Real-time Domain Adaptation in Semantic Segmentation

Angelo Bongiorno s331164 Matteo Bollo s332129 Vito Silvestri s326492

Abstract

Semantic segmentation enables pixel-level recognition of objects based on their semantic class and is critical in applications such as autonomous driving and urban scene understanding. A key challenge in deploying segmentation models in real-world scenarios is domain shift: the discrepancy between training and target data distributions. In this paper, we explore domain adaptation strategies for semantic segmentation using PIDNet as our segmentation model and the LoveDA dataset to perform training and testing. We compare the effectiveness of selected adaptation techniques and analyze their impact on cross-domain performance. We find that combinations of color and geometric augmentations and ensemble models based on image-to-image translation lead to the highest performance.

1. Introduction

Semantic segmentation is a fundamental task in computer vision that involves assigning a semantic label to each pixel in an image. It plays a crucial role in numerous real-world applications such as autonomous driving, remote sensing, and medical imaging. While modern deep learning models can achieve excellent performance on benchmark datasets, their effectiveness often decreases when deployed in real-world scenarios that differ from the training domain, a phenomenon known as domain shift. This degradation is caused by variations in visual appearance, feature distributions, and class priors between the source and target domains [9].

Domain adaptation techniques aim to bridge this gap, enabling models trained on a labeled source domain to generalize effectively to an unlabeled target domain. In the context of real-time applications, it becomes essential to investigate domain adaptation strategies that maintain high accuracy while preserving low latency and computational efficiency. In this work, we explore several domain adaptation strategies for semantic segmentation in real-time settings, using PIDNet [7], a lightweight yet accurate segmentation network. Our experiments are conducted on the LoveDA dataset [6], a domain adaptation benchmark which features a cross-domain setting between urban and rural environments.

(a) Urban (b) Rural

Figure 1. Urban and rural images.

We first quantify the impact of domain shift by evaluating a PIDNet model trained on the urban subset of LoveDA and tested on the rural subset (see Figure 1) To mitigate this performance drop, we apply a series of adaptation strategies, including:

- **Image-level augmentations**, to enhance generalization by simulation appearance variations.
- **Adversarial learning**, to align feature distributions between source and target domains through a domain discriminator.
- **Image-to-image translation techniques**, specifically DACS (Domain Adaptation via Cross-domain Sampling), which leverages class-consistent image mixing to transfer spatial and semantic patterns across domains.

In the final phase of the project, we extended our investigation beyond the baseline approaches by exploring additional strategies to enhance domain adaptation. We incorporated BiSeNetV2 [8], a real-time segmentation network, into our pipeline and conducted a comparative analysis of its performance against PIDNet under domain adaptation settings. Furthermore, we designed and implemented a novel variant

of the DACS method in which the set of semantic classes used for image mixing is altered at each training.

Our study provides an in-depth analysis of how different real-time domain adaptation techniques perform under a challenging cross-domain segmentation setting, while also introducing an original contribution that explores class-level prioritization to improve adaptation efficacy.

2. Related Work

2.1. Models

Semantic segmentation has been extensively studied in recent years, with various architectures proposed to improve accuracy and efficiency. A comprehensive overview of deep learning-based approaches can be found in Ulku and Akagündüz's survey [5]. More recently, increasing attention has been given to real-time semantic segmentation models, motivated by the demands of applications such as autonomous driving and surveillance. These applications require models that can make rapid predictions while maintaining a reasonable level of accuracy.

However, real-time segmentation models typically face a fundamental trade-off between accuracy and responsiveness. Shallow networks tend to lack sufficient receptive fields, limiting their ability to capture high-level contextual information, while reducing input resolution often results in a noticeable drop in segmentation accuracy. To address this, several architectural innovations have been proposed. One notable example is BiSeNet [8], which introduces a two-branch structure to balance spatial precision and contextual understanding. The Spatial Path captures detailed spatial information at high resolution, while the Context Path, based on a lightweight backbone, provides a large receptive field to incorporate semantic context. The outputs of both branches are fused to generate the final segmentation map.

Building upon this multi-branch paradigm, PIDNet [7] incorporates principles from control theory and introduces a three-branch architecture. Specifically, the Proportional branch preserves fine-grained spatial details, the Integral branch captures long-range contextual information to enhance feature representation, and the Derivative branch focuses on high-frequency features, aiding in delineating object boundaries. The outputs of these branches are fused to produce accurate and efficient segmentation results, making PIDNet well suited for real-time scenarios.

2.2. Dataset

We employ the LoveDA dataset, a remote sensing benchmark composed of images sourced from Google Earth. The dataset is partitioned into two distinct domains: Urban and Rural, each containing dedicated training, validation, and test sets with corresponding pixel-level segmentation masks. LoveDA comprises seven semantic classes: building, road, water, barren, forest, agricultural, and background. As detailed in [6], the dataset was specifically designed to address challenges in domain adaptation. It exhibits significant class imbalance across domains, and the same object classes may appear in substantially different spatial arrangements and forms. Additionally, the dataset includes multi-scale objects, requiring models to be robust to variations in object scale and resolution.

3. Method

This study investigates the impact of domain shift in semantic segmentation by employing multiple segmentation architectures and domain adaptation techniques. Specifically, we consider three convolutional neural networks—DeepLabV2, PIDNet, and BiSeNetV2—that represent different trade-offs between segmentation accuracy and computational efficiency. To address the performance degradation induced by domain discrepancies, we explore several domain adaptation approaches aimed at improving generalization across domains.

3.1. Architectures

3.1.1 DeepLabV2

DeepLabV2 [1] is a high-accuracy semantic segmentation network that builds upon fully convolutional networks by incorporating atrous (dilated) convolutions and multi-scale contextual reasoning.

In particular, DeepLabV2 employs a backbone architecture—typically ResNet or VGG—modified with atrous convolutions to preserve spatial resolution in deeper layers. Additionally, the model integrates Atrous Spatial Pyramid Pooling (ASPP), a module designed to capture context at multiple scales by applying parallel atrous convolutions with different dilation rates. This enhances the network's ability to handle objects with varying spatial extents. To refine the predicted segmentation maps, DeepLabV2 optionally incorporates fully connected Conditional Random Fields (CRFs) as a post-processing step. These CRFs enforce spatial consistency and sharpen object boundaries by modeling long-range dependencies among pixels, based on both spatial proximity and appearance similarity.

Thanks to its strong performance and rich contextual modeling, DeepLabV2 is used in this work as an upper-bound reference for evaluating the effectiveness of real-time networks and adaptation strategies under fully supervised conditions only on the target domain.

3.1.2 PIDNet

PIDNet [7] is a real-time semantic segmentation network that introduces a novel architecture inspired by classical Proportional-Integral-Derivative (PID) control theory. The model is specifically designed to achieve a balance between segmentation accuracy and computational efficiency, making it well-suited for resource-constrained or real-time applications.

Specifically, it is composed of:

- The *Proportional branch (P)*: focuses on preserving high-resolution spatial details that are critical for accurate boundary localization. By maintaining fine-grained representations, this branch enables the network to retain local structures that are typically lost in aggressive downsampling processes.
- The *Derivative branch (D)*: is designed to extract high-frequency features, with a specific emphasis on capturing object boundaries. It enhances the model's sensitivity to rapid spatial changes, which is particularly beneficial for highlighting transitions between different semantic regions.
- The *Integral branch (I)*: is responsible for aggregating contextual information over large spatial regions. It operates on deeper feature maps and captures long-range dependencies, providing a global understanding of the scene. This branch plays a key role in achieving semantic consistency across spatially distant regions.

The three-branch architecture of PIDNet is deliberately designed to balance computational efficiency and representational capability.
Three core modules enhance PIDNet's architecture. The Pixel-Attention-Guided Fusion Module (Pag) injects semantic context from the Integral branch into the Proportional branch via attention mechanisms. The Boundary-Attention-Guided Fusion Module (Bag) combines outputs from all three branches, emphasizing the Derivative branch near boundaries and the Proportional branch elsewhere. Finally, the Parallel Aggregation Pyramid Pooling Module (PAPPM) strengthens contextual understanding by aggregating multi-scale features within the Integral branch. Notably, the Bag module exemplifies PIDNet's control-theory-inspired design, enabling efficient multi-scale fusion with minimal latency.

3.1.3 BiSeNetV2

BiSeNetV2 is a real-time semantic segmentation network that employs a two-branch architecture designed to balance accuracy and efficiency. It decouples the feature extraction process into two specialized branches:

- The *Detail Branch*, composed of shallow convolutional layers, is responsible for preserving high-resolution spatial and edge-related features.
- The *Semantic Branch* applies deeper, downsampled convolutions to efficiently capture high-level contextual information.

The outputs of these branches are fused using the *Bilateral Guided Aggregation (BGA)* module, which integrates semantic information into the spatial path, enabling precise yet context-aware predictions. During training, BiSeNetV2 leverages four auxiliary segmentation heads to enhance feature learning. These additional heads contribute to the loss value computation but are discarded during inference.

3.2. Loss Functions

The baseline DeepLabV2 model is trained using a standard pixel-wise **Cross-Entropy Loss**, which is sufficient for evaluating general segmentation accuracy. Meanwhile, PIDNet and BiSeNetV2 are trained using **Online Hard Example Mining Loss (OHEM)**. OHEM Loss is a variant of the Cross-Entropy Loss which computes the loss only from hard samples (i.e. pixels whose loss value is higher than a threshold). By prioritizing difficult samples, OHEM Loss encourages the model to focus on improving predictions where it struggles most, ultimately enhancing segmentation quality in complex regions and for less common classes. This is particularly useful in the domain shift context.

3.2.1 Loss Computation in PIDNet

More specifically, **PIDNet** employs a **composite loss function** tailored to its multi-branch structure. The total loss is defined as the weighted sum of four components.

$$\mathcal{L}_{\text{total}} = \lambda_0 \mathcal{L}_0 + \lambda_1 \mathcal{L}_1 + \lambda_2 \mathcal{L}_0 + \lambda_3 \mathcal{L}_3 \tag{1}$$

where:

- l_0 is a semantic loss, implemented through OHEM, computed from the output of the first Pag module.
- l_1 is a weighted binary Cross Entropy loss, used to address the problem of boundary detection.
- l_2 is a standard cross-entropy loss computed on the final semantic output.
- l_3 is a boundary-aware cross-entropy loss that uses the output of the boundary head. It selectively emphasizes pixels near object boundaries based on a confidence threshold t, encouraging alignment between the predicted boundaries and semantic segmentation.

The weights λ_P, λ_D, and λ_{aux} are hyperparameters that control the relative importance of each loss term.
This composite loss formulation ensures that the network not only learns semantically rich and spatially coherent features but also maintains precise object boundaries, resulting in improved segmentation accuracy, especially for small or densely packed objects.

3.3. Domain Adaptation methods

To mitigate the negative impact of domain shift between the source and target domain, we implemented and evaluated a series of domain adaptation techniques. These methods were designed to improve the generalization ability of the segmentation network when applied to target domain data, which differs in appearance and distribution from the source domain training set.

3.3.1 Image Augmentations

To further bridge the domain gap, we applied on-the-fly standard image augmentations during training. These transformations introduce variability in the source data, helping the model generalize better to unseen target-domain images. These augmentations simulate variations in lighting, orientation, and geometry that may differ across domains, making the segmentation model more robust to domain shifts.

3.3.2 Adversarial Domain Adaptation

Adversarial learning for domain adaptation [4] employs a discriminator network to align the output space distributions of the source and target domains. Specifically, the segmentation model's predictions are passed to a discriminator, which is trained to distinguish whether the segmentation map originates from a source-domain or a target-domain image. Simultaneously, the segmentation network is trained to fool the discriminator, using an adversarial loss. This encourages the segmentation model to generate output distributions for source-domain inputs that are indistinguishable from those of the target domain, thereby achieving domain alignment at the prediction level.

The adversarial loss used in this setting plays a key role in driving the domain alignment. It consists of two components: a standard binary cross-entropy loss for the discriminator, and an adversarial objective for the segmentation network. The discriminator D is optimized to classify the segmentation outputs as either coming from the source or target domain, by minimizing:

$$\mathcal{L}^D = -\sum_{h,w}(1-z)\log D(G(x_t) + z\log\left(D(g(x_s)\right) \quad (2)$$

where G denotes the segmentation model, and x_s, x_t are inputs from the source and target domains, respectively.

Conversely, the segmentation model G is trained to fool the discriminator by maximizing the probability that its outputs on the target domain are classified as source domain predictions:

$$\mathcal{L}_{adv} = -\sum_{h,w}\log D(G(x_t)) \quad (3)$$

This adversarial term is integrated into the total training loss of the segmentation model (PIDNet loss, see section 3.2.1) to form the total loss of adverrsarial learning, as shown below:

$$\mathcal{L}_{total} = \lambda_{seg}\cdot\mathcal{L}_{seg} + \lambda_{adv}\cdot\mathcal{L}_{adv} \quad (4)$$

With λ_{adv} and λ_{seg} to baance the two losses.

3.3.3 Domain Adaptation via Cross-domain Mixed Sampling (DACS)

DACS [3] addresses the unsupervised domain adaptation problem by leveraging mixed image-label pairs constructed from both source and target domains. In each training iteration, a target-domain image is passed through the segmentation network to generate pseudo-labels. A second image, sampled from the source domain, is then partially overlaid on the target image, creating a synthetic mixed-domain image. Correspondingly, the label for the new image is formed by overlaying the source mask onto the target pseudo-labels. The model is then trained using both the standard source-domain data and the mixed-image data, with a combined loss used for backpropagation. This strategy improves generalization by exposing the model to a blend of labeled and pseudo-labeled domain data.

3.3.4 Contrastive Learning for Unpaired Image-to-Image Translation (CUT)

CUT [2] is an unpaired image-to-image translation technique that modifies the appearance of source domain images to resemble those of the target domain, while preserving their structural content. It achieves this through a patchwise contrastive loss, which encourages corresponding patches from the source image and its translated version (at the same spatial location) to stay similar. Since the translated images retain the semantic structure of the original source images, their ground truth labels remain valid. This effectively provides us with a new synthetic dataset whose appearance matches the target domain, while maintaining accurate annotations from the source domain (see Figure 2).

4. Experiments

The code for the following experiments is available at https://github.com/AngeloBongiorno/AML_2025_project4.

(a) Original urban image. (b) Style-transfered urban image.

Figure 2. An example of style transfer from the urban domain to the rural domain.

4.1. Evaluation metrics

The main evaluation metric is the **Mean Intersection over Union (mIoU)**, also known as the Jaccard index. It is computed by averaging IoU across all semantic classes defined in the LoveDA dataset. The IoU metric quantifies the overlap between predicted and ground truth segmentation masks, defined as:

$$\text{IoU} = \frac{\text{target} \cap \text{prediction}}{\text{target} \cup \text{prediction}} \tag{5}$$

Per-class IoUs were also recorded during domain shift experiments to assess class-specific performance. Additionally, we compared PIDNet and DeepLabV2 in terms of inference latency, FLOPs and parameter count to evaluate their computational efficiency.

4.2. Dataset and Splitting Strategy

We conduct experiments on the LoveDA dataset, which contains high-resolution satellite imagery from two distinct domains: Urban and Rural. To simulate domain shift, we designate the Urban domain as the source and the Rural domain as the target. For the baseline experiment without domain shift, we utilized the Train/Urban subset of the LoveDA dataset. This subset was split into training and validation sets using an 80/20 ratio. The Validation/Urban subset was used as the test set. This setup is necessary due to the lack of ground truth masks for the official test set provided by LoveDA, which makes direct evaluation on it infeasible. In all subsequent domain adaptation experiments, we maintain the same split of the Train/Urban subset for training and validation in order to ensure fair comparisons with the baseline. However, the test set is replaced with the Validation/Rural subset to simulate domain shift and evaluate the generalization capability of the models.

During training on the LoveDA dataset, a possible imbalance between classes in the two domains was noted, with some categories significantly more represented than others. This discrepancy introduces a bias during model training, where the network tends to favor predictions for majority classes, resulting in poor generalization on rare or underrepresented categories. Consequently, standard training procedures may lead to suboptimal segmentation performance, particularly for minority classes.

For this reason, to address the issue of class imbalance more effectively, we computed class-specific weights to adjust the loss function during training. These weights are calculated based on the class frequency distribution observed in the source domain. Specifically, we iterate over the training dataset and count the number of pixels belonging to each class. From these counts, we derive normalized class frequencies, which are then used to compute the weights using one of the following strategies:

- **Inverse frequency**: assigns higher weights to less frequent classes by taking the reciprocal of the class frequency;
- **Median frequency balancing**: scales each class inversely proportionally to its frequency, normalized by the dataset-wide median frequency;

Once computed, these class weights are incorporated into the training process by modifying both the standard Cross-Entropy loss and the OHEM loss. This adjustment encourages the model to pay greater attention to underrepresented categories and penalize misclassifications of minority classes more heavily. The use of class reweighting helps to mitigate the bias introduced by the dataset's natural imbalance and improve overall segmentation performance, particularly for rare classes.

4.3. Results

Before evaluating the impact of domain shift and various domain adaptation techniques, we performed a hyperparameter tuning phase for both the DeepLabV2 and PIDNet models in order to determine effective training configurations. These hyperparameters were then used consistently across experiments involving domain shift, domain augmentation, and style transfer.

All models were trained for 20 epochs with a batch size of 16. During training, all input images were resized to 512×512 pixels to ensure uniformity across batches and reduce computational load.

For **DeepLabV2**, we employed Stochastic Gradient Descent (SGD) as the optimizer, with a fixed learning rate of $1e^{-2}$.
While for **PIDNet**, training was also performed using Stochastic Gradient Descent (SGD), with a learning rate set to $1e^{-2}$, a weight decay of $1e^{-3}$ to regularize the model, and a momentum term of 0.8 to accelerate convergence. The

Model	mIoU	Latency	GFLOPS	Parameters
DeepLabV2	0.3761	0.125s	185.01	43.02 M
PIDNet	0.3455	0.011s	5.933	7.624 M

Table 1. Performance comparison.

loss function adopted for training was the OHEM loss, chosen to prioritize difficult samples during optimization and improve segmentation performance on less frequent classes. These values were selected after tuning based on validation performance, aiming to strike a balance between training stability, segmentation accuracy, and computational efficiency.

To establish a performance baseline, we conducted a comparative experiment between DeepLabV2 and PIDNet, representing a conventional non-real-time segmentation network and a real-time network, respectively. The results of this comparison are reported in Table 1.
Although DeepLabV2 delivers the highest segmentation accuracy (mIoU = 0.3761), this gain comes at a substantial computational cost: it requires 185.0 GFLOPs per forward pass and contains 43.0 M parameters. In contrast, PIDNet reduces the computational load (5.9 GFLOPs) and shrinks the size of the model (7.6 M), despite a moderate decrease in precision (mIoU = 0.3455). The performance gap between inference latencies is large as expected, with PIDNet being $\sim$ 11 times faster than DeepLabV2.

4.3.1 Domain Shift

To assess the impact of domain shift, we begin by considering the results in 1 obtained by evaluation PIDNet on the same domain it was trained on (Urban $\rightarrow$ Urban). The mIoU score of 0.3455 serves as a meaningful upper bound for this model, indicating the maximum segmentation performance it can achieve under ideal conditions where the training and test data are drawn from the same distribution. When we instead evaluate PIDNet in a domain shift setting -i.e., training on Urban and testing on Rural data (PIDNet w domain shift in 2)- we observe a substantial degradation in performance. As reported by the table, the mIoU drops to 0.255, confirming the expected negative effect of distributional discrepancies between the domains of origin and destination.
A more detailed class-wise analysis reveals that this performance drop is not uniform across categories. While some classes like Background (0.502) and Road (0.336) retain moderate segmentation accuracy, others suffer severe degradation. Notably, the Barren (0.065), Forest (0.121), and Agriculture (0.239) categories exhibit very low IoU scores, highlighting the model's difficulty in generalizing to rural-specific semantic patterns it did not encounter during training. These results clearly demonstrate the need for **domain adaptation** techniques to close the performance gap caused by the domain shift and to recover accuracy on the target distribution.

4.3.2 Domain Augmentation

As a first step towards mitigating performance degradation caused by domain shift, we explored the effectiveness of various domain augmentation techniques applied during training on the source domain. We evaluated several augmentation strategies, both independently and in combination. These included: *Horizontal Flip (HF)*,*Gussian Blur (GB)*, *Elastic Transform (ET)*, *Hue, Saturation and Value (HSV)*, *Color Jitter (CJ)*, *Spatter*, *HF + GB*, *ET + HSV* and *HF + GB + CJ*.
The results, shown in Table 2, indicate that most augmentation strategies provide modest improvements over the baseline (PIDNet without domain adaptation, mIoU = 0.255). Among all tested methods, the combination of Horizontal Flip, Gaussian Blur and HSV (HF + GB + HSV) achieved the highest mIoU of 0.280, outperforming all other single and combined techniques. This suggests that both spatial and appearance-level variations contribute positively to robustness under domain shift.

4.3.3 Style Transfer

As an additional strategy to mitigate the effects of domain shift, we experimented with generating a stylized version of the source dataset using the Contrastive Unpaired Translation (CUT) framework [2]. The CUT model was trained on the full source and target domains for 20 epochs using a batch size of 4. After training, the generator was used to translate the entire Urban training set into Rural-styled images. These translated images, paired with their original labels, were then used to train PIDNet in a domain shift setting ($\text{Urban}_{\text{stylized}} \rightarrow$ Rural).

Despite the conceptual appeal of this approach, the resulting segmentation performance was significantly lower than expected. As shown in Table 2, the model trained on CUT-translated data achieved an mIoU of only 0.134, notably lower than even the baseline with no domain adaptation (mIoU = 0.255), and far below the best-performing augmentation-based models (e.g., HF+GB with mIoU = 0.280).

This poor performance may be attributed to several factors. First, the CUT model may not have been trained for a sufficient number of epochs to produce stylistically consistent and semantically reliable transformations. The resulting images might lack the fidelity or domain-specific information necessary for effective learning and generalization. Additionally, although the translated images visually resemble

Method	mIoU	Background	Road	Building	Water	Barren	Forest	Agriculture
DeepLabV2 no domain shift	0.376	-	-	-	-	-	-	-
PIDNet no domain shift	0.345	-	-	-	-	-	-	-
PIDNet w domain shift	0.255	0.502	0.336	0.260	0.263	0.065	0.121	0.239
PIDNet Horizontal Flip	0.251	0.431	0.305	0.261	0.319	0.020	0.085	0.339
PIDNet Gaussian Blur	0.214	0.412	0.290	0.208	0.280	0.018	0.040	0.247
PIDNet Elastic Transform	0.272	0.521	0.353	0.262	0.297	0.056	0.063	0.352
PIDNet HueSaturationValue	0.273	**0.527**	0.210	0.237	0.453	0.037	0.135	0.312
PIDNet ColorJitter	0.271	0.514	0.326	0.278	0.280	0.051	0.121	0.324
PIDNet Spatter	0.237	0.489	**0.376**	0.217	0.272	0.045	0.080	0.183
PIDNet HF + GB	0.280	0.523	0.369	0.251	0.387	0.039	0.103	0.292
PIDNet ET + HSV	0.274	0.423	0.273	0.296	0.420	0.025	0.176	0.308
PIDNet HF + GB + CJ	**0.283**	0.508	0.326	**0.307**	0.356	0.075	0.046	0.365
PIDNet Adversarial	0.186	0.436	0.185	0.245	0.279	0.057	0.038	0.064
PIDNet Adversarial weighted	0.226	0.278	0.168	0.137	0.441	0.050	0.144	0.366
PIDNet Image-to-Image	0.208	0.400	0.257	0.205	0.281	0.031	0.089	0.195
PIDNet Image-to-Image weighted	0.259	0.246	0.172	0.253	0.481	0.051	**0.199**	0.407
BiSeNetV2 Adversarial	0.191	0.381	0.129	0.166	0.415	0.054	0.089	0.105
BiSeNetV2 Image-to-Image	0.243	0.220	0.314	0.255	0.223	**0.162**	0.028	**0.499**
PIDNet Style Transfer	0.134	0.375	0.109	0.192	0.127	0.043	0.021	0.069
PIDNet Image-to-Image Ens. (top 2)	0.251	0.217	0.367	0.264	0.441	0.084	0.032	0.351
PIDNet Image-to-Image Ens. (top 2) OHEM	**0.283**	0.381	0.222	0.245	**0.541**	0.078	0.188	0.326

Table 2. Results.

the target domain, they may still introduce subtle artifacts or distortions that mislead the segmentation network, especially in classes predominant in the Rural validation set such as Forest and Agriculture, where the per-class IoUs remain extremely low.

4.3.4 Adversarial Domain Adaptation

Before running the adversarial training, we conducted a hyperparameter search to determine the optimal configuration that maximizes the segmentation performance, measured in terms of mean Intersection-over-Union (mIoU) on the validation set. For this purpose, we used **Weights & Biases (WandB)**, a platform for experiment tracking and hyperparameter tuning. WandB allows efficient logging, comparison, and visualization of training runs and supports automated sweeps to explore the hyperparameter space effectively.

The following shows the hyperparameters that have been tuned, with the corresponding values: Learning Rate for the segmentation model: $5.3e^{-}4$, Discriminator Learning Rate: $2.7e^{-}6$, Weight for the adversarial loss (λ_{adv}^{target}): 0.0074 and Weight for the supervised segmentation loss (λ_{seg}): 0.45

We conducted experiments using both an unweighted OHEM loss and a median-frequency weighted OHEM loss. The weighted variant yielded better performance in terms of mIoU (0.226 compared to 0.186), primarily due to the increased emphasis on underrepresented classes. However, its overall performance remained inferior when compared to that achieved through conventional image augmentation techniques.

4.4. Image-to-Image Domain Adaptation

As with adversarial training, we employed **WandB** for experiment tracking and hyperparameter optimization in the DACS setting. The best-performing configuration was identified through a hyperparameter sweep and includes the following values: Learning Rate: $3.296e^{-4}$, Threshold:0.9 [1], Momentum: 0.85.

Image-to-image domain adaptation via DACS produced more encouraging results, particularly when combined with an inverse-frequency weighted OHEM loss during training. Additionally, we explored an ensemble strategy in which seven separate DACS-trained models were fine-tuned, each intentionally restricted from transferring pixels of a specific class to the target domain. The goal of this ablation-like approach was to identify whether excluding individual classes

[1] The **threshold** controls the confidence level required for pseudo-labels to be considered reliable. Only predictions with a maximum softmax probability above this value are trusted during training on the unlabeled (target) domain.

Excluded Class	mIoU	Background	Building	Road	Water	Barren	Forest	Agriculture
Background	0.208	0.065	0.249	0.261	0.405	0.021	0.034	0.426
Building	0.220	0.189	0.123	0.265	0.458	0.034	0.049	0.422
Road	0.218	0.190	0.262	0.258	0.315	0.036	0.091	0.378
Water	0.192	0.155	0.172	0.212	0.307	0.047	0.115	0.335
Barren	**0.252**	0.285	0.164	0.254	0.477	0.040	0.158	0.384
Forest	0.220	0.197	0.275	0.203	0.345	0.040	0.104	0.372
Agriculture	**0.253**	0.401	0.215	0.193	0.460	0.090	0.167	0.249
Ensemble (Barren + Agriculture)	**0.283**	0.381	0.222	0.245	**0.541**	0.078	0.188	0.326

Table 3. Per-class IoU and mean IoU (mIoU) for DACS models trained while excluding one class at a time. The final ensemble combines the two top-performing models (Barren and Agriculture excluded) by averaging their output logits.

could mitigate negative transfer effects and improve adaptation performance.

As shown in Table 3, the highest mIoU values were obtained by the models excluding the *Barren* (0.252) and *Agriculture* (0.253) classes. Interestingly, while excluding a class naturally led to a performance drop for that specific category (e.g., IoU of *Agriculture* drops to 0.249 when excluded), the overall model generalization on other classes improved—particularly on challenging domains such as *Water* and *Background*.

These insights suggest that excluding certain classes during adaptation may reduce domain shift interference, likely due to their high visual variability or semantic ambiguity across domains. Based on these findings, we selected the two top-performing models—those that omitted the *Barren* and *Agriculture* classes—and averaged their output logits during inference.

This **ensemble approach** achieved a mean IoU of **0.283**, comparable to our best-performing image augmentation configuration, suggesting that strategic model diversity—especially through selective class exclusion—can rival traditional domain adaptation strategies.

4.5. Additional BiSeNetV2 experiments

We also experimented with both adversarial and image-to-image domain adaptation techniques using BiSeNetV2 in place of PIDNet. As observed with PIDNet, the image-to-image approach yielded the best results among the tested methods. However, BiSeNetV2 demonstrated slightly lower performance compared to PIDNet when applying the same domain adaptation strategies.

5. Conclusion

In this work, we explored a range of domain adaptation strategies for real-time semantic segmentation in the presence of domain shift. Our experiments, conducted using the PIDNet and BiSeNetV2 architectures on the LoveDA dataset, demonstrate that simple image-level augmentations—especially combinations of geometric and color transformations—are highly effective in improving cross-domain generalization, yielding the best results among all tested methods.

We also evaluated more advanced domain adaptation techniques, including adversarial training and DACS (Domain Adaptation via Cross-domain Sampling), both in standard and ensemble configurations. While these approaches showed some improvement over the baseline domain shift scenario, their overall performance remained below or equal to that of the best augmentation strategies.

We hypothesize that one of the main factors limiting the effectiveness of these techniques is the strong class imbalance present in the LoveDA dataset. Rare classes, such as Barren, Forest, and Agriculture, are underrepresented in the source domain and appear in very different forms in the target domain.

Future work could address these challenges by incorporating more advanced class-balancing strategies, increasing the training duration, leveraging larger input resolutions, or experiment with a wider variety of image augmentation combinations.

References

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

[2] Taesung Park, Alexei A. Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation, 2020. 4, 6

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

[4] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learn-

ing to adapt structured output space for semantic segmentation. *CoRR*, abs/1802.10349, 2018. 4

[5] Irem Ulku and Erdem Akagündüz. A survey on deep learning-based architectures for semantic segmentation on 2d images. *Applied Artificial Intelligence*, 36(1), Feb. 2022. 2

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

[7] Jiacong Xu, Zixiang Xiong, and Shankar P. Bhattacharyya. Pidnet: A real-time semantic segmentation network inspired by pid controllers, 2023. 1, 2, 3

[8] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation, 2018. 1, 2

[9] 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. 1