Real-time Domain Adaptation in Semantic Segmentation

Attrovio Mario, Ghisolfo Giorgia, Russo Michele

January 30, 2025

1 Abstract

Semantic segmentation is a critical task in computer vision, enabling pixel-wise classification of images. However, the performance of segmentation models often degrades when applied to data from different domains, a challenge known as domain shift. This report explores real-time semantic segmentation in the context of domain adaptation using PIDNet as the backbone. We investigate the performance drop caused by domain shift between urban and rural datasets and evaluate mitigation strategies, including data augmentation and advanced domain adaptation techniques like adversarial training and image-to-image translation (DACS)[1]. Experimental results on the LoveDA dataset [2] demonstrate that these methods reduce the impact of domain shift while maintaining real-time inference capabilities, achieving a balanced trade-off between accuracy and computational efficiency. The code can be found on our project website: `https://github.com/MichelePoli/AMLProject`.

2 Introduction

Semantic segmentation is a foundational task in computer vision, where each pixel in an image is assigned a label corresponding to a predefined class. It plays a vital role in applications such as autonomous driving, medical imaging, and remote sensing. Recent advancements in deep learning have yielded high-performing models, but these often struggle with domain shift—a phenomenon where a model trained on a source domain (e.g., urban images) performs poorly on a target domain (e.g., rural images) [2]. Addressing this challenge is crucial for real-world deployments where annotated data for all target domains is scarce or unavailable.

Domain adaptation aims to bridge this performance gap by aligning the source and target domains without requiring extensive labeled data from the target domain. While several methods exist, real-time semantic segmentation introduces additional constraints, such as maintaining high inference speed and low computational cost. PIDNet [3], a real-time segmentation network inspired by Proportional-Integral-Derivative (PID) controllers, serves as the backbone for our study due to its efficiency and accuracy in real-time tasks.

This report focuses on evaluating and improving the performance of PIDNet for domain-adaptive semantic segmentation using the LoveDA dataset [2] (Figure 1). We first quantify the performance degradation caused by domain shift. Next, we implement data augmentation techniques and two domain adaptation approaches—adversarial training and image-to-image translation (DACS)[1] —to mitigate this issue. Our findings highlight the potential of these methods to enhance generalization while preserving the real-time capabilities of the model.

Figure 1: Satellite imagery: (Top) Rural images, (Bottom) urban images

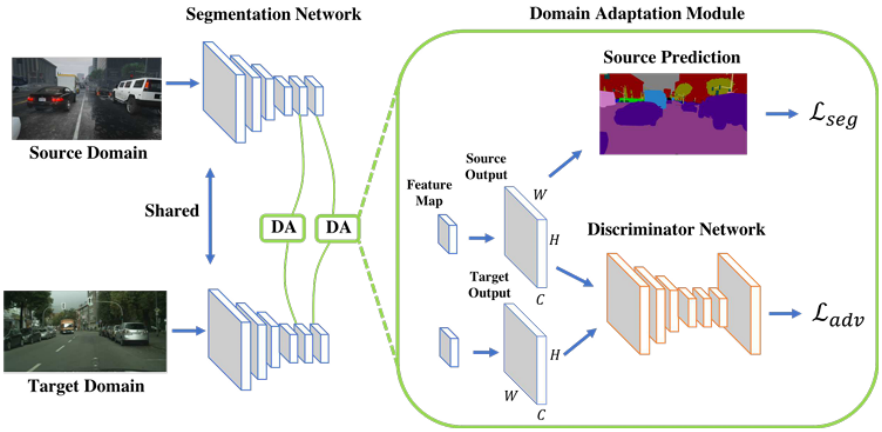

Figure 2: Algorithmic overview. Given images with the size H by W in source and target domains, we pass them through the segmentation network to obtain output predictions. For source predictions with C categories, a segmentation loss is computed based on the source ground truth. To make target predictions closer to the source ones, we utilize a discriminator to distinguish whether the input is from the source or target domain. Then an adversarial loss is calculated on the target prediction and is back-propagated to the segmentation network. We call this process as one adaptation module, and we illustrate our proposed multi-level adversarial learning by adopting two adaptation modules at two different levels here.

3 Related work

Semantic segmentation has evolved significantly with deep learning.

Foundational architectures Semantic segmentation's deep learning era began with the introduction of Fully Convolutional Networks (FCNs) [4]. By replacing fully connected layers with convolutional ones, FCNs enabled end-to-end segmentation, capable of handling images of arbitrary sizes and providing pixel-wise predictions. This innovation laid the groundwork for modern segmentation models. Further advancements were introduced with DeepLabV2 [5], which leveraged atrous (or dilated) convolutions. Atrous convolutions expanded the receptive field without increasing the number of parameters, making it possible to aggregate multi-scale contextual information efficiently.

Real-time efficiency Real-time semantic segmentation requires optimizing the balance between accuracy and computational efficiency. BiSeNet [6] introduced lightweight backbones and parallel structures, achieving an effective speed-accuracy trade-off. This architecture focused on enhancing real-time processing capabilities for applications with strict latency requirements. PIDNet [3] further pushed the boundaries of efficiency by adopting principles from PID (Proportional-Integral-Derivative) controllers. The model effectively balanced high-, mid-, and low-level features, offering improvements in both speed and segmentation quality.

Domain adaptation techniques Semantic segmentation models often suffer performance degradation when applied across different domains (e.g., urban vs. rural settings). To address this, various domain adaptation techniques have been developed two main method. Adversarial Methods [7]: these approaches train a discriminator to differentiate between source and target domain features (Figure 2). By doing so, the feature extractor is encouraged to generate domain-invariant representations, improving generalization.

Image-to-Image translation Methods like DACS [1] blend domains by leveraging mixed sampling techniques. This strategy generates pseudo-labeled target domain images, enhancing the model's ability to generalize across domains.

4 Methods

4.1 Baseline model development

4.1.1 2a Classic segmentation architecture

We implemented DeepLabV2 [5] with ResNet-101 backbone, pre-trained on ImageNet, using the LoveDA-urban dataset for training. This architecture leverages atrous spatial pyramid pooling to capture multi-scale contextual information through dilated convolutions. While effective for dense prediction tasks, its computational complexity makes it suboptimal for latency-sensitive applications. The model was initialized with ImageNet weights and trained and evaluated on urban scenes.

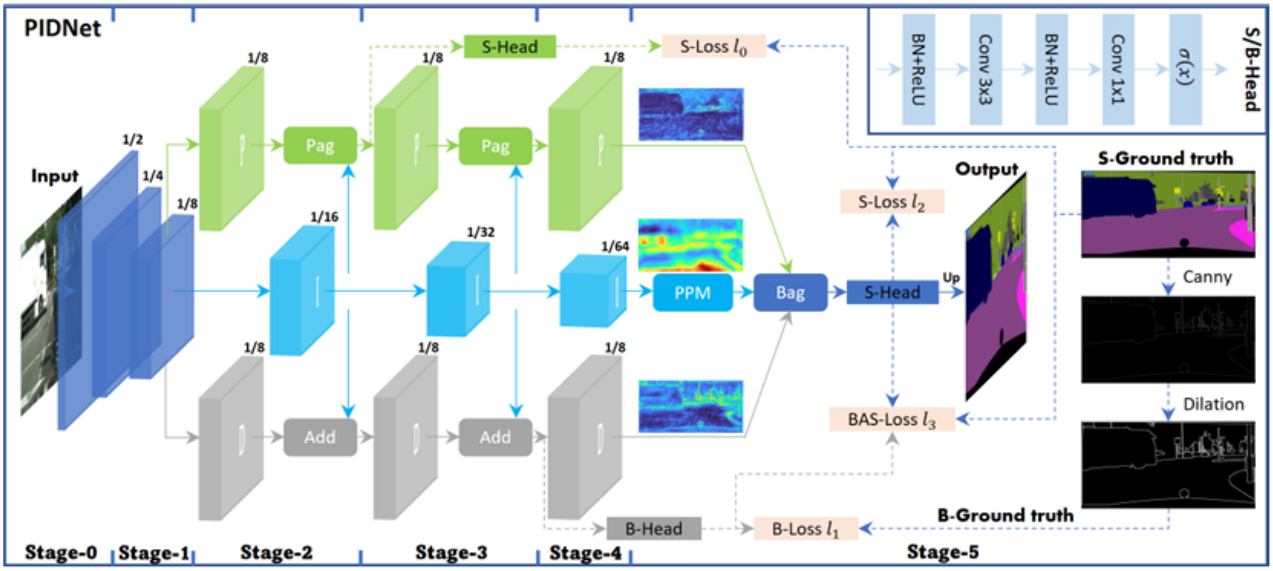

Figure 3: An overview of the basic architecture of our proposed Proportional-Integral-Derivative Network (PIDNet)

4.1.2 2b Efficient segmentation framework

To establish real-time capability baselines, we employed PIDNet-S - a streamlined architecture that mimics proportional-integral-derivative control mechanisms through three parallel branches. These branches explicitly manage high-frequency details, mid-level structures, and low-level spatial relationships respectively. The model was similarly initialized with ImageNet weights and trained and evaluated on urban scenes.

4.1.3 PIDNet architecture details

The core PIDNet architecture [3] consists of three branches (Figure 3)

- **Proportional (P) branch**: Processes full-resolution inputs using shallow layers to preserve spatial precision. Contains edge-aware convolutions for sharp boundary detection.
- **Integral (I) branch**: Leverages deep layers with dilated convolutions and spatial pooling to capture multi-scale context. Integrates a Semantic Guidance Module (SGM) to filter low-level noise.
- **Derivative (D) branch**: Computes high-frequency feature discrepancies between adjacent stages using depthwise separable convolutions. Acts as a boundary corrector by amplifying transitional regions.

The branches are fused through a *Three-Way Attention Fusion (TWAF)* module that dynamically combines features using spatial and channel attention weights.

4.2 3a Domain adaptation

We investigated cross-domain generalization by evaluating our urban-trained PIDNet-S on the LoveDA-rural dataset without fine-tuning. This experimental design isolates the domain shift problem between man-made urban environments (characterized by geometric regularity and dense infrastructure) and rural landscapes (featuring organic shapes, sparse structures, and varied terrain). (Figure 4)

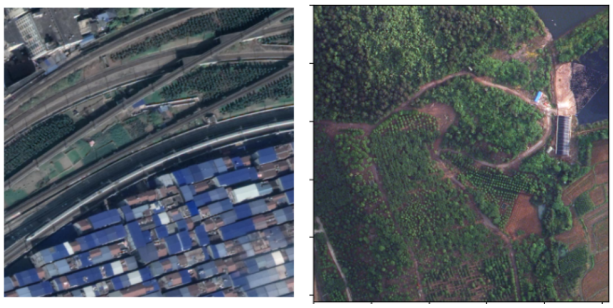

Figure 4: Example of urban image (left) and rural image (right)

4.2.1 Domain adaptation challenges

The performance degradation observed could be caused by:

- 1) **Object scale variance**: Urban structures maintain relatively consistent scales, while rural elements exhibit greater size variability (trees, water bodies)
- 2) **Contextual dependencies**: Urban scene semantics rely on positional relationships (e.g., roads between buildings), whereas rural contexts depend more on texture and color patterns
- 3) **Surface color distribution**: Artificial materials dominate urban areas (concrete, glass) (Figure 4), contrasting with natural materials (soil, vegetation) prevalent in rural regions.

This analysis motivates subsequent domain adaptation strategies to bridge the feature distribution gap between structurally distinct environments.

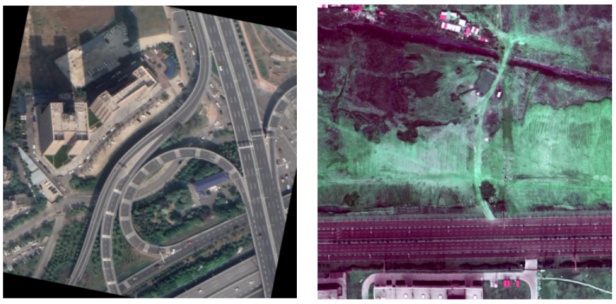

Figure 5: The left image is modified by augmentation A1, while the right image is modified by augmentation A2.

4.3 3b Data augmentation

Data augmentation was used to improve the model's generalization capability by increasing the diversity of the training data. Two augmentation strategies were applied during training with a probability of 0.5: (Figure 5)

- **A1**: Geometric transforms (horizontal/vertical flips, 30° rotation)
- **A2**: Photometric transforms (ColorJitter, GaussianBlur)

Photometric augmentations were more effective because they help the model become invariant to changes in lighting and color, which are common differences between the urban and rural domains.

4.4 Unsupervised domain adaptation

This method aligns domains through adversarial training on segmentation outputs [7], comprising three phases:

Phase 1: Source domain training The segmentation network G processes source images I_s to produce softmax probability maps:

$$P_s = softmax(G(I_s)) \in R^{C \times W \times H} \quad (1)$$

where C denotes semantic classes. Training uses standard cross-entropy loss with source labels Y_s.

Phase 2: Adversarial alignment The segmentation network now receives gradients to confuse D by:

- Maximizing discriminator uncertainty on P_t (domain label 1)
- Enforcing similar output distributions $P_s \approx P_t$

Phase 3: Discriminator training A fully-convolutional discriminator D learns to distinguish:

- Source softmax outputs P_s (domain label 1)
- Target softmax outputs $P_t = softmax(G(I_t))$ (domain label 0)

This single-stage approach achieves domain invariance by directly matching the structured output distributions rather than intermediate features.

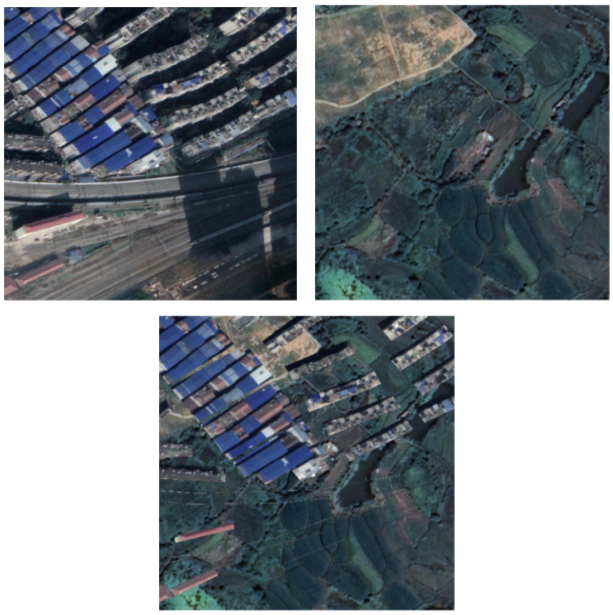

Figure 6: (top left) urban sample, (top right) rural sample, (bottom) hybrid sample

4.4.1 4b DACS

Domain Adaptation (Figure 7) via Cross-domain Mixed Sampling (DACS) [1] addresses unsupervised domain adaptation (UDA) by blending data from the source and target domains through class-level mixing. The main aspects of DACS are:

- **Augmented training samples:** DACS creates hybrid samples by combining (Figure 6):
  - A source domain image, which comes with ground-truth labels.
  - A target domain image, which has pseudo-labels generated by the model.

  During this process, a subset of classes is selected from the source image using its semantic map, and the corresponding pixels are pasted onto the target image.
- **Composite label generation:** The labels for the mixed image are produced by merging:
  - Ground-truth labels from the source image.
  - Pseudo-labels predicted for the target image.

  This ensures that the model is trained on hybrid data containing both reliable source annotations and target pseudo-labels, encouraging feature invariance across domains.
- **Training loss:** The overall loss combines:
  - Supervised learning on the source domain data.
  - Consistency regularization on the mixed-domain samples.

  Importantly, this does not require any annotations from the target domain.

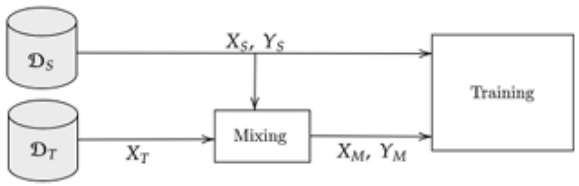

Figure 7: Diagram showing DACS. The images X_S and X_T are mixed together, using Y_S for the labels of X_S, instead of a pre dicted semantic map to determine the binary mask. The segmenta tion network is then trained on both batches of augmented images and images from the source dataset.

4.4.2 Extensions

We explored three alternative real-time semantic segmentation architectures:

- **BiSeNetV1** [6]: Employs a dual-path design with a *spatial path* (high-resolution stream for fine details) and a *context path* (fast downsampling with global average pooling for long-range dependencies). The features are fused using a specialized Feature Fusion Module (FFM).
- **LinkNet** [8]: Uses a lightweight encoder-decoder structure with residual skip connections. The encoder reduces spatial dimensions through cascaded convolutional blocks, while the decoder employs transposed convolutions for upsampling, with direct additive links between corresponding encoder-decoder layers.
- **STDC n** [9]: Features a Short-Term Dense Concatenate backbone that progressively aggregates multi-scale features through dense connections in early stages, followed by a Context Path with Attention Refinement Modules (ARMs) to enhance contextual awareness.

These models emphasize distinct architectural strategies: BiSeNetV1's parallel multi-resolution processing, LinkNet's symmetrical skip connections, and STDC-Net's dense feature aggregation.

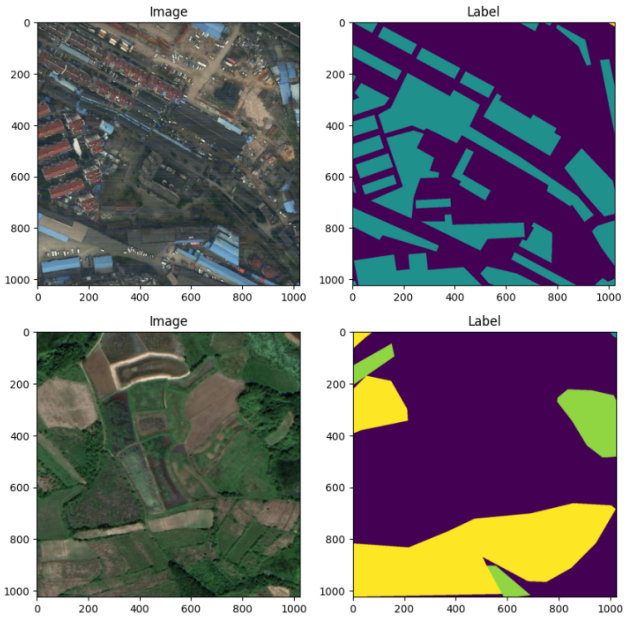

Figure 8: Class distribution similarities between urban (top) and rural (bottom) domains in LoveDA: the background class dominates both domains.

5 Experimental results

5.1 LoveDA dataset

The LoveDA dataset [2] is a high-resolution (0.3 m) remote sensing dataset designed for domain-adaptive semantic segmentation in urban and rural environments. It contains 5,987 images across three Chinese cities (Nanjing, Changzhou, Wuhan), annotated with seven classes: *background*, *building*, *road*, *water*, *barren*, *forest*, and *agriculture*. The dataset is explicitly divided into urban (2,522 images) and rural (3,465 images) domains to study domain shift challenges.

Key characteristics of LoveDA include:

- **Multi-scale objects**: Urban scenes feature densely packed buildings and structured roads, while rural areas contain scattered agricultural plots and irregular water bodies. Buildings in urban regions exhibit larger size variance compared to rural regions (Figure 2).
- **Complex backgrounds**: The *background* class dominates both domains (Figure 8), encompassing diverse elements like vehicles and undeveloped land, which introduces high intra-class variance.
- **Domain shift**: Urban and rural domains exhibit divergent class distributions (e.g., urban has 32% buildings vs. rural's 8%) and spectral properties (lower variance in rural areas due to homogeneous landscapes)(Figure 9).

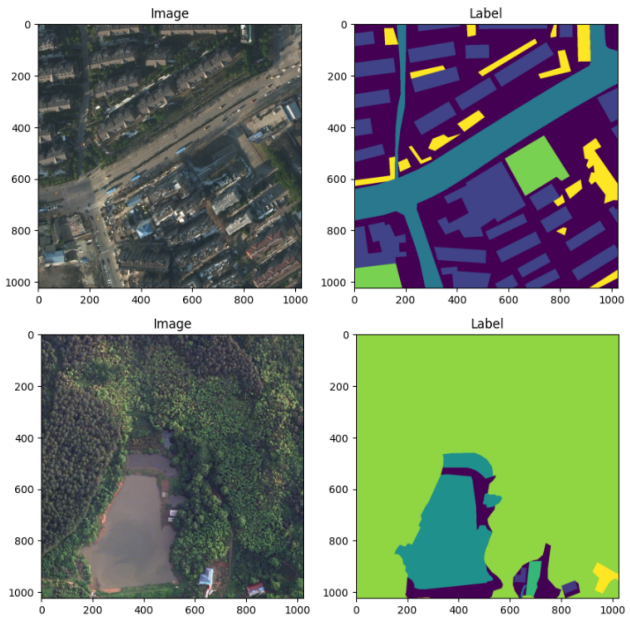

Figure 9: Class distribution diversity between urban (top) and rural (bottom) domains in LoveDA.

5.2 Performance on source domain

Table 1 shows the performance of DeepLabV2 and PIDNet-S on the LoveDA-urban dataset (source domain). PIDNet-S, designed for real-time performance, achieved a significantly higher mIoU than DeepLabV2 while also providing latency, FLOPs, and parameter count.

Model	mIoU (%)	Latency (ms)	FLOPs	Params
DeepLabV2	33.41	13.42	11.535G	62.231M
PIDNet-S	48.67	9.23	6.35G	7.72M

Table 1: Performance comparison of different hyperparameter tuning.

5.3 Domain shift evaluation

Table 4 quantifies the domain shift from LoveDA-urban to LoveDA-rural. The baseline PIDNet-S model, trained on urban data, experienced a significant performance drop when tested on rural data. Data augmentations (A1 and A2) improved the mIoU, with A2 (photometric transforms) being the most effective. Adversarial training and DACS further mitigated the domain shift, achieving similar performance.

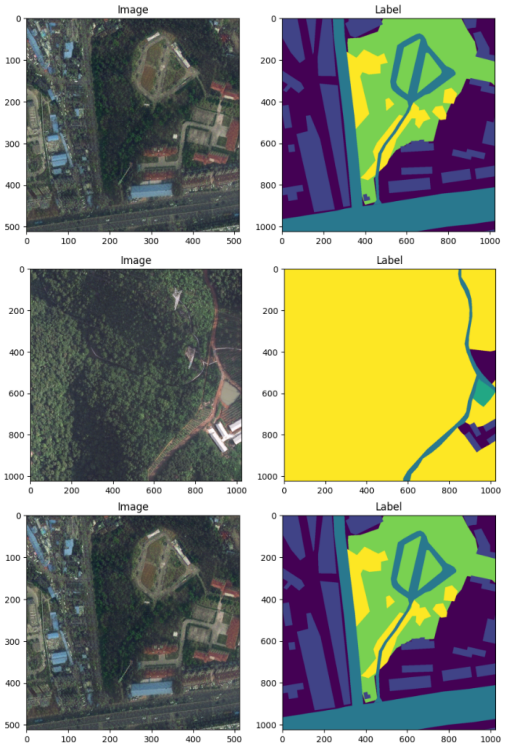

Figure 10: urban image

Model	Road	Building	Water	Barren	Forest	Agric.	mIoU
PIDNet	16.34	23.99	35.46	3.12	8.83	31.82	23.98
+ A1	29.33	40.77	36.71	9.17	9.53	33.13	29.91
+ A2	31.41	38.37	31.33	10.26	15.10	37.51	30.80
+ A1 + A2	27.92	32.97	33.98	10.50	10.69	37.45	29.35
+ Adv.	0.36	13.41	32.72	8.28	49.73	12.26	30.59
+ DACS	31.32	36.60	42.70	4.49	2.70	40.60	30.63

Table 2: Performance comparison of different version of aumented dataset.

5.4 Extensions

Table 3 presents the results of the extensions on the LoveDA-rural dataset. Style transfer preprocessing improved the mIoU slightly compared to the baseline but was less effective than data augmentation or domain adaptation techniques. BiSeNetV1 and LinkNet showed competitive performance, with BiSeNetV1 outperforming PIDNet-S on the target domain. Notably, BiSeNet STDC achieved the highest mIoU of 34.10%, surpassing both BiSeNetV1 and LinkNet. This improvement is likely due to its lightweight yet effective spatial and channel attention mechanisms, which enhance feature extraction while maintaining efficiency. The superior performance of BiSeNet STDC suggests that optimizing network architecture can yield significant gains in rural scene segmentation.

Model	mIoU (%)
PIDNet + Style Transfer (Figure 10)	25.46
BiSeNetV1	32.21
LinkNet	30.43
BiSeNet Stdc	34.10

Table 3: Performance comparison of different models on rural images.

5.4.1 Hyperparameter tuning Rresults and analysis

Table 4 presents the results of hyperparameter tuning experiments conducted on the LoveDA-rural dataset. The default configuration (batch size = 16, learning rate = 10^{-4}) achieved the highest validation mIoU of 30.63%. Smaller batch sizes (2, 8) resulted in worse performance, likely due to instability in training and poor gradient estimation. Increasing the learning rate to 10^{-3} degraded performance, probably because of convergence issues. The adaptive lambda adjustment also did not improve results, showing limited generalization benefits.

Configuration	Best Validation mIoU (%)
Batch Size = 16, LR = 10^{-4} (Default)	30.63
Batch Size = 2, LR = 10^{-4}	15.00
Batch Size = 8, LR = 10^{-4}	16.59
Batch Size = 16, LR = 10^{-3}	19.09
Batch Size = 16, Adaptive Lambda	18.26

Table 4: Performance comparison of different models on urban images.

Smaller batch sizes (2, 8) resulted in significantly lower performance, likely due to higher gradient noise and instability in training, leading to suboptimal convergence. A higher learning rate (10^{-3}) caused a decrease in mIoU, probably due to overshooting during optimization, preventing the model from reaching a good local minimum. The introduction of an adaptive lambda strategy, while theoretically beneficial for balancing losses dynamically, did not improve the results, possibly due to inadequate adaptation to the dataset characteristics or suboptimal parameter tuning.

5.4.2 Loss function comparison

Table 5 presents the results of different loss functions evaluated on the LoveDA-rural dataset. The default configuration utilized OhemCrossEntropy with the DACS loss formulation, achieving the highest validation mIoU of 30.63%. Boundary loss resulted in the worst performance, with a validation mIoU of 8.50% and a considerably higher loss value of 11.1707. This high loss value points to issues with gradient propagation, potentially caused by boundary loss not being as effective at focusing on difficult regions. The model likely failed to effectively learn from the more complex areas of the image, resulting in low accuracy and unstable training. CrossEntropy loss improved upon boundary loss, with a validation mIoU of 24.59% and a loss value of 0.8962. While the loss value is lower than that of boundary loss, the performance still lags behind the default OhemCrossEntropy + DACS setup. The absence of Online Hard Example Mining (Ohem),

which helps prevent bias towards background pixels, could explain this drop in mIoU. The model might have focused on easier examples and ignored the more difficult ones, leading to a suboptimal performance. Dice loss performed slightly worse than CrossEntropy loss, achieving a validation mIoU of 22.66% with a corresponding loss value of 0.6475. Dice loss is typically used to address class imbalance, but in this case, it did not outperform the default setup. The lack of an explicit mechanism to focus on difficult or hard-to-learn pixels, which Ohem provides, might explain why Dice loss was not as effective in this context. While the lower loss value is a positive indicator, it does not directly translate to improved mIoU.

Loss Function	Validation mIoU (%)	Loss Value
OhemCrossEntropy + DACS (Default)	30.63	0.7993
Boundary Loss	8.50	11.1707
CrossEntropy Loss	24.59	0.8962
Dice Loss	22.66	0.6475

Table 5: Performance comparison of different loss functions.

6 Conclusion

This study highlights the effectiveness of PIDNet for real-time semantic segmentation, achieving a strong performance on the LoveDA-urban dataset (9.23ms latency, 48.67% mIoU). However, the impact of domain shift, as observed on LoveDA-rural (23.98% mIoU), underscores the challenges of applying segmentation models across divergent environments. Among the evaluated mitigation strategies, photometric data augmentation (A2) showed the most consistent improvement (+6.82% mIoU), while adversarial training and DACS offered moderate yet comparable gains (+6.61% and +6.65% mIoU, respectively). Despite these advancements, the trade-off between accuracy and computational efficiency remains a key limitation, particularly in edge computing scenarios.

Our experiments demonstrate that while PIDNet excels in maintaining real-time capabilities, alternative architectures like BiSeNetV1 can surpass it in domain-adaptive accuracy, at a slight cost to speed. These findings suggest that a hybrid approach, integrating the efficiency of PIDNet with enhanced feature representation techniques from other models, could be a promising direction. Future research could focus on:

1. **Hybrid Adaptation Strategies**: Combining DACS with adversarial training or exploring ensemble techniques to synergize their strengths.

2. **Optimized Style Transfer Pipelines**: Refining preprocessing steps to reduce the spectral and textural differences between domains.

3. **Domain-Specific Fine-Tuning**: Introducing lightweight domain-specific adaptations during deployment to address localized variations.

4. **Longer Training Regimes**: Exploring optimized training schedules that balance performance improvements with real-time constraints.

Ultimately, this work provides a foundation for advancing domain-adaptive real-time semantic segmentation, offering practical insights for deploying robust models in dynamic, real-world environments where domain variability is unavoidable.

References

[1] W. Tranheden, V. Olsson, J. Pinto, and L. Svensson, "Dacs: Domain adaptation via cross-domain mixed sampling," *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1479–1489, 2021.

[2] J. Wang, Z. Zheng, A. Ma, X. Lu, and Y. Zhong, "Loveda: A remote sensing land-cover dataset for domain adaptive semantic segmentation," *CoRR*, vol. abs/2110.08733, 2021. [Online]. Available: https://arxiv.org/abs/2110.08733

[3] J. Xu, Z. Xiong, and S. P. Bhattacharyya, "Pidnet: A real-time semantic segmentation network inspired by pid controllers," 2023. [Online]. Available: https://arxiv.org/abs/2206.02066

[4] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 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, 2018.

[6] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Bisenet: Bilateral segmentation network for real-time semantic segmentation," *Proceedings of the European conference on computer vision (ECCV)*, pp. 325–341, 2018.

[7] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker, "Learning to adapt structured output space for semantic segmentation," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7472–7481, 2018.

[8] A. Chaurasia and E. Culurciello, "Linknet: Exploiting encoder representations for efficient semantic segmentation," *2017 IEEE Visual Communications and Image Processing (VCIP)*, pp. 1–4, 2017.

[9] M. Fan, S. Lai, J. Huang, X. Wei, Z. Chai, J. Luo, and X. Wei, "Rethinking bisenet for real-time semantic segmentation," *CoRR*, vol. abs/2104.13188, 2021. [Online]. Available: https://arxiv.org/abs/2104.13188