Real-time Domain Adaptation in Semantic Segmentation

Della Croce Andrea 313362   Terramagra Federica 305777   Zafonte Francesca 319331

Politecnico di Torino, AML course 2024/2025

Abstract

Domain adaptation in semantic segmentation addresses the challenge of performance degradation when models trained on a source domain are applied to a distinct target domain. This survey focuses on two unsupervised domain adaptation (UDA) techniques, as presented in [5] and [4], specifically applied to real-time semantic segmentation networks and their efficacy under substantial domain shifts. The LoveDA (Land-cOVEr Domain Adaptive) dataset [3], designed to promote advancements in semantic and transferable learning, serves as the reference dataset for our experiments. We train the PIDNet model [1], a state-of-the-art real-time segmentation network, on the LoveDA Urban-images (source domain) and evaluate its performance on the LoveDA Rural-images (target domain). Subsequently, we apply domain adaptation techniques. However, the results indicate that these approaches do not lead to significant improvements, highlighting their limitations when dealing with datasets characterized by unbalanced class distributions. This project investigates these results and proposes considerations for future research. The code and data are available at `https://github.com/Zafonte/Real-time-Domain-Adaptation-in-Semantic-Segmentation`.

1. Introduction

In this paper, we address two key challenges in semantic segmentation: achieving high performance while maintaining efficiency, and collecting pixel-level annotations. To address the first challenge, we employ a real-time segmentation network (PIDNet), while for the second, we explore the reuse of data from an annotated source domain and the issues arising from the domain gap, exploring Unsupervised Domain Adaptation techniques to mitigate this problem. We apply these methods within the context of High Spatial Resolution (HSR) land cover mapping using the LoveDA dataset, which presents three major challenges: multi-scale objects, complex background samples, and inconsistent class distributions.

1.1. Classic and Real-time Semantic Segmentation

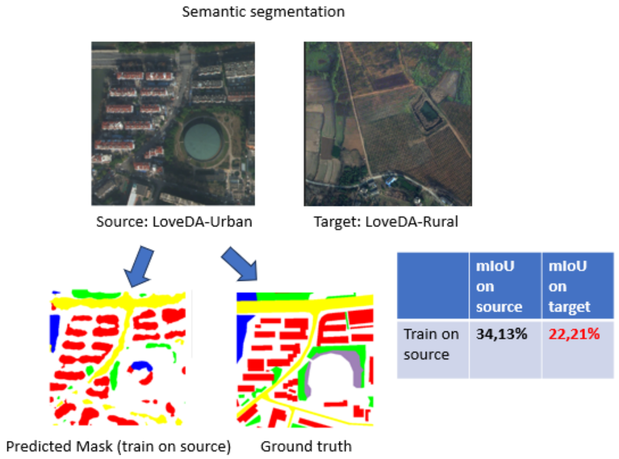
Semantic segmentation is a critical task in computer vision and machine learning, enabling pixel-wise classification of images for applications ranging from autonomous driving to remote sensing. It has evolved significantly over the years, with two distinct approaches emerging to address different needs.

Classic semantic segmentation networks focus on high-quality, using deep, complex architectures like Fully Convolutional Networks (FCNs), DeepLab, and U-Net. These networks prioritize segmentation accuracy and precision, making them suitable for tasks requiring detailed analysis, such as medical imaging or high-definition object recognition. However, these methods tend to be computationally expensive and slow, making them unsuitable for real-time applications.

In contrast, real-time semantic segmentation networks have been developed to address the increasing demand for fast and efficient segmentation in real-time applications, such as autonomous driving, robotics, and mobile devices. Networks like ENet, Fast-SCNN, MobileNetV2, and PidNet are optimized for speed and efficiency, trading off some segmentation accuracy in favor of reduced computational overhead, lower memory usage, and faster inference times.

1.2. Domain Shift challenge and Domain Adaptation solutions

Obtaining large quantities of labeled data is a costly and time-consuming task, primarily due to the need for manual annotation. Although pretraining a model on a source domain and fine-tuning it on a target domain has become a common practice, this approach still requires substantial amounts of labeled training data in the target domain. As a result, this solution is not always feasible for domains with limited or unavailable labeled data. A potential alternative is direct transfer across domains, where models trained on a large-scale labeled source domain (e.g., a simulation domain) are applied to sparsely labeled or even unlabeled target domains (e.g., real-world domains). However, this



	mIoU on source	mIoU on target
Train on source	34,13%	22,21%

Figure 1. An example of domain shift for pixel-wise semantic segmentation task, direct transfer of the model trained on the source domain to the target domain results in a performance drop.

direct transfer often performs poorly due to two key challenges: domain shift and dataset bias. (see Figure 1)

Domain shift refers to the difference in data distributions between the source and target domains. Dataset bias occurs when the source domain data does not account for all possible variations that the model can encounter in the target domain.

To address the challenges posed by domain shift and dataset bias, Domain Adaptation (DA) [6] has emerged as a promising solution. DA aims to reduce domain shift by modifying either the model or the data, enabling a model trained on one domain to perform effectively on a related but different domain. The aim is to enhance the model's ability to generalize effectively in the target domain, despite the challenges introduced by domain shift and dataset bias. This project explores Unsupervised Domain Adaptation (UDA) techniques to adapt a model trained on the labeled LoveDA-Urban domain to the unlabeled LoveDA-Rural domain, where the source and target domains exhibit different data distributions. We focus on two main approaches: adversarial learning and image-to-image translation. Adversarial methods align feature distributions across domains, while image-to-image translation techniques transform source domain images to better match the target domain's characteristics, thereby improving the model's generalization ability in the target domain.

2. Related works

2.1. LoveDA dataset

LoveDA: Land-cOVEr Domain Adaptive Semantic Segmentation dataset [3], specifically created for both semantic and transferable learning, differs from most high spatial resolution (HSR) land-cover datasets by considering the diverse environmental styles across regions. It consists of two domains: Urban, characterized by high-density infrastructure, and Rural, with more natural landscapes. These domains introduce several challenges: 1) multi-scale objects, where objects of the same class appear in vastly different contexts; 2) complex backgrounds, where HSR and intricate scenes lead to greater intra-class variance for the background samples; and 3) inconsistent class distributions, with urban areas predominantly containing artificial objects (buildings, roads), while rural areas feature more natural elements (forests, water, agricoltural). This dataset is particularly well-suited for both semantic segmentation methods and unsupervised domain adaptation (UDA). There are two tasks that can be evaluated on the LoveDA dataset: 1) Semantic segmentation. 2) Unsupervised domain adaptation. The UDA process focuses on two cross-domain adaptation sub-tasks: a) Urban → Rural and b) Rural → Urban. The stark differences in shape, layout, scale, spectra, and class distribution between urban and rural areas significantly challenge model generalization, making domain adaptation for large-scale land-cover mapping more difficult. Notably, the LoveDA dataset is valuable for studying domain shift and class imbalance, both of which are common in real-world land-cover mapping applications.

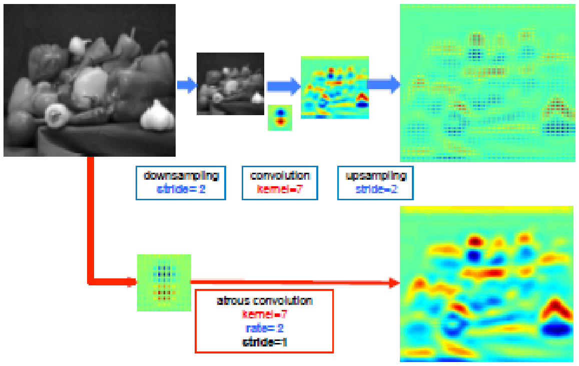


Figure 2. Illustration of atrous convolution in 2-D. Top row: sparse feature extraction with standard convolution on a low resolution input feature map. Bottom row: Dense feature extraction with atrous convolution with rate r = 2, applied on a high resolution input feature map

2.2. DeepLabV2

DeepLabv2 [2] enhances semantic segmentation by integrating three advanced techniques: 1) Atrous convolution, which employs dilated filters to expand the receptive field without increasing computational complexity, enabling the model to capture the richer contextual information; Figure 3. 2) Atrous Spatial Pyramid Pooling (ASPP) which facilitates multi-scale object segmentation by captur-

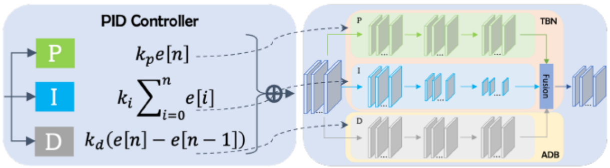


Figure 3. The analogy between PID controller and proposed network

ing image context at different scales. DeepLabv2 adopts an encoder-decoder architecture, where a backbone network (e.g., ResNet) extracts high-level features, while the decoder progressively upsamples these features to produce pixel-level predictions; 3) Fully Connected Conditional Random Fields (CRFs) which model spatial dependencies between pixels, refining object boundaries and enhancing segmentation accuracy. These components work together to enhance both feature extraction and refinement, leading to more precise semantic segmentation. (see Figure 2)

2.3. PIDNet

PIDNet(Parameterized Impulse Response Network) [1] is a a three-branch network inspired by PID (Proportional-Integral-Derivative) controllers, designed to enhance segmentation by balancing details, contextual information, and object boundaries. Unlike conventional two-branch architectures, PIDNet introduces a third derivative branch to mitigate overshoot and improve segmentation precision. Each branch serves a distinct purpose: Proportional (P) branch: preserves high-resolution details; Integral (I) branch: captures long-range contextual dependencies; Derivative (D) branch: detects object edges and refines boundaries. The third branch (D) helps preserve small object features and improves segmentation accuracy at object boundaries. PIDNet dynamically filters features using a parameterized impulse response, allowing the model to enhance critical information while suppressing irrelevant details, ensuring efficient computational usage. To further optimize efficiency, PIDNet employs both depthwise and pointwise convolutions, reducing redundancy while maintaining high segmentation performance.

PIDNet-S is a simplified version of the original architecture, designed specifically for real-time segmentation. While retaining the core feature-filtering mechanisms, PIDNet-S reduces computational complexity, achieving a superior balance between accuracy and efficiency. This optimization enables high-speed inference while preserving competitive segmentation quality. Overall, PIDNet stands out for its progressive and lightweight architecture, achieving state-of-the-art real-time segmentation performance with minimal computational overhead. These capabilities make PIDNet highly suitable for resource-constrained applications requiring low-latency decision-making, such as autonomous driving, robotics, and edge computing. Compared to existing models, PIDNet delivers an optimal trade-off between speed and accuracy, demonstrating strong potential for real-time applications. (see Figure 3)

3. Methods

3.1. Data Augmentation

Data augmentation is a widely used technique in semantic segmentation to improve model generalization by artificially increasing the diversity of the training data. It involves applying transformations to input images and labels to simulate real-world variations. Common augmentations include: geometric transformations such as flipping, rotation, scaling, and cropping, which help models adapt to variations in object orientation and size; photometric adjustments such as brightness, contrast, and color changes, which simulate different lighting conditions.

These augmentations not only prevent overfitting but also enhance the model's ability to generalize to unseen environments.

3.2. Adversarial approaches

Adversarial approaches have gained popularity in semantic segmentation techniques due to their ability to reduce the domain gap between source and target data. In [5] the authors propose a multi-level adversarial learning framework for domain adaptation in semantic segmentation. The approach uses a convolutional network as a discriminator to distinguish between source and target domain features. The discriminator provides a binary output (real from source or fake from target), guiding the model to align feature distributions across domains. This adversarial training helps the model learn domain-invariant features, improving performance on target domain data. The method shows significant improvements in synthetic-to-real and cross-city domain adaptation tasks, enhancing segmentation accuracy in new, unseen environments.

3.3. Image-to-image translation approaches

Existing methods often rely on pseudo-labeling, but suffer from incorrect labels due to domain shift. DACS (Domain Adaptation via Cross-domain Mixed Sampling) solves this problem by mixing images from the two domains, combining real labels and pseudo-labels. It enhances learning by augmenting target-domain samples through class-wise mixing with source-domain images, enabling more effective knowledge transfer from the source domain to the target domain. This improves training and leads to state-of-the-art results on the GTA5 → Cityscapes benchmark. [4] Mixing augmentation techniques applied directly to UDA cause class conflation (confusion between similar classes, e.g. ”sidewalk” and ”road”). DACS solves this problem by mixing images across domains rather than just within the target domain.

4. Experiments

The primary objective of this project is to investigate Domain Adaptation techniques applied to real-time Semantic Segmentation, with a particular focus on improving the generalization ability of a model across different domains. To this end, a series of experiments were conducted to examine the effect of domain shift and evaluate the performance of a real-time segmentation network with various Domain Adaptation strategies.

The images were preprocessed by resizing to either 256x256 or 512x512 pixels and normalizing the pixel values based on mean and standard deviation of ImageNet, because DeepLabv2 and PIDNet are pretrained on it. The label mapping process adjusted semantic segmentation labels to align with the model's requirements and handled cropped black borders in the masks. A mapping function converted black pixels (`0`) to `-1` to ignore them during loss computation, while other labels were shifted to start from `0`. The optimization process leveraged Stochastic Gradient Descent (SGD) with a momentum of 0.9 and a weight decay factor of 0.0001 to mitigate overfitting and improve generalization. The `CrossEntropyLoss` function was applied to guide the network's learning process, with the best-performing model checkpoint determined based on the highest mIoU score on the validation set.

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C}\mathbf{1}[y_i = c]\log(p_{i,c})$$

Where: $\mathcal{L}$ is the total cross-entropy loss; N is the number of pixels in the output; C is the total number of classes; $\mathbf{1}[y_i = c]$ is an indicator function that is 1 if the true class y_i for pixel i is c, and 0 otherwise; $p_{i,c}$ is the predicted probability for class c at pixel i. The resulting loss formulation ensured pixel-wise accuracy for the multi-class segmentation task.

A polynomial learning rate scheduler was adopted, which stabilized convergence.

The evaluation metrics include Mean Intersection over Union (mIoU), which assesses the overlap between predicted and ground truth segments; Latency, which measures the computational efficiency of the real-time model by recording the time required to process each image; FLOPs (Floating Point Operations), representing the computational complexity by counting the number of floating point operations needed per image; and the Number of Parameters, which reflects the model's complexity based on the total trainable parameters.

Note: For all the following experiments, the `CrossEntropyLoss` and `poly_lr_scheduler` were used.

4.1. Classic semantic segmentation network

Implementation details. DeepLabV2 model was employed using the LoveDA-urban dataset. The model was configured with a ResNet101 backbone, pre-trained on ImageNet, ensuring robust feature extraction capabilities for high-resolution urban imagery. The training process spanned over 20 epochs with a resolution of 256x256 and a batch size of 16. **Results.** (see Table 1)

Metric	Value
mIoU (%)	**37.31**
Latency	35.468 ms
FLOPs	47.79 GMac
Params	42.91M

Table 1. Performance metrics for DeepLabV2 trained on LoveDA-urban dataset over 20 epochs with a resolution of 256 and a batch size of 16.

4.2. Real-time semantic segmentation network.

Implementation details. PIDNet-S model, pre-trained on ImageNet, was employed using the LoveDA-urban dataset. The training process spanned over 20 epochs with a resolution of 256x256 and a batch size of 16. **Results.** (see Table 2)

Metric	Value
mIoU (%)	**34.13**
Latency	13.293 ms
FLOPs	1.49 GMac
Params	7.62M

Table 2. Performance metrics for PIDNet trained on LoveDA-urban dataset (20 epochs, resolution 256, batch size 16).

While DeepLabV2 (Table 1) shows slightly better segmentation performance (higher mIoU), PIDNet excels in speed and efficiency (lower latency, FLOPs, and parameters). This makes PIDNet more suitable for real-time applications

4.3. Domain Shift Analysis

To quantify domain shift, we trained PIDNet on the LoveDA-urban dataset (source) and evaluated its performance on the LoveDA-rural dataset (target). Results show a substantial drop in mean Intersection over Union (mIoU), with specific categories (e.g., agricultural and barren) performing poorly (see Table 3). The discrepancies highlight the distinct visual and structural differences between the domains.

Metric	Urban-to-Urban	Urban-to-Rural
mIoU (%)	**34.13**	22.21
Background IoU (%)	31.18	38.85
Building IoU (%)	40.48	28.69
Road IoU (%)	42.30	25.11
Water IoU (%)	56.81	29.30
Barren IoU (%)	16.75	6.82
Forest IoU (%)	34.12	8.96
Agricultural IoU (%)	17.28	17.73

Table 3. Performance for PIDNet trained on LoveDA-urban dataset and evaluated on LoveDA-urban and PIDNet trained on LoveDA-urban dataset and evaluated on LoveDA-rural. Resolution 256, batch size 16

Note: After conducting experiments with different resolutions and batch sizes for the domain shift problem, we found that the best configuration was with a resolution of 512 and batch size 32. From now on, for data augmentation and domain adaptation techniques, we will employ this configuration.

4.4. Data augmentation

The objective of these experiments was to analyze the effectiveness of data augmentation techniques in addressing the domain shift. The focus was on evaluating the performance of the PIDNet segmentation model under various augmentation strategies and determining their impact on class-specific and overall mean Intersection over Union (mIoU). The results, detailed in Table 4, reveal interesting insights into the performance of PIDNet with different data augmentations. The horizontal flip augmentation consistently achieved the highest mIoU (31.91%), excelling across most classes. This demonstrates the effectiveness of simple geometric transformations in addressing domain shifts, particularly for classes with consistent spatial characteristics between domains. In contrast, Gaussian blur showed weaker performance with an overall mIoU of 27.44%. This suggests that simulating blur-like conditions might not align well with the Rural domain's visual distribution, as these distortions do not help the model generalize effectively. Interestingly, the combination of horizontal flip and Gaussian blur resulted in a slight improvement to 28.97% mIoU compared to Gaussian blur alone. However, the combined strategy still underperformed relative to horizontal flip alone, indicating that the added complexity of Gaussian blur might introduce noise rather than aid adaptation. Vertical flip achieved a mIoU of 27.60%. The Multiply augmentation, which simulates brightness variations, showed a modest mIoU of 28.80%. This suggests that brightness augmentation may not align well with the intrinsic variability of these challenging classes. Lastly, applying all augmentations together resulted in the lowest mIoU of 26.62%. This outcome indicates potential overfitting or conflicting augmentation effects.

Overall, the results underline the importance of selecting appropriate augmentation strategies tailored to the specific domain shift challenges and the target classes' characteristics. While augmentations like horizontal flip demonstrate clear benefits, overly complex or mismatched combinations may lead to diminished performance, particularly for harder-to-generalize classes.

4.5. Domain Adaptation with Adversarial Approach

The results from the adversarial training approach highlight both the potential and challenges of applying adversarial domain adaptation to semantic segmentation tasks.

Implementation details. Domain adaptation is achieved through adversarial learning, where the segmentation network acts as a generator, producing domain-invariant features, and the discriminator (`FCDiscriminator` [5]) distinguishes between source (e.g., LoveDA-urban) and target (e.g., LoveDA-rural) features. The `AdamW optimizer` was used for the discriminator to enhance its training stability, with an initial LR_D of 0.005, while the `SGD optimizer` with an initial LR of 0.01, was used for the generator. A polynomial learning rate scheduler was adopted also for the discriminator.

Several loss functions were used to guide the optimization of the model and the discriminator. The segmentation loss uses the `CrossEntropyLoss()` to measure the difference between the predicted segmentation output and the ground truth labels for the source domain. It computes the cross-entropy loss for each pixel in the image. The discriminator loss uses `BCEWithLogitsLoss()`, which applies binary cross-entropy loss to the logits from the discriminator. The discriminator is trained to distinguish between source and target domain predictions. In particular, it is trained to classify source domain predictions as real (0) and target domain predictions as fake (1). The adversarial loss (`loss_func_adv`) also uses `BCEWithLogitsLoss()`. In this adversarial setup, the generator (model) aims to ”fool” the discriminator. The adversarial loss ensures that the model's predictions on the target domain are similar to those of the source domain.

$$\mathcal{L}_{\text{adv}} = -\log D(f(\hat{X}))$$

Where: $f(\hat{X})$ is the model's prediction for the target domain; $D(f(\hat{X}))$ is the probability output by the discriminator for those predictions. The discriminator attempts to classify the target domain predictions as originating from the source domain, and this loss is weighted by the factor λ_{adv}.

Method (PIDNet - 20 epochs)	mIoU (%)	Background	Building	Road	Water	Barren	Forest	Agricultural
Horizontal flip	**31.91**	49.36	30.82	41.99	29.66	8.91	18.81	43.83
Gaussian blur	27.44	47.59	32.22	27.70	38.80	4.69	9.57	31.52
Horizontal flip + Gaussian blur	28.97	49.00	31.56	29.86	36.54	6.69	12.58	36.97
Vertical flip	27.60	50.17	34.13	29.22	27.79	9.13	8.88	33.85
Multiply	28.80	46.75	38.13	28.88	37.80	5.03	8.81	36.16
Total augmentations	26.62	43.20	33.02	34.17	32.48	1.93	7.09	34.47

Table 4. Performance metrics for PIDNet trained with various augmentations. Resolution 512x512 and batch size 32.

The generator (model) loss (`loss_G`) is the total loss for the generator, combining the segmentation loss (L_{seg}) and the adversarial loss (L_{adv}):

$$L_G = L_{\text{seg}} + \lambda_{\text{adv}} L_{\text{adv}}$$

The segmentation loss ensures accurate predictions on the source domain, while the adversarial loss helps align the target domain predictions with the source domain, making it harder for the discriminator to distinguish between them.

The discriminator loss (`loss_D`) is the total loss for the discriminator, which is the average of the losses for the source and target domains:

$$L_D = \frac{1}{2}(L_{\text{Dsrc}} + L_{\text{Dtgt}})$$

The model and the discriminator engage in a minimax game, where the model tries to fool the discriminator, and the discriminator tries to correctly classify the domains.

These losses together allow for domain adaptation, where the model is trained to perform semantic segmentation well on both the source and target domains by leveraging adversarial training.

Results. The model achieved a mIoU of 19.91%, which is considerably lower compared to the data augmentation strategies in previous step. While adversarial approaches can mitigate domain shifts, the performance here indicates that further tuning or enhancements may be necessary to achieve significant gains. The adversarial training setup provides an improvement for certain classes but fails to generalize effectively for others. The results underscore the complexity of aligning feature spaces between highly diverse domains such as Urban and Rural. The visual and structural differences in these domains remain challenging, even for adversarial techniques.

4.6. Domain Adaptation with Image-to-Image Translation

We explore DACS technique enabling more effective knowledge transfer from the LoveDA urban dataset (source domain) to the LoveDA rural dataset (unlaballed target domain).

Implementation details. Since the target domain lacks ground truth annotations, pseudo-labels are generated by computing class predictions (via softmax) and selecting the maximum predicted probability. The confidence threshold (C_t) for pseudo-label creation starts at 0.7 ($\min_{c\_t}$) and is linearly increased to 0.9 ($\max_{c\_t}$) over the training epochs, following:

$$C_t = \min_{c\_t} + (\max_{c\_t} - \min_{c\_t}) \times \frac{\text{epoch}}{\text{NUM\_EPOCHS}}$$

After obtaining pseudo-labels, the DACS methodology is applied. An image from the source domain is selected, and half of its classes are randomly chosen to be pasted onto a target-domain image. The Background class is included with a 25% probability to account for its overabundance in the source dataset. By superimposing source-region pixels onto the target image, a "mixed" image and a corresponding "mixed" label are constructed. Simultaneously, a pixel-wise weight map is generated to indicate which pixels contribute to the loss function. For pixels where source annotations are available and where pseudo-label predictions are over the confidence threshold, the weight is set to 1; otherwise, it is set to 0. This design yields a supervised loss for source pixels (where ground truth is available) and a semi-supervised loss for target pixels (where only mixed-labels guide learning).

Training is conducted with batches of 16 samples drawn in parallel from each domain. Although effective in principle, this approach encounters two notable challenges. First, superimposing source-image portions onto target images can result in highly abstract and unrealistic scenes that impede the model's ability to learn coherent spatial relationships. Second, the source domain is dominated by certain classes (e.g., building and background), which can bias the model into over-predicting these classes in the target domain, especially when pseudo-labels reinforce this imbalance. On the other hand, the generated pseudo-labels contain extensive areas falling below the confidence threshold.

Results. While training progressively enhances the model's performance on the validation dataset (in terms of mIoU), this improvement is not valid to ensure a reasonable amount of predictions over the confident threshold. Accord-

Method (Urban → Rural)	mIoU (%)	Background	Building	Road	Water	Barren	Forest	Agricultural
Augmentation (Horizontal Flip)	**31.91**	49.36	41.99	30.82	29.66	8.91	18.81	43.83
Adversarial Approach	19.91	46.11	24.54	17.81	29.10	7.82	1.18	12.81
Image-to-Image (DACS)	21.05	48.03	23.10	25.54	28.84	13.63	1.44	6.80

Table 5. Comparison of mIoU and class-specific performance across three approaches: data augmentation with horizontal flip, adversarial training, and image-to-image translation (DACS) (Urban → Rural).

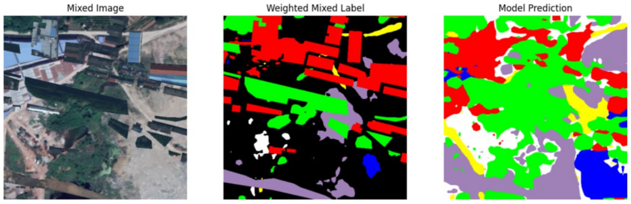


Figure 4. The results of the application of DACS tecnique.

ingly, the contribution of this technique remains limited and can even result counterproductive (see Figure 4), as proved by a mIoU that is substantially lower compared to experiments that only apply domain shift. Results further indicate that the model struggles with the Barren, Forest, and Agricultural classes (especially Forest 1.44% and Agricultural 6.80%), which are likely the most prominent in a rural environment.

Inconsistent class distributions Inconsistent class distributions between urban and rural scenes lead to better performance of image-to-image methods over adversarial methods in domain adaptation. Rural scenes have fewer artificial objects and more natural ones, while urban scenes are dominated by man-made structures. Adversarial methods struggle with this imbalance, resulting in lower accuracy. In contrast, image-to-image methods generate pseudo-labels for target images, reducing class distribution divergence and improving performance. Despite the performance difference, both approaches still highlight the need for further refinement to fully address the complex challenges of cross-domain adaptation in land-cover segmentation tasks. (see Table 5)

4.7. Extensions

To enhance the performance achieved by the baseline best setting in 4.4 experiment, some strategies have been explored. Our efforts focused on improving both domain adaptation pipelines by introducing different loss functions and performing hyperparameter tuning. The loss functions tested included Focal Loss, Weighted Loss, and a custom Weighted Focal Loss, all aimed at addressing class imbalance.

The **focal loss** tackles both class imbalance and the challenge of difficult examples in semantic segmentation by reducing the impact of well-classified samples with high confidence. This refocuses the model on harder examples, such as those with complex details or overlapping classes, improving predictions in uncertain areas. It also increases the influence of underrepresented classes, promoting balanced learning and better generalization on challenging data.

The **weighted cross-entropy** loss adjusts the standard cross-entropy by scaling the loss for each class based on its frequency. (see Fig. 5) This reduces bias toward most represented classes, enhancing the model's ability to recognize minority classes and leading to more balanced and robust performance across the dataset. For the adversarial approach, these modifications largely failed to produce significant results. Only the version that delivered the best improvements for the image-to-image approach was able to generate some performance gains also for the adversarial approach, although these gains were still smaller than those achieved with the image-to-image method. (see table 6)

4.7.1 Hyperparameters tuning

For hyperparameter tuning, trials were conducted with the AdamW optimizer in combination with the MultiStepLR scheduler, along with the original hyperparameters. The pairing of SGD and MultiStepLR outperformed the original PolyLRScheduler. While PolyLRScheduler gradually reduces the rate and risks premature stabilization, MultiStepLR decreases it by a factor of 0.1 at predefined intervals (epoch 10, epoch 15), enabling more substantial updates and avoiding suboptimal minimum. This approach delayed convergence, leading to more robust adaptation and better generalization, for this reason in the following steps, for Loss calculations, we employed MultiStepLR.

4.7.2 Class balancing

Addressing the central challenge of class imbalance in this dataset entailed the introduction of new loss functions that incorporate class-specific weighting. These weights were initially computed for the cross-entropy function as follows:

$$\text{class\_weights}[c] = \frac{1}{\text{class\_freq}[c] + 10^{-6}}, \quad \forall c$$

$\text{class\_freq}[c]$: this calculates the relative frequency of class c with respect to the total number of pixels. It is defined

Approach	mIoU (%)	Background	Building	Road	Water	Barren	Forest	Agricultural
Adversarial Approach	19.91	**46.11**	**24.54**	17.81	**29.10**	**7.82**	1.18	12.81
Weighted Loss	**22.39**	42.49	20.35	**24.55**	32.26	**7.62**	7.62	**21.82**

Table 6. IoU per category and mIoU for different adaptation approaches.

Loss Function	mIoU (%)	Background	Building	Road	Water	Barren	Forest	Agricultural
Image-to-Image	21.05	48.03	23.10	25.54	28.84	13.63	1.44	6.80
Focal Loss	23.89	**49.43**	**24.83**	**28.93**	34.28	12.91	5.12	11.73
Weighted Loss	**26.12**	43.33	21.91	17.47	40.13	**19.83**	**19.86**	**20.35**
Weighted Focal Loss	23.93	30.72	22.28	12.60	**51.66**	17.65	14.39	18.19

Table 7. mIoU for different loss functions.

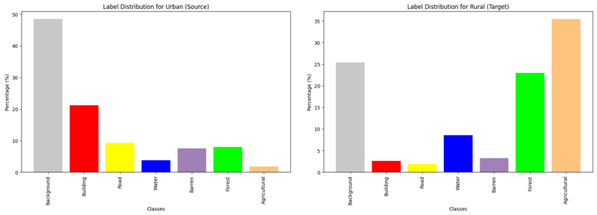


Figure 5. Urban source vs Rural target distribution.

as the ratio of the number of pixels belonging to class c (class_counts$[c]$) to the total number of pixels (total_pixels); class_weights$[c]$: this determines the weights for each class c, avoiding division by zero. It is defined as the inverse of the class frequency class_freq$[c]$, with an added safety term 10^{-6} to prevent numerical instability. The result for class_weights was [0.12, 0.27, 0.61, 1.51, 0.74, 0.72, 3.03]. In practice, class_weights is too imbalanced, severely degrading performance on the Background and Building classes. As a result, manual fine-tuning was adopted: the weight for Agricultural was reduced while the weights for Background and Building were slightly raised to retain adequate accuracy, and those for Barren and Forest were increased to improve their performance. (see Figure 5) The manually adjusted weights were [0.3, 0.5, 0.8, 1.4, 1.2, 1.2, 1.6].

4.7.3 Different Loss Functions

Focal Loss preserved mIoU for the Background, Building, Road, and Barren classes relative to step 4.6, while producing modest yet meaningful improvements in Water, Forest, and Agricultural.

$$\mathcal{L}_{\text{Focal}}(p_t) = -\alpha_t (1 - p_t)^\gamma \log(p_t)$$

Weighted Loss delivered the most pronounced overall gain, elevating mIoU by 5.07%. The greatest improvements were observed for Forest (+18.39%) and Agricultural (+13.55%) at the cost of slight reductions for Background (-4.70%) and Road (-8.07%).

$$\mathcal{L}_{\text{Weighted}} = -\sum_{i=1}^{N} w_{c_i} \cdot y_i \log(\hat{y}_i)$$

Due to the great result achieved by Weighted Loss we have decided to fuse the two previous Loss Functions into one, a sort of Weighted Focal Loss. The incorporation of class weighting introduced excessive penalization for dominant classes such as Background and Road. Consequently, although gains were observed for Barren, Forest, and Agricultural, these did not surpass those achieved with Weighted Loss alone. The disproportionate penalty imposed on most frequent classes reduced its overall efficacy, implying that Weighted Focal Loss does not provide an optimal trade-off for improving performance across all classes in this setting.

$$\mathcal{L}_{\text{Weighted Focal}} = -\sum_{i=1}^{N} w_{c_i} \cdot \alpha_t (1 - p_t)^\gamma \log(p_t)$$

5. Conclusions

These experiments show that urban-to-rural domain shifts severely affect segmentation accuracy, especially for classes that are rare or visually complex in the source domain. While in other cases adversarial and image-to-image methods slightly boost performance, they struggle with LoveDA's imbalanced distributions, often favoring dominant classes and overlooking minority ones. Weighted loss emerges as the most effective solution in this setting, improving recognition of underrepresented categories without heavily penalizing frequent classes. Nonetheless, the minimal gains underscore the difficulty of adapting real-time models to heterogeneous land-cover data. Future work should focus on strategies that better address class imbalance, potentially through refined sample selection or more sophisticated hybrid losses, to achieve robust adaptation in challenging, high-resolution domains.

References

[1] Jiacong Xu Zixiang Xiong Shankar P. Bhattacharyya. Pidnet: A real-time semantic segmentation network inspired by pid controllers. 2023. 1, 3

[2] Liang-Chieh Chen, George Papandreou, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. 2017. 2

[3] Ailong Ma Xiaoyan Lu Yanfei Zhong† Junjue Wang, Zhuo Zheng. Loveda: A remote sensing land-cover dataset for domain adaptive semantic segmentation. 2022. 1, 2

[4] Wilhelm Tranheden, Viktor Olsson, Juliano Pinto, and Lennart Svensson. Dacs: Domain adaptation via cross-domain mixed sampling. 2020. 1, 3

[5] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. 2019. 1, 3, 5

[6] Sicheng Zhao, Xiangyu Yue, Shanghang Zhang, Bo Li, Han Zhao, Bichen Wu, Ravi Krishna, Joseph E. Gonzalez, Alberto L. Sangiovanni-Vincentelli, Sanjit A. Seshia, and Kurt Keutzer. A review of single-source deep unsupervised visual domain adaptation. 2020. 2