

This work was written as part of one of the author's official duties as an Employee of the United States Government and is therefore a work of the United States Government. In accordance with 17 U.S.C. 105, no copyright protection is available for such works under U.S. Law.

Public Domain Mark 1.0

<https://creativecommons.org/publicdomain/mark/1.0/>

Access to this work was provided by the University of Maryland, Baltimore County (UMBC) ScholarWorks@UMBC digital repository on the Maryland Shared Open Access (MD-SOAR) platform.

Please provide feedback

Please support the ScholarWorks@UMBC repository by emailing scholarworks-group@umbc.edu and telling us what having access to this work means to you and why it's important to you. Thank you.

An Online Continuous Semantic Segmentation Framework With Minimal Labeling Efforts

Masud Ahmed*, Zahid Hasan*, Tim Yingling*, Eric O'Leary†, Sanjay Purushotham*, Suya You‡, Nirmalya Roy*

*University of Maryland Baltimore County, USA

†University of Maryland, USA

‡DEVCOM Army Research Laboratory, USA

*{mahmed10, zhasan3, ww74617, psanjay, nroy}@umbc.edu, †eoleary@umd.edu, ‡suya.you.civ@army.mil

Abstract—The annotation load for a new dataset has been greatly decreased using domain adaptation based semantic segmentation, which iteratively constructs pseudo labels on unlabeled target data and retrains the network. However, realistic segmentation datasets are often imbalanced, with pseudo-labels tending to favor certain “head” classes while neglecting other “tail” classes. This can lead to an inaccurate and noisy mask. To address this issue, we propose a novel hard sample mining strategy for an active domain adaptation based semantic segmentation network, with the aim of automatically selecting a small subset of labeled target data to fine-tune the network. By calculating class-wise entropy, we are able to rank the difficulty level of different samples. We use a fusion of focal loss and regional mutual information loss instead of cross-entropy loss for the domain adaptation based semantic segmentation network. Our entire framework has been implemented in real-time using the Robotics Operating System (ROS) with a server PC and a small Unmanned Ground Vehicle (UGV) known as the ROSbot2.0 Pro. This implementation allows ROSbot2.0 Pro to access any type of data at any time, enabling it to perform a variety of tasks with ease. Our approach has been thoroughly evaluated through a series of extensive experiments, which demonstrate its superior performance compared to existing state-of-the-art methods. Remarkably, by using just 20% of hard samples for fine-tuning, our network has achieved a level of performance that is comparable ($\approx 88\%$) to that of a fully supervised approach, with mIOU scores of 60.51% in the In-house dataset.

Index Terms—Semantic Segmentation; Domain Adaptation; Active Learning; Continual Learning; Robotics Operating System.

I. INTRODUCTION

Semantic segmentation has a large number of applications in various computer vision applications e.g. autonomous driving [1], medical diagnosis [2], robot manipulation [3], etc. Recently, convolutional neural network based deep learning architectures have achieved state-of-the-art results in semantic segmentation based tasks [4], [5]. However, the deep learning network development phases require massive manually annotated pixel labels or masks. Further, these network depends on the data domain and often fails to generalize beyond the trained data domain due to distribution shifts in the target domain.

Domain adaptation (DA) techniques [6]–[8] provide a feasible solution against domain shift by enforcing learning domain invariant features from different data domains and transferring the knowledge to the new target domain. Unsupervised domain adaption (UDA) further enables such transferring by aiming to develop domain independent feature representations and reduce the data distribution disparity between the source and target domains using the unlabeled target domain data. This facilitates reliable downstream tasks of semantic segmentation with minimal annotation. However,

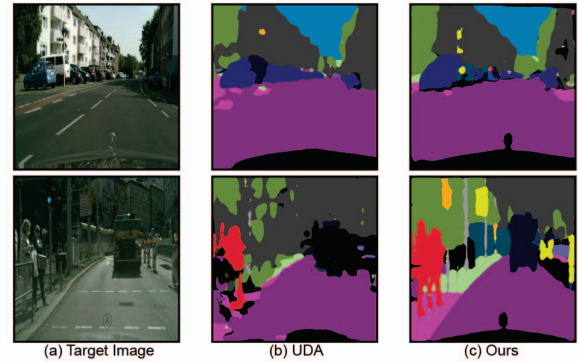


Fig. 1: **Proposed framework result.** From the left to right column: (a) target image, (b) output generated by UDA, (c) output generated by our approach. In the first row, the sample is taken from a sample that contains no tail classes; whereas the sample in the second row contains few tail classes.

these approaches suffer performance degradation in knowledge transfer of tail classes, as shown in Fig 1. Active domain adaptation (ADA) provides an alternative solution, which requires a small amount of labeled target domain data.

In this research work, we address performance issues in tail classes and investigate the optimal quantity and quality of required target-labeled data for successful downstream task performances in ADA semantic segmentation. We identify the pixel-wise class consideration limitation of state-of-the-art DA for semantic segmentation as across the domain the pixel properties vary. Moreover, we suspect that traditional binary cross entropy (*BCE*) loss of semantic segmentation also contributes to performance degradation of the tails classes. We hypothesize that region based semantic loss, such as **Region Mutual Information (RMI)** loss [9] with constrained on tail classes will enable learning domain invariant features as across the domain the neighborhood pixels correlate with each other for image data and improve the tail class generalization. Further, we hypothesize that it would prevail better hard samples for the network and reduce the data labeling complexity in fine-tuning stage using a small number of labeled target data. Particularly, we investigate if the region based losses mine better hard samples for efficient ADA semantic segmentation learning.

We also implement the entire framework in real-time using the Robotics Operating System (ROS) on a server PC, along with a small Unmanned Ground Vehicle (UGV) called the ROSbot2.0 Pro. ROS is a widely used middleware for robotics

that provides software libraries and tools for developing robot applications¹. It offers a standardized way for different components of a robotic system to communicate with each other, making it easier to develop complex robot systems. Our implementation with ROS allows the ROSbot2.0 Pro to access any type of data at any time, enabling it to perform a variety of tasks with ease. Additionally, all components of ROS adhere to a common structure, making it easy for users to access and utilize other community-developed software programs.

In a nutshell, we pose our contributions as follows:

- We combine region based mutual information loss and focal loss with DA for semantic segmentation to address the class imbalance problem and perform exhaustive experiments with different dataset combinations. We also provide insight regarding optimal performances.
- Besides publicly available datasets, we test the approach on our collected In-house dataset collected by a small unmanned ground vehicle from a lower point of view from a suburban environment. Our In-house dataset is publicly available².
- Finally, we design a framework using ROS noetic on a server PC and a small UGV ROSbot2.0 Pro. In the framework, the server PC is the master and ROSbot2.0 Pro is the client. ROSbot2.0 Pro collects the data using the camera node, and the server PC performs semantic segmentation using the segmentation node. Besides performing semantic segmentation, it also stores the hard samples that can be used to retrain the network in a later stage.

II. RELATED WORKS

UDA. In recent days, UDA has become a popular research topic in the field of computer vision for image classification [10], [11], object detection [12], [13], and semantic segmentation [14], [15]. A practical application scenario is to perform semantic segmentation on unlabeled real-time target domain images, using labeled synthetic source domain images [16], [17]. Initial research in the field of DA uses adversarial learning or generative approaches [8], [18] to reduce the difference between the source and target domain by minimizing the domain feature discrepancy. In the field of semantic segmentation application, researchers nowadays apply DA in three ways: alignment of the output space [16], [19], [20], feature matching [21], [22], and appearance transfer [23], [24]. Lately, self-training has become a popular alternative research subject to UDA [25]–[29]. In the self-training network, besides using source images and labels, the semantic segmentation network is fine-tuned by the target domain, feeding its own segmented mask as pseudo-labels. One method is to creatively design class-confidence criteria to disguise incorrect predictions [6], [30]. Other research suggests using pseudo-labels with various regularization strategies to reduce both the inter-domain and intra-domain gap [31], [32]. Inspired by these works, we propose the ranking framework in this work.

ADA. Though semi-supervised domain adaptation (SSDA) and ADA operate almost in a similar way there is a slight difference between these two approaches. SSDA takes random label target samples as input, whereas ADA concentrates on

labeling target instances for UDA based on certain criteria [33]. Existing ADA works mostly concentrate on classification [34]–[36]. There are several approaches to ADA, for example, including uncertainty and variety in an acquisition round and SSDA into a unified framework [35], selecting target samples in an SSDA setup based on predicted entropy and targetness is determined by an adversarial discriminator [37]. Recently, researchers begin to concentrate on the task of semantic segmentation ADA, which considerably improves semantic segmentation accuracy on the target domain samples [38], [39]. Approaches adopted by the researchers for semantic segmentation ADA are selecting a subset of images using multi-anchor strategy [38], point based labeling with a pixel selector that can be adaptive [39], region based selection strategy via region impurity and prediction uncertainty [40], label acquisition strategy by clustering uncertainty-weighted embeddings [35].

III. METHOD

A. Overview

Notation: We denote the labeled source dataset by $S = \{(x_i, y_i)\}_{i=1}^{N_s}$ where $x_i \in S$ are images from source data distribution S with pixel-wise label y_i corresponds to one of the C classes. We represent target dataset $T = \{x_i\}_{i=1}^{N_t}$ where images $x_i \in T$ are from distribution T and $S \neq T$. We utilize semantic segmentation network, $\phi_e: S/T \rightarrow \mathbb{R}^{|C|}$ to project images to label space. We denote discriminator network, $\phi_d: \mathbb{R}^{|C|} \rightarrow \mathbb{R}^2$, to apply adversarial losses on feature embedding.

The UDA transfers the source domain S knowledge to the target domain T by learning domain invariant features while performing comprehensively on both S and T . We observe the semantic segmentation network ϕ_e performs well on T for a few classes C from label space y . However, they under-perform for tail classes C_t due to limited instances N_{C_t} from S . These very few instances have a different distribution than the same class instance from the target domain T . Therefore, ϕ_e can perform well in the source domain only for the tail distribution classes. There are several approaches to solve this problem. In this work, we fine-tune the ϕ_e with labels y_t of T . The fine-tuning stage utilizes small T data to achieve a relatively good result like supervised training. Here, we mine the 20% hard samples x_t from the target domain and provide active label y_t by calculating the entropy E from the predicted label \hat{y}_t for those samples to fine-tune the network. We fine-tune the network using the continual learning approach. We hypothesize that if the segmentation network predicts the wrong class for any pixel, then the entropy value will be high for that pixel. For this purpose, we modified the loss function in the semantic segmentation network. The BCE \mathcal{L}_{ce} is usually used to train the semantic segmentation network ϕ_e , which we replace by the RMI loss \mathcal{L}_{rmi} . We experimented with the mutual information loss to prevail better learning capacity in the UDA approach, by combining with the focal loss \mathcal{L}_f . The overall pipeline of our work is shown in Fig 2.

B. Entropy Ranking

Due to varying weather conditions, motion, and shadow, target domain images gathered from other domains may have a diverse distribution. Therefore, the semantic segmentation network that is trained on the source domain, performs poorly for a few classes, especially for the tail classes. One simple

¹<https://www.ros.org/>

²<https://ieee-dataport.org/documents/cad-edgetune>

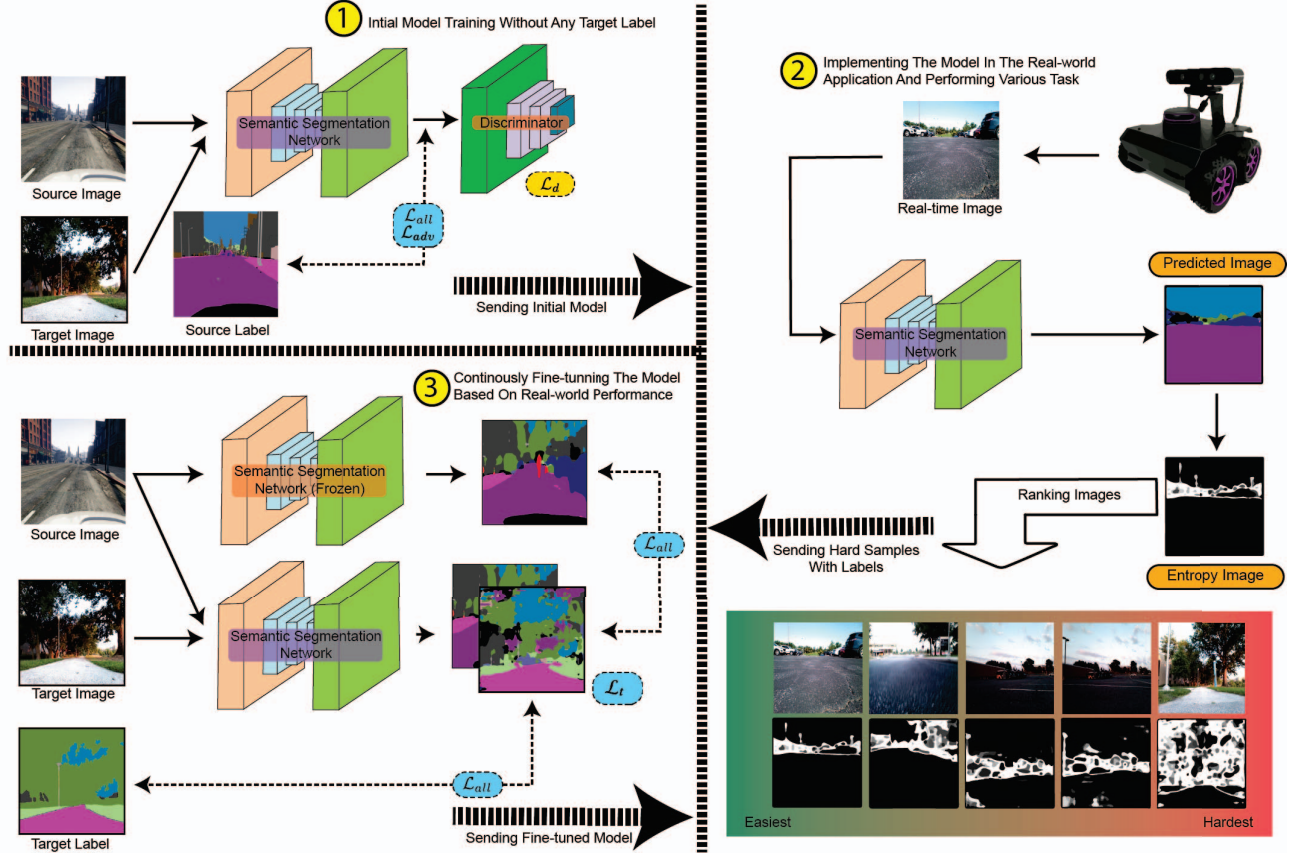


Fig. 2: **The overview of our proposed framework.** First, we train the network in a UDA setting using labeled source data, and unlabeled target data. Then we rank the prediction on target data using entropy. We split a small portion of hard samples from target data and provide labels. Finally, we fine-tune the network using only the labeled hard samples in a continual learning approach.

solution to this problem is to fine-tune the network with the target domain image and mask. But this will spoil the original motive of the domain adaption, which is to lessen the annotation burden. To tackle this, we can fine-tune the whole network which is a very small portion of target data and mask, for which the network generates noise and incorrect predictions. Fine-tuning with that small chunk of data can solve the problem of the tail class issue and perform a similar supervised setup. But one of the major issues is target domain image does not have label data. Therefore, we can not calculate the mean IOU of an image and declare that as a noisy sample. An alternative solution is ranking images based on entropy [31]. The noisy prediction has a higher entropy value whereas a clean prediction has a lower entropy value. So, after predicting the mask from the semantic segmentation network, we generate the entropy image I_t from the prediction mask \hat{y}_t . After that, we calculate the mean entropy value per class wise, defined by Eq 1.

$$E_t = \frac{1}{HW} \sum_{h,w} \sum_c I_t \quad (1)$$

C. Overall Loss

The overall loss of our network is intended to address the problem of preventing a network from predicting the wrong class with low entropy. To deal with the problem we begin

with binary cross entropy loss \mathcal{L}_{ce} . Cross entropy is calculated from the difference between the probability distribution of predicted \hat{y} and actual label y , defined by Eq 2 [41].

$$\mathcal{L}_{ce}(y, \hat{y}) = -(y \log(\hat{y}) + (1-y) \log(1-\hat{y})) \quad (2)$$

One of the major drawbacks of this loss for DA based semantic segmentation is it predicts wrong tail classes in the target domain with low entropy. To overcome this weighted cross entropy loss \mathcal{L}_{wce} can be applied, defined by Eq 3 [42].

$$\mathcal{L}_{wce}(y, \hat{y}) = -(\beta * y \log(\hat{y}) + (1-y) \log(1-\hat{y})) \quad (3)$$

Here, β is the weight matrix. But these weight values are predefined, which is unable to solve the class imbalance issue. Therefore, we adapt *focal loss* [43] instead of cross-entropy loss \mathcal{L}_f , defined by Eq 4.

$$\mathcal{L}_f = -(1-p_t)^\gamma \log(p_t) \quad (4)$$

Here γ is hyperparameter, $\gamma \in [0, 5]$. Also, p_t is defined by Eq 5.

$$p_t = \begin{cases} p & \text{if } y=1 \\ 1-p & \text{otherwise} \end{cases} \quad (5)$$

Though this solves the class imbalance problem, the loss that occurs due to regional information still exists. To solve this issue in the field of DA based semantic segmentation, we incorporate the region mutual information (*RMI*) loss [9]. In contrast to *BCE* or focal loss, which considers pixels as individual samples, *RMI* defines a pixel by incorporating its

neighbor pixels with that pixel. The image is then converted into a multidimensional distribution of these high-dimensional points for each pixel in the image. Thus, by optimizing the mutual information (MI) between their multidimensional distributions, prediction, and ground truth can attain high-order consistency. The mutual information I_{mi} in quadrangular region size $\mathcal{R} \times \mathcal{R}$ is defined by Eq 6.

$$I_{mi} = \int_Y \int_{\hat{Y}} f(y, \hat{y}) \log \frac{f(y, \hat{y})}{f(y)f(\hat{y})} dy d\hat{y} \quad (6)$$

By incorporating the mutual information proposed *RMI* loss is defined by Eq 7.

$$\mathcal{L}_{rmi} = \frac{1}{B} \sum_B \sum_C (-I_{mi}) \quad (7)$$

Here, B , and C denote the mini-batch and all the classes respectively. The author combined *BCE* [9] loss with *RMI* loss to calculate the overall loss. Here, we update the overall loss by incorporating focal loss instead of *BCE* loss as in Eq 8.

$$\mathcal{L}_{all} = \lambda \mathcal{L}_f + (1 - \lambda) \mathcal{L}_{rmi} \quad (8)$$

Here, λ is the loss weight factor, where $\lambda \in [0, 1]$. This focal mutual information loss addresses the issues of both the class imbalance issue and neighboring pixel dependency.

D. Adversarial Learning Objective

The adversarial learning aims to enhance robustness by learning domain invariant features for the semantic segmentation network [20]. A discriminator is designed to predict whether the mask generated from the segmentation network is coming from the source domain or the target domain. Given the output of semantic segmentation network \hat{y} , we feed this \hat{y} to ϕ_d , which then predicts the source or target label for that \hat{y} . The loss calculates for the discriminator is a cross-entropy loss, defined by Eq 9.

$$\mathcal{L}_d = -\sum (1 - z) \log(\phi_d(\hat{y})^{(0)}) + z \log(\phi_d(\hat{y})^{(1)}) \quad (9)$$

Here z denotes the domain identifier for the image, $z=0$ for the source, and $z=1$ for the target.

To adversarial train the ϕ_e , we incorporate adversarial loss \mathcal{L}_{adv} as defined in Eq 10.

$$\mathcal{L}_{adv} = -\sum \log(\phi_d(\hat{y}_t)^{(1)}) \quad (10)$$

This loss intends to train the segmentation network and deceive the discriminator by increasing the likelihood that the target prediction will be accepted as the source prediction. Therefore there is a min-max criterion exists between the discriminator and semantic segmentation network, which can be represented by Eq 11.

$$\max_D \min_{SS} \mathcal{L}(x_s, x_t) \quad (11)$$

The ultimate target is to minimize semantic segmentation loss.

E. Continual Learning Objective

In the domain-incremental scenario, the data distribution $P(x)$ varies with each task increment, but the class distribution $P(y)$ remains constant, $P(x_s) \neq P(x_t)$ and $P(y_s) = P(y_t)$. In other words, we see the same classes but they look different. A domain increment of this type may be transferred from synthetically created data to camera-recorded data, from one terrain to another, or from one-time instance to another for as long as the set of labeled classes remains constant. Consequently, continually fine-tuning on newer, target domain data (e.g., images from the In-house dataset in our case) may

result in the network forgetting previously learned knowledge from the source domain (commonly referred to as catastrophic forgetting [44]). To solve this issue, we use a replay based continual learning method similar to recent works [45].

During the fine-tuning stage, we use labeled In-house data and a fraction of previous domain data. First, we create a duplicate network ϕ'_e by cloning the ϕ_e . We keep ϕ'_e frozen. Then we fine-tune the ϕ_e by calculating loss from the following Eq 12.

$$\mathcal{L}_t = \mathcal{L}_{all}[y_t, \phi_e(x_t)] + \mathcal{L}_{all}[\phi'_e(x_s), \phi_e(x_s)] \quad (12)$$

IV. EXPERIMENT

A. Experimental Setting

We experiment with two publicly available state-of-the-art datasets (GTAV [46], Cityscapes [47]) as well as our gathered In-house dataset to validate our approach. We use GTAV as our source domain data, which contains RGB and annotated ground truth mask. GTAV is a synthetic dataset consisting of around 25K annotated data, generated by a realistic open-world video game. The image resolution of the images is 1914×1052 . As a target domain data, we use our In-house dataset, which is collected in a semi-urban area. For this setup, we train the network only for 14 common classes, the rest of the classes are classified as a void classes. We implement the real-time experiment using this setup. Besides, we also evaluate our framework on another state-of-the-art dataset Cityscapes, where we use GTAV as the source domain dataset. Cityscapes is a real-world dataset collected in urban areas. For evaluation, we use similar setups as we have in earlier research studies [6], [18], [20], [40]. Details of our In-house dataset are given below.

B. In-house Dataset

The In-house dataset is collected from a sub-urban campus area using a Husarion ROSbot2.0 and ROSbot2.0 Pro with the collection speed set to 5 frames per second. The dataset can be divided into noon, dusk, and dawn subsets as we want to capture our environment in distinct lighting conditions. We collect 17 sequences with a total number of 8080 frames, 1619 of which are annotated by hand using an open-source pixel annotation tool. We decide on annotating every 5 of the images as adjacent images were very similar to each other and we can feed the network duplicate images if needed. When annotating the In-house dataset, we use soft labeling which allows us to move faster through the frames since the annotation process can be very time-consuming. The annotation tool that we use allows us to make small markings within an object in an image and the classification would extend to similar pixels. As a result of using this method, annotations may not be perfectly accurate which sacrifices some performance accuracy but the drop-off would not be significant enough to warrant creating pixel-perfect annotations. Fig 3 shows an example of some of the images taken around noon alongside the masks that are drawn for them. Since all of the images are using the RosBots, they are from a lower perspective than most other datasets, which introduces another form of variation to our target domain.

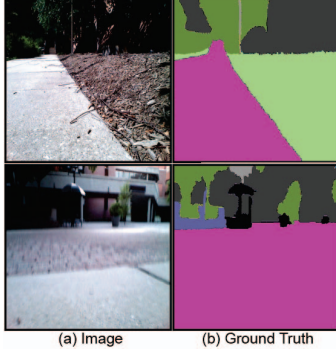


Fig. 3: **Example of the In-house dataset.** From left to right column: (a) image, (b) ground truth. These two samples of the dataset are collected during noon and dusk time.

C. Deep-learning Network Details

To design the network, we choose the PyTorch deep learning framework and experimented with NVIDIA RTX 3090 GPU. We adopt DeepLab-v3+ architecture [48] with ResNet101 [49] pre-trained on Model Zoo as ϕ_e . In terms of training, we apply the SGD optimizer with 0.9 momentum, 5×10^{-4} weight decay, the initial learning rate of 1×10^{-3} , and polynomial learning rate decay. We use the batch size of 2 for each iteration and trained the whole network till the loss curve is saturated. Typically, we achieve the validation loss saturation in about 100 epochs. We select $\gamma=1$, $\mathcal{R}=7$ and $\lambda=0.75$ as our loss function parameters. As the discriminator network, we use the similar network proposed in DCGAN [50].

D. Real-time Device Implementation Details

ROS is an open-source and modular software platform. We use two devices server PC as master and ROSbot2.0 pro as client. Ubuntu 20.04 operating system is installed in both of the devices, and on top of that, we install the ROS noetic on both of the devices. ROS includes numerous software components encapsulated as nodes. Fig 4 shows the structure of the ROS with library layer, package layer, and node layer. In ROS, the master needs to start the */roscore* node first, and all the clients can get connected with the framework. ROS allows distributed network data communication. The master node manages and oversees the operation of the functional nodes as well as their peer-to-peer communications [51]. Though we use WiFi for communication, there are several other communication approaches that we can adopt in the future, e.g., optoacoustic communication [52].

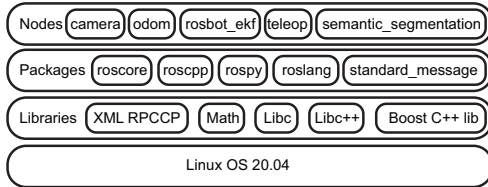


Fig. 4: **ROS architecture blocks.** Important nodes, packages, libraries of our experiment are mentioned here.

In the ROS system, we create several nodes that can be subscribed to by both devices anytime. Among them, the two most important nodes are */camera/rgb/image_raw/image_topics* and */semantic_segmentation_topics*. The ROSbot2.0 Pro

navigates in the real-world wild areas where it collects data using an RGB camera and publishes it to the */image_topics*. Subscribing this topic the server PC fetches the RGB image, performs semantic segmentation, and publishes it to */semantic_segmentation_topics*. During this semantic segmentation, it also keeps track of hard images and stores them in the hard disk, so that we can use it to fine-tune the network in the later stage. As the whole framework is designed in ROS, ROSbot2.0 Pro can subscribe */semantic_segmentation_topics* anytime, and perform any task based on that. There are other topics that can be used to navigate and provide commands to the ROSbot2.0 Pro. The ROS computation graph is shown in Fig 5.

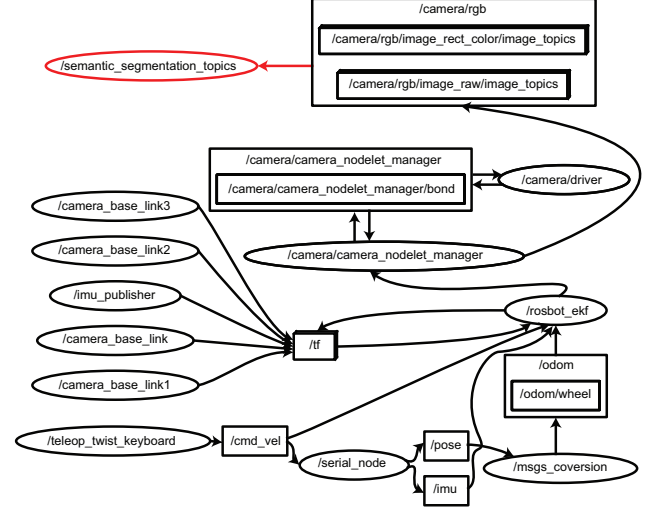


Fig. 5: **ROS computation graph.** In this graph our major contribution is implemented in */semantic_segmentation_topics* node.

V. RESULTS

A. Result Reporting Details

We use the following format to report our experiment results.

- **Supervised:** This network is trained with 100% labeled (1619 instances) In-house data. As this is the baseline network, no approach should beat this network's performance.
- **UDA:** This is UDA settings, where we train the network with labeled GTAV dataset and unlabeled In-house dataset. Results of both BCE and F-RMI loss based networks are reported.
- **20%e-split:** After initial UDA training the network is fine-tuned with 20% easy samples from the In-house dataset. The result is reported for the F-RMI loss based network only, to show that easy sample based fine-tuning will not perform better.
- **20%h-split:** Instead of fine-tuning with easy samples, we fine-tune the network with hard samples. Our approach is shown in bold form.

B. Quantitative Result

We present the detailed result of our approach in GTAV → In-house dataset setting including all other approaches too in Tab I. As the domain knowledge gap between the In-house dataset and GTAV is relatively large, we can see that the performance of UDA based approaches is relatively poor. Also, we can notice two huge performance boosts firstly when we

TABLE I: Results in GTAV \rightarrow In-house Dataset setting. Our approach is reported in **bold** form

Models	Road	Sidewalk	Building	Wall	Fence	Pole	TSignal	Viation	Terrain	Sky	Person	Car	Truck	Bus	mIOU
Supervised	95.76	96.88	78.96	37.18	74.11	28.33	8.28	88.04	79.07	92.26	51.29	81.74	71.24	68.65	67.99
UDA(<i>BCE</i>)	15.10	14.19	28.45	2.29	7.52	8.40	3.08	57.81	25.87	67.47	4.85	40.01	0.35	6.19	20.11
UDA(<i>F-RMI</i>)	15.18	11.34	30.51	1.04	0.64	6.41	3.45	61.96	28.27	74.47	5.37	31.90	0.38	4.18	19.65
20%e-split(<i>F-RMI</i>)	48.29	75.18	41.18	0.22	5.23	8.82	6.64	65.81	25.22	83.82	11.99	42.97	29.28	0.00	31.76
20%h-split(<i>BCE</i>)	77.50	87.80	55.61	9.32	32.02	15.31	6.04	80.01	67.50	88.45	13.28	60.79	1.02	58.33	46.64
20%h-split(<i>F-RMI</i>)	83.86	90.55	71.02	36.90	53.08	24.66	7.90	87.89	74.03	90.88	49.48	80.88	35.38	60.63	60.51

TABLE II: Results in GTAV \rightarrow Cityscapes setting. Our approach is reported in **bold** form

Models	Road	Sidewalk	Building	Wall	Fence	Pole	TLight	TSignal	Viation	Terrain	Sky	Person	Car	Truck	Bus	Train	Meycle	Bicycle	mIOU
Supervised	95.72	78.48	85.50	47.11	51.78	22.02	29.88	37.01	85.88	60.43	85.54	72.62	89.29	44.47	71.15	66.47	47.52	56.41	62.62
UDA(<i>BCE</i>)	71.36	23.73	53.63	15.19	13.62	8.07	4.42	0.62	65.48	23.88	58.97	35.78	63.52	1.47	4.68	0.00	12.77	0.00	25.40
UDA(<i>F-RMI</i>)	53.57	12.48	49.13	8.66	6.54	8.50	2.40	2.09	62.08	24.60	55.24	24.10	42.24	4.24	7.16	0.01	10.40	0.17	20.76
20%e-split(<i>F-RMI</i>)	75.69	30.14	51.53	8.09	4.08	5.17	1.29	8.27	68.80	21.55	72.19	17.48	55.05	3.50	5.04	0.05	2.37	3.47	24.10
20%h-split(<i>BCE</i>)	86.27	55.98	69.23	25.02	37.08	20.56	23.65	30.77	81.25	49.65	77.81	65.06	78.20	34.31	13.15	19.88	28.91	53.31	47.23
20%h-split(<i>F-RMI</i>)	91.42	73.42	76.69	40.43	50.77	21.15	24.36	33.80	81.08	60.39	81.51	61.17	81.01	41.63	64.05	58.20	41.41	43.70	57.02

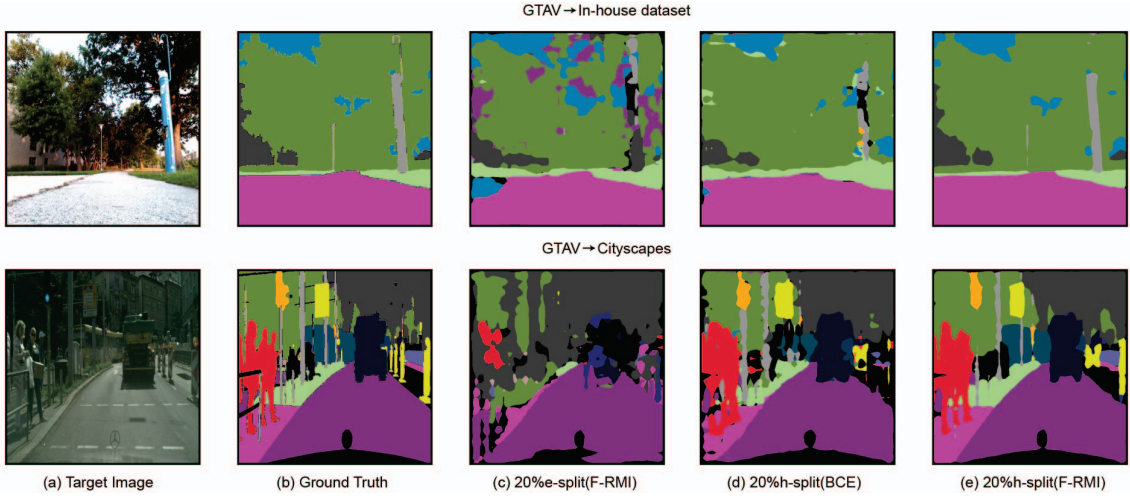


Fig. 6: Predicted results visualization of our approach in three different settings. From left to right column: (a) target image, (b) ground truth, (c) output of the network fine-tuned by 20% easy split samples, (d) output of the network fine-tuned by 20% hard split samples and *BCE* based loss, (e) output of the network fine-tuned by 20% hard split samples and *F-RMI* based loss.

use hard splits instead of easy splits, and secondly when we use *F-RMI* loss instead of *BCE* loss. For all the classes we see that *F-RMI* loss and hard splits based fine-tuned network outperforms all the other networks. We also evaluate our approach in GTAV \rightarrow Cityscapes setting shown in Tab II. As there are 18 common classes between GTAV and Cityscapes dataset, we report IOU for 18 classes. The performance in this setting is similar to GTAV \rightarrow In-house dataset setting.

C. Qualitative Result

In Fig 6, we illustrate the segmentation performance generated by our approach and compare them to other approaches. According to the result, we can see that 20% easy split fine-tuned network performs relatively poor than the other two networks as this network is fine-tuned with relatively easier samples. Also, we observe that in *BCE* based fine-tuned network predict some false positive cluster inside a true positive cluster. As *F-RMI* loss considers neighbor pixels during loss calculation, it solves this issue.

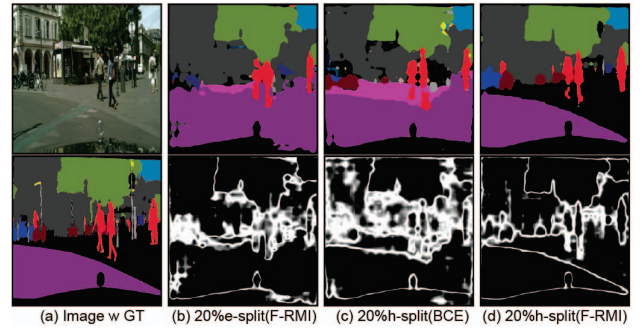


Fig. 7: Entropy image visualization. From left to right column: (a) image with ground truth, (b)-(d) output and corresponding entropy image of the network fine-tuned by 20% easy split samples, the network fine-tuned by 20% hard split samples and *BCE* based loss, the network fine-tuned by 20% hard split samples and *F-RMI* based loss respectively. The entropy value is represented by the brightness of the entropy image.

Also, in Fig 7, we show the entropy image of the same sample for three different fine-tuned networks. We can see that F-*RMI* loss based both easy-split and hard split based fine-tuned networks usually predict a true positive with low entropy and a false positive with high entropy. On the other hand, the *BCE* loss based hard split based predicts a true positive with relatively high entropy. For the reliability issue, we prefer F-*RMI* loss rather than any other loss.

D. Real-Time Implementation Result

We run the real-time experiment near our testing lab. We design a python-opencv API based GUI to visualize the result. The framework takes ≈ 300 ms to segment each image. The real-time result of the segmentation is shown in Fig 8.



Fig. 8: **Example of real-time implementation.** From the left to right column: (a) real-time scenario during robot navigation, (b) output of the GUI.

E. Discussion

Our approach performs well if the ranking system ranks the target image properly. The ranking system is highly dependent on entropy. To control the entropy, we have three important hyperparameters that we need to fine-tune based on our needs.

- **Adjustable parameter γ for focal loss:** This is one of the important hyperparameters to set the focal loss. Based on the γ value this loss will act like a weighted loss for the tail object. Based on the class imbalance ratio, we can increase or decrease the value of the γ . According to the authors who proposed this focal loss [43], $\gamma \in [0, 5]$. If we set the value of $\gamma=0$, it will become a *BCE* loss. In this work, we set the value of $\gamma=1$.
- **Square region size \mathcal{R} for *RMI* loss:** Another important hyperparameter to set the *RMI* loss is \mathcal{R} . Based on the \mathcal{R} value, the *RMI* loss will consider more neighbor pixels during loss calculation. According to the authors who proposed *RMI* loss [9], $\mathcal{R} \in [1, 10]$. If we set the value of the \mathcal{R} higher, then the network performs better. But increasing the value of \mathcal{R} means additional computational cost. Therefore, we need to do a tradeoff. In this work, we set the value of $\mathcal{R}=7$.
- **Loss weight factor λ for overall loss:** In the end, we need to find a suitable value of loss weight factor λ to set the

overall loss. This is the most important factor, as we need to decide the ratio of focal loss and *RMI* loss by this factor. According to our proposal, $\lambda \in [0, 1]$. If the data distribution gap between the two domains is high, we set the value low. In this work, we set the value of $\lambda=0.75$.

VI. CONCLUSION

In this research work, we look beyond traditional cross-entropy based losses for semantic segmentation in the ADA setting and propose to incorporate region based regularization to increase accuracy and consistency across the domains. Further, we propose to utilize focal loss alongside region based loss to tackle the tail distribution of the classes for both the source and target domain. For better performance on the target domain, we fine-tune the network with hard target samples. With only 20% of active annotation on the target domain, our approaches performed comparatively with completely annotated supervised settings in the target domain. We implement our framework with ROS in a real-time environment. As we do not have access to any GPU based UGV, most of the computationally heavy tasks are performed on the server PC. In the future, we plan to perform most of the computational tasks in the UGV, so that we can use the server PC to fine-tune the network simultaneously.

ACKNOWLEDGEMENTS

This work has been partially supported by ONR Grant #N00014-23-1-2119, U.S. Army Grant #W911NF2120076, NSF CAREER Award #1750936, NSF REU Site Grant #2050999, NSF CNS EAGER Grant #2233879.

REFERENCES

- [1] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "Denseaspp for semantic segmentation in street scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3684–3692, 2018.
- [2] A. Shahsavari, T. Khatibi, and S. Ranjbari, "Skin lesion detection using an ensemble of deep models: Slided," *Multimedia Tools and Applications*, pp. 1–20, 2022.
- [3] R. Sheikh, A. Milioto, P. Lottes, C. Stachniss, M. Bennewitz, and T. Schultz, "Gradient and log-based active learning for semantic segmentation of crop and weed for agricultural robots," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1350–1356, IEEE, 2020.
- [4] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.
- [5] 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*, vol. 40, no. 4, pp. 834–848, 2017.
- [6] Y. Zou, Z. Yu, B. Kumar, and J. Wang, "Unsupervised domain adaptation for semantic segmentation via class-balanced self-training," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 289–305, 2018.
- [7] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7167–7176, 2017.
- [8] S. Sankaranarayanan, Y. Balaji, A. Jain, S. N. Lim, and R. Chellappa, "Learning from synthetic data: Addressing domain shift for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3752–3761, 2018.
- [9] S. Zhao, Y. Wang, Z. Yang, and D. Cai, "Region mutual information loss for semantic segmentation," *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [10] J. Liang, D. Hu, and J. Feng, "Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation," in *International Conference on Machine Learning*, pp. 6028–6039, PMLR, 2020.

- [11] S. Li, M. Xie, K. Gong, C. H. Liu, Y. Wang, and W. Li, "Transferable semantic augmentation for domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11516–11525, 2021.
- [12] V. Vs, V. Gupta, P. Oza, V. A. Sindagi, and V. M. Patel, "Mega-cda: Memory guided attention for category-aware unsupervised domain adaptive object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4516–4526, 2021.
- [13] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool, "Domain adaptive faster r-cnn for object detection in the wild," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3339–3348, 2018.
- [14] Y. Liu, W. Zhang, and J. Wang, "Source-free domain adaptation for semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1215–1224, 2021.
- [15] M. Ahmed, Z. Hasan, N. Khan, N. Roy, S. Purushotham, A. Gangopadhyay, and S. You, "Benchmarking domain adaptation for semantic segmentation," in *Unmanned Systems Technology XXIV*, vol. 12124, pp. 151–162, SPIE, 2022.
- [16] 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*, pp. 2517–2526, 2019.
- [17] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International conference on machine learning*, pp. 1180–1189, PMLR, 2015.
- [18] 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*, pp. 1989–1998, Pmlr, 2018.
- [19] Z. Liu, Z. Miao, X. Pan, X. Zhan, D. Lin, S. X. Yu, and B. Gong, "Open compound domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12406–12415, 2020.
- [20] Y.-H. Tsai, W.-C. Hung, S. Schuster, K. Sohn, M.-H. Yang, and M. Chandraker, "Learning to adapt structured output space for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7472–7481, 2018.
- [21] G. Kang, Y. Wei, Y. Yang, Y. Zhuang, and A. Hauptmann, "Pixel-level cycle association: A new perspective for domain adaptive semantic segmentation," *Advances in Neural Information Processing Systems*, vol. 33, pp. 3569–3580, 2020.
- [22] Z. Wang, M. Yu, Y. Wei, R. Feris, J. Xiong, W.-m. Hwu, T. S. Huang, and H. Shi, "Differential treatment for stuff and things: A simple unsupervised domain adaptation method for semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12635–12644, 2020.
- [23] 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*, pp. 4085–4095, 2020.
- [24] Y. Li, L. Yuan, and N. Vasconcelos, "Bidirectional learning for domain adaptation of semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6936–6945, 2019.
- [25] Y. Cheng, F. Wei, J. Bao, D. Chen, F. Wen, and W. Zhang, "Dual path learning for domain adaptation of semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9082–9091, 2021.
- [26] K. Mei, C. Zhu, J. Zou, and S. Zhang, "Instance adaptive self-training for unsupervised domain adaptation," in *European conference on computer vision*, pp. 415–430, Springer, 2020.
- [27] I. Shin, S. Woo, F. Pan, and I. S. Kweon, "Two-phase pseudo label densification for self-training based domain adaptation," in *European conference on computer vision*, pp. 532–548, Springer, 2020.
- [28] Y. Wang, J. Peng, and Z. Zhang, "Uncertainty-aware pseudo label refinery for domain adaptive semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9092–9101, 2021.
- [29] P. Zhang, B. Zhang, T. Zhang, D. Chen, Y. Wang, and F. Wen, "Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12414–12424, 2021.
- [30] Y. Zou, Z. Yu, X. Liu, B. Kumar, and J. Wang, "Confidence regularized self-training," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5982–5991, 2019.
- [31] F. Pan, I. Shin, F. Rameau, S. Lee, and I. S. Kweon, "Unsupervised intra-domain adaptation for semantic segmentation through self-supervision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3764–3773, 2020.
- [32] Z. Zheng and Y. Yang, "Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation," *International Journal of Computer Vision*, vol. 129, no. 4, pp. 1106–1120, 2021.
- [33] X. Ma, J. Gao, and C. Xu, "Active universal domain adaptation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8968–8977, 2021.
- [34] H. Rangwani, A. Jain, S. K. Aithal, and R. V. Babu, "S3vaada: Submodular subset selection for virtual adversarial active domain adaptation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7516–7525, 2021.
- [35] V. Prabhu, A. Chandrasekaran, K. Saenko, and J. Hoffman, "Active domain adaptation via clustering uncertainty-weighted embeddings," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8505–8514, 2021.
- [36] B. Fu, Z. Cao, J. Wang, and M. Long, "Transferable query selection for active domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7272–7281, 2021.
- [37] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The journal of machine learning research*, vol. 17, no. 1, pp. 2096–2030, 2016.
- [38] M. Ning, D. Lu, D. Wei, C. Bian, C. Yuan, S. Yu, K. Ma, and Y. Zheng, "Multi-anchor active domain adaptation for semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9112–9122, 2021.
- [39] I. Shin, D.-J. Kim, J. W. Cho, S. Woo, K. Park, and I. S. Kweon, "Labor: Labeling only if required for domain adaptive semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8588–8598, 2021.
- [40] B. Xie, L. Yuan, S. Li, C. H. Liu, and X. Cheng, "Towards fewer annotations: Active learning via region impurity and prediction uncertainty for domain adaptive semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8068–8078, 2022.
- [41] M. Yi-de, L. Qing, and Q. Zhi-Bai, "Automated image segmentation using improved pcnn model based on cross-entropy," in *Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing, 2004.*, pp. 743–746, IEEE, 2004.
- [42] V. Pihur, S. Datta, and S. Datta, "Weighted rank aggregation of cluster validation measures: a monte carlo cross-entropy approach," *Bioinformatics*, vol. 23, no. 13, pp. 1607–1615, 2007.
- [43] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.
- [44] C. V. Nguyen, A. Achille, M. Lam, T. Hassner, V. Mahadevan, and S. Soatto, "Toward understanding catastrophic forgetting in continual learning," *arXiv preprint arXiv:1908.01091*, 2019.
- [45] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.
- [46] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *European conference on computer vision*, pp. 102–118, Springer, 2016.
- [47] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.
- [48] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818, 2018.
- [49] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- [50] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.
- [51] E. Dey, J. Hossain, N. Roy, and C. Busart, "Synchrosim: An integrated co-simulation middleware for heterogeneous multi-robot system," in *2022 18th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pp. 334–341, IEEE, 2022.
- [52] M. Mahmud, M. S. Islam, M. Younis, and G. Carter, "Optical focusing-based adaptive modulation for optoacoustic communication," in *2021 30th Wireless and Optical Communications Conference (WOCC)*, pp. 272–276, IEEE, 2021.