural scenes, where the availability of annotated data is often quite limited. Our strategy involves the initial training of a powerful domain adaptive architecture, followed by a refinement stage, where segmentation masks predicted by the Segment Anything Model (SAM) are used to improve the accuracy of the predictions on the target dataset. These refined predictions serve as pseudo-labels to supervise the training of a final distilled architecture for real-time deployment. Extensive experiments conducted in two real-world scenes demonstrate the effectiveness of the proposed method. Specifically, we show that our pipeline enables the training of a MobileNetV3 that achieves significant mIoU gains of 3.60% and 11.40% on our two datasets compared to the DAFormer while only demanding 1/15 of the latter's inference time. Code and datasets are available at `https://github.com/VIS4ROB-lab/nature_uda_rt_segmentation`.**

I. Introduction

Being a fundamental task in computer and robot vision, semantic segmentation aims at assigning a class label to every pixel in an image, providing a detailed understanding of the observed scene. This usually lays the foundation for high-level reasoning in various downstream applications for intelligent vehicles, robotics [1], and agriculture.

Over the past decade, research in semantic segmentation has seen prominent advances due to the flourishing of Convolutional Neural Networks (CNNs) [2] and, more recently, transformer architectures [3]. However, the regular supervised training of these models heavily relies on the availability of abundant labeled data, whose acquisition is extremely labor-intensive, and inference performance is quite sensitive to domain shifts. These aspects severely hinder the application of semantic segmentation networks in real-world scenarios. To mitigate these challenges, a promising approach, referred to as Unsupervised Domain Adaptation (UDA), has emerged in the literature for transferring knowledge from a labeled source domain to an unlabeled target domain during training. While numerous UDA strategies have been proposed in recent years, ranging from adversarial training [4], [5], [6], [7] to self-training [8], [9], [10], [11], advancements in this field have been typically showcased using large, powerful architectures and mainly evaluated on autonomous driving datasets.

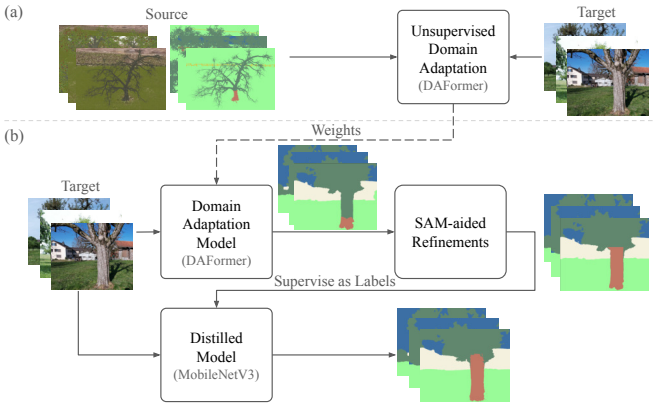


Fig. 1: Our proposed two-step domain adaptation pipeline. (a) The first phase performs regular UDA training of a powerful architecture. (b) Then, the model's predictions on the target domain are refined offline with the help of SAM and used as pseudo-labels to supervise the training of the final distilled model.

Here, on the contrary, we are particularly interested in facilitating the realistic deployment of semantic segmentation models in natural scenes, where the availability of annotated data for training is quite limited. Performing segmentation in these types of environments is necessary for agricultural and forestry purposes, such as crop monitoring, precise farming or creating digital twins of trees. However, as opposed to autonomous driving scenarios that feature somewhat repetitive scene layouts, datasets captured in natural environments might contain substantial variability regarding viewpoints and appearance (e.g. caused by seasonal changes), making it more difficult for semantic segmentation models to generalize to novel scenes. In addition, especially in robotic applications, using lightweight models is often mandatory due to real-time requirements or limited computing resources. Overall, these challenges remain largely unexplored in existing literature.

Therefore, in this paper, we bridge the gap between existing generic UDA approaches for semantic segmentation and their practical application to natural scenes, specifically targeting lightweight models that can run on edge devices. To this end, we propose a two-step training pipeline. In the first step, regular UDA training of a large model is performed. Next, the predictions on the target domain are obtained and refined offline using the high-quality segmentation masks predicted by Segment Anything Model (SAM) [12]. These

This work has been partly funded by the European Research Council (ERC), as part of the project SkEyes (Grant agreement no. 101089328) and by Unity Technologies.

refined predictions are used as pseudo-labels to supervise the training of a lightweight network. We test our approach for sim-to-real adaptation, using images rendered from a simulated tree-covered area as source domain and two different real-world datasets as target domains. Trained with the refined labels, the lightweight MobileNetV3 consistently and considerably outperforms the domain adaptive architecture DAFormer's segmentation quality on both of target datasets, with noticeably 1/15 of the latter's inference time.

In brief, the contributions of this work are the following:

- We propose a novel two-step UDA training pipeline for semantic segmentation that focuses on improving the quality of the pseudo-labels and is particularly suitable for training lightweight architectures.
- We design a pseudo-label refinement scheme that leverages SAM to produce highly accurate segmentation masks in the target domain.
- We perform extensive experiments on sim-to-real scenes, with a simulated farm dataset and two real-world datasets comprising farm and park scenes, demonstrating the effectiveness of the training pipeline.

II. Related Work

A. Semantic Segmentation Architectures

For many years now, Convolutional Neural Networks (CNNs) [2], [13] have been widely used to address semantic segmentation tasks. In particular, the development of lightweight models, represented by MobileNetV3 [14], ICNet [15], Fast-SCNN [16] and BiSeNet [17], has enabled real-time applications or deployment on resource-constrained devices. Recently, transformer-based architectures have overtaken the dominant CNNs in the field [3]. However, the heavy computational load of attention mechanisms limits their deployment on edge devices. Latest advances alleviate this by proposing alternative attention mechanisms [18], [19], surpassing the performance of pure CNN models while maintaining or even reducing their latency.

One major obstacle to implementing modern semantic segmentation architectures, especially in highly specialized real-world applications, is the cost of obtaining large amounts of labeled training data. To address this, we intend to use unsupervised domain-adaptive training strategies.

B. Domain Adaptive Semantic Segmentation

Domain adaptive semantic segmentation aims at predicting semantics in new target scenarios without having access to ground-truth labels during training. Related works mainly concentrate on three directions to shrink the domain gap.

The first direction, adversarial learning, focuses on aligning the data distributions for both domains. These approaches [4], [20] model the task as a min-max adversarial optimization and enforce the model to learn common representations that fool the domain discriminator network. However, suboptimal results are usually reached by unstable adversarial training processes [21]. Moreover, adversarial pipelines can be difficult to train, as they call for a capacity balance between the domain discriminator and the feature encoder [22]. A style transfer pipeline for agricultural UDA semantic segmentation is introduced in [23], where ground-truth labels are still demanded for real-world images. The dataset only consists of low vegetation and terrain from top-down views, greatly limiting the application range.

The second technique exploits self-training to generate pseudo-labels for the target domain images. Sepico [24] proposes a one-stage adaptation framework, emphasizing the semantic concepts of individual pixels, while DACS [10] utilizes mixed images from both domains for training. DAFormer [11] significantly defeats the performance of CNN-based architectures by introducing transformer models and specific training tricks. Subsequent works HRDA [25] and MIC [26] further boost the performance by introducing multi-resolution training and masked image consistency as extra modules, respectively. Other approaches further exploit cross-domain knowledge by introducing pixel- and patch-wise contrast [27], using cross-domain attention [28], improving handling of imbalanced datasets [29], [30], letting the student in the self-training pipeline learn from the future teacher [31], or proposing an online pseudo-label refinement network [32]. Although all these strategies may indeed improve performance, they have mainly been demonstrated on common datasets consisting of autonomous driving scenes in urban areas. Here we focus on unstructured natural environments, where the segmentation quality can be affected by the wider variety of scene layouts and seasonal changes. Utilizing DAFormer as a starting point, our work bridges the gap between the generic approaches and a specific application by tailoring a set of pseudo-label refinement strategies to our particular use case.

The third category, which has emerged recently, adopts diffusion models to conduct domain translation. Peng *et al.* [22] suggest a semantic gradient guidance approach to assist the pixel-level label-guided image style transfer. Though outperforming DAFormer on standard datasets, it is still not open source for replicating or testing on our task.

C. Segment Anything Model

The Segment Anything Model (SAM) [12] is proposed to solve promptable segmentation tasks from interactive inputs, achieving impressive zero-shot segmentation results. SAM inspires several works, including MedSAM for medical image segmentation [33] and others that use it for refining labels in weakly supervised semantic segmentation settings [34]. However, the masks generated by SAM lack semantic labels, restricting SAM's direct contributions to semantic segmentation tasks. Although SEEM [35], following SAM's steps, is capable of providing semantic maps, the labels are restricted to the 80 COCO classes [36], which may not exactly align with the specific task in the target domain. Retraining a large foundation model, in this case, would also be expensive in terms of both data and time.

Semantic-Guided Mask Labeling (SGML) and the corresponding fusion methods are introduced in SAM4UDASS [37], boosting the UDA performance by refining the pseudo-labels at training time on autonomous driving datasets. Nev-

ertheless, the mask labeling approach imposes assumptions on the structure of a driving scene and thus is not directly applicable to natural environments. Instead of performing online pseudo-label refinement during training, here we explore the possibility of a time-efficient SAM-assisted refinement step after the adaptation, followed by the distillation to a domain-specific model.

III. Method

Our pipeline, as illustrated in Figure 1, consists of two steps: (a) domain adaptive training and (b) pseudo-supervised training with label refinements. Step (a) involves images $\mathbf{x}_s$, their corresponding semantic maps $\mathbf{y}_s$ from a labeled source domain as well as images $\mathbf{x}_t$ from an unlabeled target domain and utilizes unsupervised domain adaptation schemes to train an adaptation model. In step (b), the adaptation model provides initial predictions $\hat{\mathbf{y}}_{t,\mathrm{DA}}$ for target data $\mathbf{x}_t$. Refinements to $\hat{\mathbf{y}}_{t,\mathrm{DA}}$ are conducted to improve the quality with the assistance of SAM. The refined predictions then supervise the training of a distilled architecture, following a regular approach for semantic segmentation.

A. Domain Adaptive Training

Our target datasets comprise real-world images with six valid classes: *sky*, *terrain*, *trunk*, *canopy*, *building* and *others*. As a labeled source domain dataset is demanded to initiate training, we generated synthetic images with ground-truth labels using a state-of-the-art agricultural scene simulator [38] that uses realistic models of trees and terrain. Furthermore, to serve downstream tasks, the tree models are split into *trunk* and *canopy* at the point where the main vertical stem stops and the primary branches begin. To mitigate the issue of the model ignoring thin branches, the canopy area in the annotations is dilated. Moreover, as the building class is missing in the generated dataset, we introduce samples of buildings by means of copy-paste augmentations [39] at training time.

The domain adaptation process follows the state-of-the-art self-training scheme that consists of two models with identical architecture – a student g_θ and a teacher h_ϕ, where θ and ϕ stand for their corresponding weights. At each step t, the student network is initially trained on the source data, supervised by a categorical cross-entropy loss. The teacher makes predictions (a.k.a. pseudo-labels) on the unaugmented target data. To improve reliability, a confidence threshold is defined to filter out potentially incorrect pseudo-labels. Simultaneously, the student processes the corresponding augmented, cross-domain mixed images and updates itself based on pseudo-label supervision. Finally, the teacher model gets updated as $\phi_{t+1} \leftarrow \alpha\phi_t + (1-\alpha)\theta_t$ with a predefined parameter α.

B. SAM-aided Pseudo-label Refinement

After the initial domain adaptation training stage, we refine the predictions $\hat{\mathbf{y}}_{t,\mathrm{DA}}$ yielded by the adaptation model on the target images. The goal is to increase the accuracy of the segmentation masks so that these can be employed as pseudo-labels to supervise the training of a final distilled model.

Algorithm 1: Mask Fusion for *Region Correction*

Data: 2D masks $\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_m$, sets of 2D points $S_1, S_2, \ldots, S_m$
Result: Updated boolean 2D mask $\hat{y}$
Initialize a boolean 2D mask $\hat{y}$ with all False
for $i = 1$ **to** m **do**
  if $\exists p \in S_i$ *such that* $\hat{y}$ *at point* p *is* False **then**
    $\hat{y} \leftarrow \mathtt{OR}(\hat{y}, \hat{y}_i)$
  end
end

Our refinement scheme considers the contextual information of images and handles imperfections with a high degree of complexity. The process is designed as a point-prompt-based refinement, where SAM is leveraged as a domain-agnostic tool for providing high-quality segmentation masks. Two categories of tasks specifically tailored to natural scenes are designed: *region correction* and *boundary correction*. The *region correction* task aims to clean large, uniform, yet misclassified areas of terrain and buildings. The *boundary correction* task focuses on refining the boundaries of trunk instances. This task can be further split into two steps – boundary sharpening and label recovery. The boundaries of trunks are updated by SAM's predictions, during which previously predicted trunks are annotated as "unlabeled". Some trunk instances, nevertheless, are included by mistake, and the recovery step focuses on their label re-assignments. Figure 2 illustrates a workflow for the overall process.

In each individual task of the process, only one class is concentrated on (we refer to it as the foreground class for that particular task). As the input to the task, a binary mask can be extracted from the segmentation prediction $\hat{\mathbf{y}}$. Consequently, each task yields binary masks, which can be fused back to the prediction $\hat{\mathbf{y}}$. Our *Prediction Fusion* strategy straightforwardly updates the positive entries of $\hat{\mathbf{y}}$ in the yielded mask with the corresponding class index.

1) Region Correction: This task aims to clean imperfections in large, uniform areas of terrain and buildings. In Figure 2, the scheme of cleaning terrain areas is demonstrated. In the first step, a boolean mask for the foreground class (i.e., terrain or building) is extracted from $\hat{\mathbf{y}}_{t,\mathrm{DA}}$, eroded, and then split into disconnected regions. For every discrete region, n pixels inside are selected using a Halton sequence with bases 2 and 3. Denote the point set of the i-th region as $S_i := \{p_{i1}, p_{i2}, ..., p_{in}\}$. A single mask $\hat{y}_i$ is then generated by SAM with S_i. The masks of all m regions are then fused according to Algorithm 1.

Additionally, when refining the buildings, an auxiliary binary mask, similarly extracted as the terrain class, is introduced, such that only the masked region (i.e., terrain) defined by the auxiliary mask is corrected to limit inaccurate or even erroneous enlarging of the predicted mask.

2) Boundary Correction: This task aims to correct the boundaries of trunk instances with SAM's sharp masks. An illustration is presented in Figure 2. The task consists of two

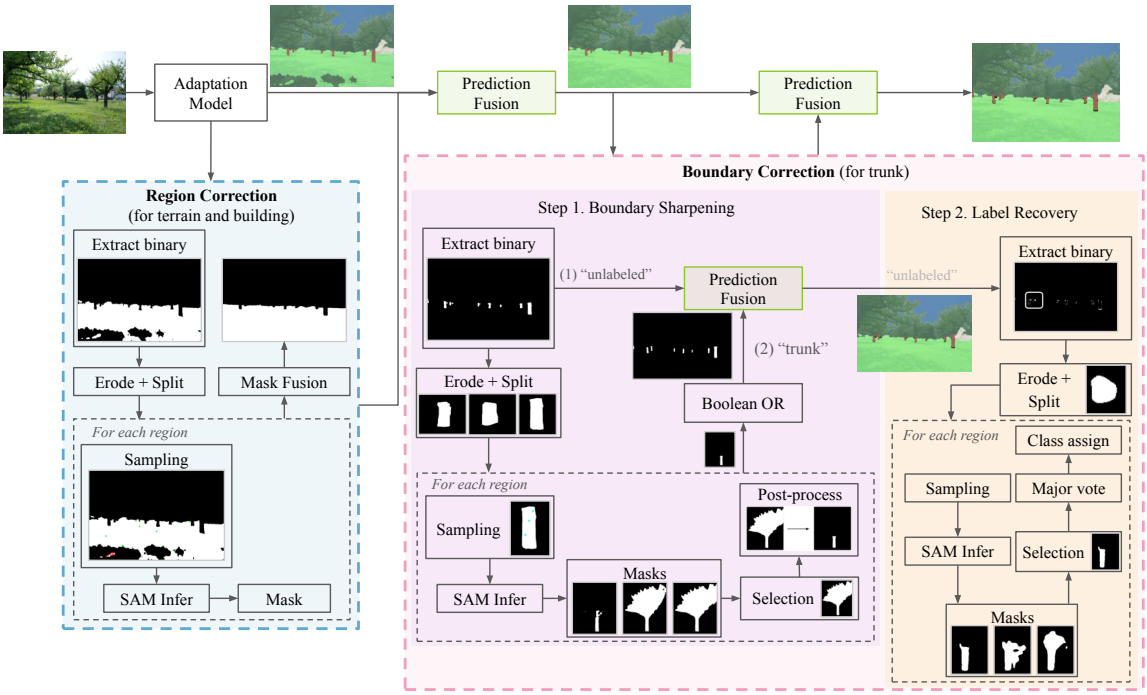


Fig. 2: Our SAM-aided pseudo-label refinement scheme. The approach incorporates two refinement tasks: *region* and *boundary correction*. The *region correction* eliminates errors in a large, uniform area by integrating inferences from split regions. In the *boundary correction* task, boundaries of trunk instances are sharpened with post-processed SAM predictions, followed by a recovery of labels that are marked as "unlabeled" by mistake. The refined masks are fused to the predictions step-by-step, and the entire process yields the refined pseudo-label.

steps: boundary sharpening and label recovering.

During the sharpening step, the binary extraction of the initial trunk masks follows the same erosion and split processes as in the *region correction*. However, here, the erosion is stronger to break accidental connections of close instances. Each disconnected region is then regarded as a trunk instance. Using pixel samples from each trunk, multiple masks are yielded from SAM's inference. Next, the largest discrete component among these masks is selected and fed to a geometric post-processing step aimed at removing canopy regions potentially included in it. Specifically, the post-processing step assumes that a tree has a "Y" or "T" shape, meaning that a sudden width variation is expected at the transition height from trunk to canopy. This characteristic enables a rough yet effective separation of trunk and canopy from a tree. The mask selection and post-processing is looped for all isolated trunk instances and the final output mask for the trunk class is achieved by boolean "OR".

To eliminate the inaccurate boundaries of the predicted trunks, the input binary mask is fused first to the refined prediction as "unlabeled", followed by the fusion of the sharpened trunk boundaries as "trunk".

The erosion process in the first step can cause some small trunk instances to be set to "unlabeled" by mistake. Therefore, the second step, label recovery, focuses on re-assigning those with labels. Figure 2 displays the process of a trunk being re-identified. In this case, we start by extracting the binary mask associated to the "unlabeled" class. After erosion, disconnected region splitting and region-by-region sampling, multiple candidate masks for each region are predicted with SAM. Denote the i-th eroded region mask as $\hat{y}_{r,i}$. The refined shape for this region is selected with the least amount of geometrically connected components and the smallest relative overlap ratio with the pixel-wise boolean "NOT" of $\hat{y}_{r,i}$ among the candidates. The label for re-assignment is determined by majority votes inside $\hat{y}_{r,i}$.

This task finishes with assigning all re-labeled regions to their corresponding votes and fusing them with the previous prediction.

IV. Experiments

A. Datasets

We perform experiments on two place-specific, real-world datasets: Kastelhof Apple Farm[1] and Andreas Park. We utilize both as target domains separately. To obtain ground-truth labels for evaluation, we run a 3D reconstruction of each scene using all the available images and annotate it manually. The annotations are then projected onto each image using the calibrated camera poses.

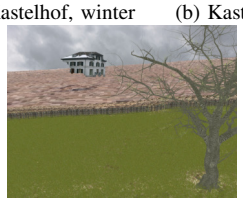
The Kastelhof dataset comprises 2572 images for training (winter 1581, summer 991) and 825 for validation (winter 469, summer 356). The depth maps are obtained for evaluation purposes, which are clipped at 255 m. In addition, the sky pixels are located at depth 0. The Andreas dataset, with a different scene layout and tree species compared to the farm, includes 601 training images and a subset of 192 validation

[1]We thank Peter Fröhlich (AgriCircle) for providing access to the orchard.

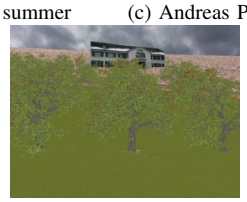
(a) Kastelhof, winter

(b) Kastelhof, summer

(c) Andreas Park



(d) Source, winter



(e) Source, summer

Fig. 3: Sample images from target domain datasets (Kastelhof and Andreas park) and simulated source domain dataset. Source domain images are the results of the copy-paste augmentation of building samples.

images that can be split into four groups of 48 images, each taken around a tree. All images are taken during winter. The generated synthetic source domain dataset includes 5852 samples in total (winter 3401, summer 2451). Furthermore, we use 1155 additional crops of buildings for copy-pasting. These are extracted from general segmentation datasets [40], building datasets [41], [42] and internal datasets. Sample images are shown in Figure 3.

B. Evaluation Methods and Implementation Details

We adopt the commonly used Intersection-over-Union (IoU) and mean IoU (mIoU) metrics for evaluation:

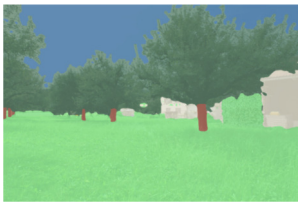
$$\mathrm{IoU} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}, \ \mathrm{mIoU} = \frac{1}{C} \sum_{c=1}^{C} \mathrm{IoU}_c.$$

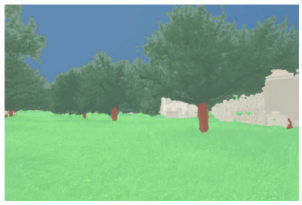
where $\mathrm{TP}, \mathrm{FP}, \mathrm{FN}$ correspondingly stand for the number of true positives, false positives and false negatives.

During domain adaptive training, we select the state-of-the-art DAFormer with ImageNet-pretrained MiT-B1 encoder with training crop size $(640, 640)$ and correspondingly $(1280, 640)$ for inference. The refinements are conducted offline with pretrained SAM ViT-H. We demonstrate results with a MobileNetV3 as the distilled model with a crop size of $(512, 512)$ for training and $(1024, 512)$ for testing. All experiments are conducted on a single Titan X GPU.

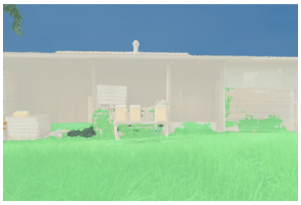
C. Kastelhof Farm

The Kastelhof model is trained on both winter and summer data in the source and target domains. The performance on the Kastelhof validation set is assessed at specific depths in addition to the conventional, full-image mIoU. This is to remove the influence of distant and small objects such as trunks, which are difficult to identify and often blend into terrain or canopy. Selected depths are 15 m and 25 m, where the tree trunks begin to merge with terrain or are entirely undetectable in MobileNetV3's predictions. Our results, including season-specific metrics as well as overall averages, are reported in Table I. The metrics (mean IoU) reported are calculated among all six valid classes: *sky*, *terrain*, *trunk*, *canopy*, *building*, *others*.





(a) The refined labels are still noisy and contain mistakes (left), where a part of the building (white) is recognized as terrain (light green), but the model demonstrates the capability to learn them robustly (right).



(b) The lightweight model (right) corrects the islands representing the false positive "others" class (yellow spots) in the pseudo-label (left). The mean IoU significantly improves from 54.84% to 72.00% at 15 m.

Fig. 4: Visualization of some typical results where the lightweight model outperforms the refined one. In each group, left: refined pseudo-labels, right: MobileNetV3. It is better viewed in color.

We observe that the proposed pseudo-label refinement strategy, on average, improves the vanilla DAFormer predictions by 5.69% and 4.75% at depths 15 m and 25 m, respectively, and by 3.72% if the entire image is considered. This advancement is particularly more pronounced in the summer subset, with significant increases of 10.22%, 8.44%, and 6.83%. The distilled model yields total increments of 6.71%, 5.46%, and 3.60% for 15 m, 25 m, and full-image mIoU compared to vanilla DAFormer predictions. Surprisingly, the distilled model slightly surpasses the refined DAFormer predictions in the validation set. This might be due to the fact that the distilled model is trained with pseudo-supervision on the target dataset only, while the pseudo-labels are obtained by training a DAFormer with both supervision on the source domain and self-supervision on the target domain. Therefore, the distilled model might be capable of learning more domain-specific features and ignoring some of the noise in the refined pseudo-labels, as visible in Figure 4. Typical results at all stages of our training pipeline are illustrated in Figure 5. The final distilled architecture eventually overtakes the results of the vanilla DAFormer, demonstrating the efficacy of the proposed approach.

D. Andreas Park

We extend our experiments to the Andreas Park dataset. In this case, the pipeline only uses the winter subset of the simulated farm environment and the Park training set, producing another MobileNetV3 checkpoint. As the validation set only contains labels for the trees and their surrounding ground area, we evaluate the mean IoU only for trunk, canopy, and terrain inside crops centered around the tree trunks in each image. Considering applications such as digital twins for trees, obtaining the accurate shapes of trunks is essential. Therefore, the trunk IoU is also reported separately. The results of different phases are presented in Table II and

Phase	Pixels with depth < 15 m			Pixels with depth < 25 m			Full Image		
	Overall	Winter	Summer	Overall	Winter	Summer	Overall	Winter	Summer
DAFormer	49.50	50.51	48.14	48.92	49.70	47.87	47.54	48.17	46.70
Refined	55.19	52.81	58.36	53.67	51.70	56.31	51.26	49.57	53.53
MobileNetV3	56.21	53.28	60.14	54.38	51.61	58.09	51.14	48.79	54.29

TABLE I: Mean IoU (mIoU, %) of the DAFormer prediction, refined prediction and MobileNetV3 on Kastelhof validation set. The overall average, winter, and summer performances, considering the full image and different depth thresholds (i.e., 15 m and 25), are all presented. A significant boost in performance occurs after the proposed refinement step, especially for summer data.

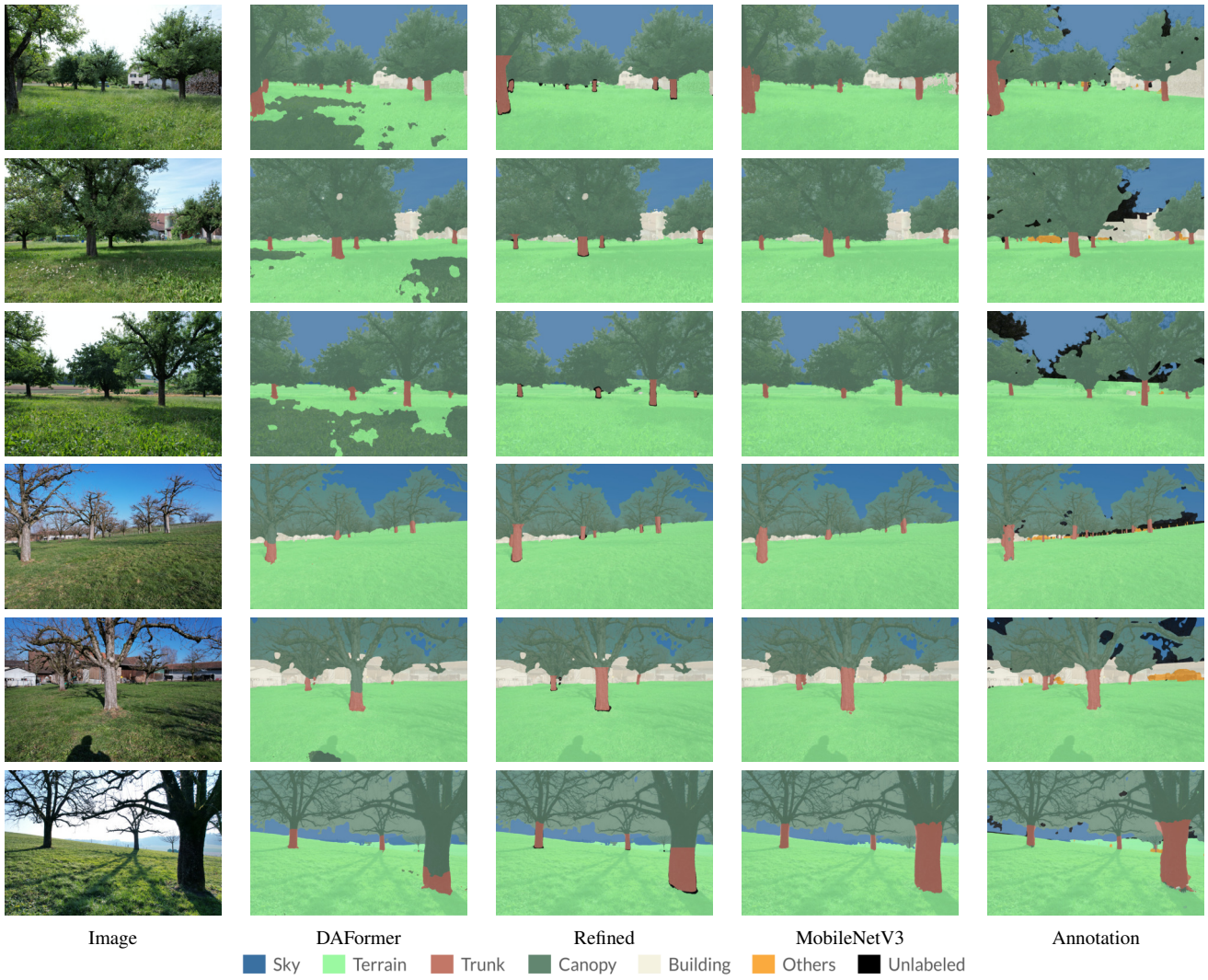


Fig. 5: Outputs as semantic-overlaid images after DAFormer, Refinement, and MobileNetV3 of summer (first three rows) and winter (last three rows) images. Trained with refined pseudo-labels, the lightweight MobileNetV3 overtakes the performance of a DAFormer in the Kastelhof dataset, especially in identifying terrain and trunk pixels.

Figure 6.

In the evaluated crops, the SAM-aided refinement improves the quality of the DAFormer segmentation masks by 12.70% in mIoU and 23.39% in trunk IoU, and the MobileNetV3 effectively learns from the refined pseudo-labels, achieving a performance increase of 11.40% in mIoU and 16.68% in trunk IoU compared to the vanilla DAFormer.

The improvement in mIoU for the Andreas Park dataset is much more significant than that for the Kastelhof Farm dataset. This difference may be due to several reasons. Firstly, the tree species in the simulation are much more similar to those in the Farm than those in the Park, resulting in imperfect predictions for the park trees and, thus, substantial room for pseudo-label improvement. Secondly, the metrics rise easily due to small evaluation regions when significant improvements for the trunk occur. The mean IoU increase can be further amplified by the limited classes involved.

E. Ablation Studies

We analyze the influence of the domain adaptation and the individual refinement steps in our pipeline with the

Step	mIoU	Trunk IoU
DAFormer	65.45	62.60
Refined	78.15	85.99
MobileNetV3	76.85	79.28

TABLE II: Evaluation results (%) after different phases on Andreas Park validation set. Only the region around the ground-truth trunk is evaluated, with classes *trunk*, *canopy* and *terrain*. A significant boost occurred after the refinement, enabling MobileNetV3 to learn effectively due to defective adaptation outputs.

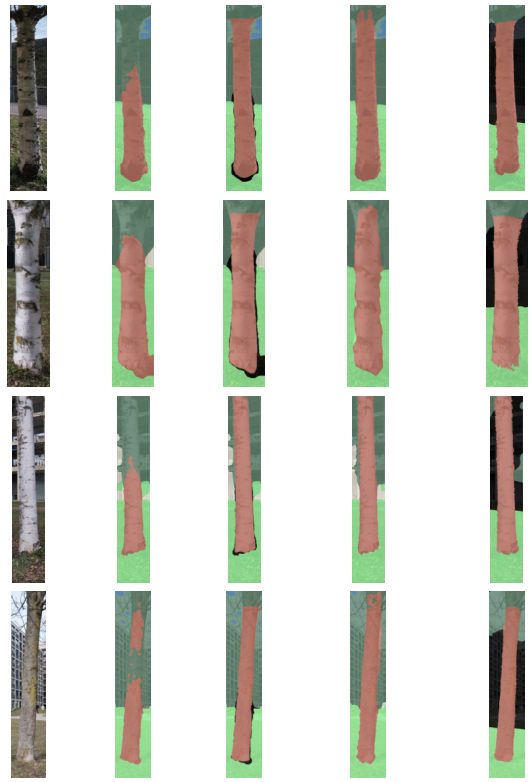


Fig. 6: Images with semantic maps after each step on Andreas test set. The trunk boundaries by MobileNetV3 are much more accurate compared with DAFormer.

Kastelhof dataset. First, we perform domain adaptation on MobileNetV3 directly, using the DAFormer training pipeline. The more powerful DAFormer architecture, trained under the same regime, achieves a performance boost of 15.83% compared to the MobileNetV3 (Table IIIa). This implies that the participation of a model with greater capacity is still necessary to yield acceptable results in the initial domain adaptive training stage of our pipeline.

Next, we analyze the influence of the proposed pseudo-label refinement strategy by training MobileNetV3s with the pseudo-labels obtained after each refinement step. The results are presented in Table IIIb, where each row corresponds to a MobileNetV3 trained under the same settings. The *region correction* contributes most to the overall improvements (2.04% of total 3.11%), as it is capable of fixing errors in sometimes large regions of terrain and buildings. Nonetheless, it can be observed that all refinement steps contribute to the overall performance, which verifies the effectiveness

Architecture	Overall	Winter	Summer
MobileNetV3	32.71	30.70	35.40
DAFormer	48.54(+15.83)	48.17(+17.47)	46.70(+11.30)

(a) Mean IoUs (%) for results of adaptation architectures.

Region Correction	Boundary Correction		Overall	Winter	Summer
	Boundary Sharpening	Label Recovery			
			48.03	47.43	48.83
✓			50.07	47.46	53.58
✓	✓		51.08	48.81	54.11
✓	✓	✓	51.14	48.79	54.29

(b) Mean IoUs (%) for MobileNetV3s trained with corresponding pseudo-labels, showing the effect of refinement tasks.

TABLE III: Effects of an architecture with a higher capacity for adaptation and each refinement task. Mean IoU metrics are calculated for full images.

Architecture	Mean	Standard Deviation
DAFormer	0.1856	0.0033
MobileNetV3	0.0121	0.0159

TABLE IV: Inference timing test with single images on the adaptation and the distilled models. Units in seconds.

of the proposed approach towards improving the quality of the final distilled model.

F. Inference Timing

Finally, inference times of both DAFormer and the distilled MobileNetV3 are measured on the Kastelhof training set, using the Titan X GPU. The results, which are listed in Table IV, showcase that the distilled architecture runs almost 15 times faster than the adapted DAFormer model and with better accuracy. This demonstrates that our approach goes beyond state-of-the-art UDA training pipelines, enabling even lightweight models to achieve reliable performance on the unlabeled target domain.

V. Conclusion

In this paper, we propose a novel training pipeline that bridges the gap between generic domain-adaptive semantic segmentation approaches and their realistic deployment on edge devices, tailoring it to specific applications in natural environments. A powerful adaptation model is initially trained for transferring knowledge from a well-annotated source domain to the target domain, where annotations are unavailable. The quality of the segmentation masks predicted by this model in the target domain is then enhanced offline with the assistance of the Segment Anything Model. Finally, the refined predictions are adopted as pseudo-labels for supervising a distilled, lightweight architecture. Extensive experiments and evaluations on sim-to-real datasets reveal that the lightweight model yielded from our proposed training pipeline consistently surpasses the performance of the larger cross-domain model with a significantly shortened inference

time, thus comprising a crucial step towards enabling semantic segmentation tasks under limited computational resources and without access to ground-truth labels for training.

Future directions involve extending the refinement strategies to more generic scenarios and applying them to few-shot domain adaptation settings, making them even more applicable in the real world.

REFERENCES

[1] L. Bartolomei, P. Teixeira, and M. Chli, "Semantic-aware active perception for uavs using deep reinforcement learning," in *International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[2] S. Hao, Y. Zhou, and Y. Guo, "A brief survey on semantic segmentation with deep learning," *Neurocomputing*, 2020.

[3] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "SegFormer: Simple and efficient design for semantic segmentation with transformers," *Advances in Neural Information Processing Systems*, 2021.

[4] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. March, and V. Lempitsky, "Domain-adversarial training of neural networks," *Journal of machine learning research*, 2016.

[5] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell, "Cycada: Cycle-consistent adversarial domain adaptation," in *International conference on machine learning*, 2018.

[6] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker, "Learning to adapt structured output space for semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[7] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez, "Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, 2019.

[8] Y. Zou, Z. Yu, B. Kumar, and J. Wang, "Unsupervised domain adaptation for semantic segmentation via class-balanced self-training," in *European conference on computer vision (ECCV)*, 2018.

[9] Y. Yang and S. Soatto, "Fda: Fourier domain adaptation for semantic segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, 2020.

[10] W. Tranheden, V. Olsson, J. Pinto, and L. Svensson, "Dacs: Domain adaptation via cross-domain mixed sampling," in *IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021.

[11] L. Hoyer, D. Dai, and L. Van Gool, "DAFormer: Improving Network Architectures and Training Strategies for Domain-Adaptive Semantic Segmentation," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[12] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment Anything," *arXiv:2304.02643*, 2023.

[13] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE transactions on pattern analysis and machine intelligence*, 2017.

[14] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Zhu, and others, "Searching for mobilenetv3," in *International conference on computer vision (ICCV)*, 2019.

[15] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, "Icnet for real-time semantic segmentation on high-resolution images," in *Proceedings of the European conference on computer vision (ECCV)*, 2018.

[16] R. P. Poudel, S. Liwicki, and R. Cipolla, "Fast-scnn: Fast semantic segmentation network," in *The British Machine Vision Conference (BMVC)*, 2019.

[17] C. Yu, C. Gao, J. Wang, G. Yu, C. Shen, and N. Sang, "Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation," *International Journal of Computer Vision*, 2021.

[18] W. Zhang, Z. Huang, G. Luo, T. Chen, X. Wang, W. Liu, G. Yu, and C. Shen, "TopFormer: Token pyramid transformer for mobile semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[19] Q. Wan, Z. Huang, J. Lu, G. Yu, and L. Zhang, "SeaFormer: Squeeze-enhanced Axial Transformer for Mobile Semantic Segmentation," in *International Conference on Learning Representations (ICLR)*, 2023.

[20] F. Pan, I. Shin, F. Rameau, S. Lee, and I. S. Kweon, "Unsupervised intra-domain adaptation for semantic segmentation through self-supervision," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[21] Z. Zheng and Y. Yang, "Adaptive boosting for domain adaptation: Toward robust predictions in scene segmentation," *IEEE Transactions on Image Processing*, 2022.

[22] D. Peng, P. Hu, Q. Ke, and J. Liu, "Diffusion-based Image Translation with Label Guidance for Domain Adaptive Semantic Segmentation," in *International Conference on Computer Vision (ICCV)*, 2023.

[23] F. Magistri, J. Weyler, D. Gogoll, P. Lottes, J. Behley, N. Petrinic, and C. Stachniss, "From one field to another—Unsupervised domain adaptation for semantic segmentation in agricultural robotics," *Computers and Electronics in Agriculture*, 2023.

[24] B. Xie, S. Li, M. Li, C. H. Liu, G. Huang, and G. Wang, "SePiCo: Semantic-Guided Pixel Contrast for Domain Adaptive Semantic Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[25] L. Hoyer, D. Dai, and L. Van Gool, "HRDA: Context-aware high-resolution domain-adaptive semantic segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2022.

[26] L. Hoyer, D. Dai, H. Wang, and L. Van Gool, "MIC: Masked Image Consistency for Context-Enhanced Domain Adaptation," in *Conference on Computer Vision and Pattern Recognition*, 2023.

[27] M. Chen, Z. Zheng, Y. Yang, and T.-S. Chua, "PiPa: Pixel-and Patch-wise Self-supervised Learning for Domain Adaptative Semantic Segmentation," *ACM Multimedia*, 2023.

[28] R. Mascaro, L. Teixeira, and M. Chli, "Domain-Adaptive Semantic Segmentation with Memory-Efficient Cross-Domain Transformers," in *The British Machine Vision Conference (BMVC)*, 2023.

[29] Y. Wang, J. Fei, H. Wang, W. Li, T. Bao, L. Wu, R. Zhao, and Y. Shen, "Balancing Logit Variation for Long-Tailed Semantic Segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[30] T.-D. Truong, N. Le, B. Raj, J. Cothren, and K. Luu, "Fredom: Fairness domain adaptation approach to semantic scene understanding," in *Conference on Computer Vision and Pattern Recognition*, 2023.

[31] Y. Du, Y. Shen, H. Wang, J. Fei, W. Li, L. Wu, R. Zhao, Z. Fu, and Q. Liu, "Learning from Future: A Novel Self-Training Framework for Semantic Segmentation," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[32] X. Zhao, N. C. Mithun, A. Rajvanshi, H.-P. Chiu, and S. Samarasekera, "Unsupervised Domain Adaptation for Semantic Segmentation with Pseudo Label Self-Refinement," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024.

[33] M. A. Mazurowski, H. Dong, H. Gu, J. Yang, N. Konz, and Y. Zhang, "Segment anything model for medical image analysis: an experimental study," *Medical Image Analysis*, 2023.

[34] X. Yang and X. Gong, "Foundation Model Assisted Weakly Supervised Semantic Segmentation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2024.

[35] X. Zou, J. Yang, H. Zhang, F. Li, L. Li, J. Gao, and Y. J. Lee, "Segment everything everywhere all at once," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[36] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision (ECCV)*, 2014.

[37] W. Yan, Y. Qian, H. Zhuang, C. Wang, and M. Yang, "SAM4UDASS: When SAM Meets Unsupervised Domain Adaptive Semantic Segmentation in Intelligent Vehicles," *IEEE Transactions on Intelligent Vehicles*, 2024.

[38] S. Rumley, A. Thoma, P. Beardsley, L. Teixeira, and M. Chli, "From perspective view to bird's eye view in agricultural environments," in *International Conference on Robotics and Automation Workshops(ICRAW)*, 2023.

[39] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E. D. Cubuk, Q. V. Le, and B. Zoph, "Simple copy-paste is a strong data augmentation method for instance segmentation," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[40] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[41] Radim Tyleček and Radim Šára, "Spatial Pattern Templates for Recognition of Objects with Regular Structure," in *Proc. GCPR*, 2013.

[42] F. Korč and W. Förstner, "eTRIMS Image Database for Interpreting Images of Man-Made Scenes," Dept. of Photogrammetry, University of Bonn, Tech. Rep., 2009.