Learning To See In The Dark and Beyond

Anirudha Ramesh

CMU-RI-TR-23-43

August 1, 2023

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Jeff Schneider, *chair*
Christoph Mertz
Srinivasa Narasimhan
Dinesh Reddy

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics.

To my parents, Usha and Ramesh.

Abstract

Robotic Perception in diverse domains such as low-light scenarios remains a challenge, even after the incorporation of new sensing modalities like thermal imaging and specialized night-vision sensors. This is primarily due to the difficulty in obtaining labeled data in these new domains across multiple tasks.

In this thesis, we provide a pathway for designing robots that can operate in new visual domains, across different tasks of varying difficulty in labeling. While our work is directed towards operating artificial agents passively at night, it can be extended to other new environments as well. We demonstrate our approach in the critically important and representative tasks of object detection and semantic segmentation, where the former corresponds to a task where label generation is often feasible, and the latter to a task where it isn't.

First, we extend the operating range of an object-detection system to enable on-robot low-light operations. We do so by employing a high-sensitivity camera and train an object detection model on it with the aid of labeled in-domain data, and deploy it for on-robot operations, thus extending the operating range of the system to function 24/7.

For the more challenging setting, where generating large quantities of new labels can be prohibitively expensive, we propose a novel label-efficient, and effective Domain Adaptation framework, Almost Unsupervised Domain Adaptation (`AUDA`), that critically accounts for biases learned by the original model in the source domain, and show it on semantic segmentation. While existing Domain Adaptation techniques, promise to leverage labels from well-lit RGB image datasets, they fail to consider the characteristics of the source domain itself, such as noise patterns, texture, glare etc. We holistically account for this by proposing Source Preparation (`SP`), a method to mitigate source domain biases. Our semi-supervised framework for realistic robotic scenarios, `AUDA`, employs Source Preparation (`SP`), Unsupervised Domain Adaptation (`UDA`), and Supervised Alignment (`SA`) from limited labeled data ($\sim$10s of images) to train models in new domains with limited labeled target data.

Our method outperforms state-of-the-art across a range of visual domains, with improvements of up $\sim$+40% in mIoU in unsupervised, and $\sim$+30% in mIoU in semi-supervised scenarios, in addition to a marked increase in robustness to realistic shifts that can occur to the target domain.

Finally, we introduce the first 'intensified' dataset captured at night time comprising images from an intensifier camera, and a high-sensitivity camera to facilitate low-light robotic operations.

Acknowledgments

There have been countless people who've supported me throughout my journey, allowing me to stand atop their shoulders. I'd like to take this opportunity to thank a small fraction of this innumerable whole.

First, I'd like to thank my advisors, Jeff Schneider, and Christoph Mertz, for their incredible support over the last two years. Their guiding hands have instilled in me the necessity to consistently look at the bigger picture, and their insight has been invaluable in my efforts to make a meaningful contribution to this domain, and to my growth as a researcher. Above all else, I'm grateful for the kindness they've always shown me.

Further, I would like to thank Srinivasa Narasimhan, and Dinesh Reddy for their role in my thesis committee. Their feedback during the process of assembling and presenting this thesis has considerably helped in strengthening this work.

This research was not done in isolation, but rather in collaboration with a large team, both at CMU and National Robotics Engineering Center (NREC). The efforts from the folk at NREC helped provide a platform over which we could turn some of our research efforts into deployed systems on real robots. A great number of wonderful people at NREC have gone beyond the clock to support my work, and I'd be remiss to not extend a special thanks to Jon, Wei, Zach, Vamsi, Dan, Rohan, Tomasz, Jarek, Krissy, Prasanna, Christof, Luis, Dave, and Felix. My gratitude extends well beyond the aforementioned to everyone else I've had the pleasure of working with. I'd also like to thank my lab-mates, Anurag, Phillip, and Talha, for all our discussions and collaborative efforts towards solving research problems addressed in this thesis, and beyond.

My friends, both at CMU and beyond, have helped make every day thoroughly enjoyable, and I'm grateful for their part in my life.

Finally, I want to thank my family - Keshav, Usha, Ramesh, Chellappa, Rama, Ramaswamy, and Sushila - for their love and unconditional support throughout the years, without whom none of this would be possible.

Funding

This material is based upon work supported by the U.S. Army Research Office and the U.S. Army Futures Command under Contract No. W911NF-20-D-0002. The content of the information does not necessarily reflect the position or the policy of the government and no official endorsement should be inferred.

Contents

1 Introduction **1**

2 Related Work **7**
2.1 Domain Adaptation with Limited Supervision 7
2.2 Unsupervised Domain Adaptation and Domain Generalization 9
2.3 Robot Vision in Low Light 10
2.4 Real-time on-robot Object Detection Systems 10

3 Extending the Operating Range of Object Detection Systems to Enable On-Robot Low-Light Operations **13**
3.1 System Setup and Testing Environment 14
3.2 Our Solution 19
3.3 Results 20
3.3.1 Quantitative 21
3.3.2 Qualitative 25

4 Training Models on New Domains Effectively in a Label Efficient Manner **27**
4.1 Methodology 27
4.1.1 Problem Setup 27
4.1.2 Overview of Proposed `AUDA` Framework 28
4.1.3 Source Preparation 29
4.1.4 Unsupervised Domain Adaptation 31
4.1.5 Supervised Alignment 32
4.2 Experiments and Results 32
4.2.1 Effect of Source Preparation 33
4.2.2 AUDA for Effective Label Efficient Domain Adaptation across large Domain Gaps 36
4.2.3 SP with an Alternate Domain Adaptation Approach 39
4.2.4 Towards Appropriately Stacking Different SP Schemes 40
4.2.5 Analyzing Qualitative Results 41

4.2.6 Additional Details on Methods, Experimental Set-up, and Compute Use 48

5 PittIntensified : Intensified Images for Vision in Low-Light Scenarios 51
5.1 Collection and Labeling Set Up 52
5.2 Qualitative Examples 53
5.3 Quantitative Analysis 57

6 Conclusions 59

Bibliography 61

When this dissertation is viewed as a PDF, the page header is a link to this Table of Contents.

List of Figures

1.1 Target domains exhibit characteristics distinct from the source domain, such as high photon noise in intensifier images and infrared reflections in thermal camera images. Similarly, source domain-specific characteristics exist, and a source model overfitting to these characteristics, such as high-frequency detail which corresponds primarily to noise in intensifier images, can hinder Domain Adaptation. To mitigate this, we propose Source Preparation as an alternative to conventional source model training. Source Preparation enhances domain adaptation by minimizing overfitting in the source domain while implicitly encouraging learning domain invariant features, i.e. features more relevant to the target domain. 4

3.1 Our Robotic System with high-sensitivity cameras, Canon ME20F-SH (pictured bottom-right), for night operations mounted at the top. . . 15

3.2 Images of our robot in our operating environment. The first corresponds to an image taken prior to start of operations, the second at time of operations illuminated with a flash-light to illustrate the surrounding darkness, and the third captured with another high-sensitivity camera during operation. 16

3.3 Illustrative examples of instances of each class. Note that person class includes mannequins. 18

3.4 We collect and label 6000 night-time images captured with a High-Sensitivity RGB camera, in addition to using roughly 5000 day-time regular camera images to train `YoloV5s`, a real-time object detection model, to obtain our low-light night-ops capable object detection model. 20

3.5 `P`, `R`, and `PR` curves of our model on our test set. Plot associated with PR curve contains class-wise `AP50` in labels. 23

3.6 `AP50` - person vs Frames Per Second (FPS) for different object-detection algorithms on our set-up. 24

3.7 Some qualitative outputs of our object-detection model from our demo runs. Our results consistently detect all relevant objects with high quality bounding boxes. "grizzly" is an alias to class "friendly". . . . 25
3.8 Select 'failure-cases' of our object detection model. The first two images correspond to over-counting leading to spurious detections, while the last shows a frame where an object is misclassified when it appeared towards the edge of the frame. 26

4.1 This figure illustrates our proposed framework, AUDA, for realistic robotic scenarios where some labeled target samples can be obtained. In contrast to traditional UDA, our approach includes Source Preparation (SP) to create a more 'adaptable' model for UDA, thus enhancing it, while still being fully unsupervised. This is then following by Supervised Alignment (SA) to leverage the limited labeled data available in the target domain. 28
4.2 MixStyle is used for Source Preparation (SP) by making the highlighted modification in SegFormer's encoder [58]. These modifications are used only for SP, and not UDA or SA. 30
4.3 Examples from DarkZurich augmented with rain, snow, fog, increased motion blur, and cartoonification. 35
4.4 Stage-wise qualitative results, with the fourth row corresponding to AUDA = UDA + SP + SA. Each step of AUDA improves our final outputs. The inclusion of SP with UDA massively reducing both false positives and false negatives as compared to predictions with just UDA. This is particularly true for 'vehicle' class predictions. Further inclusion of SA refines our labels, as can be seen in the case of 'person' class predictions. 43
4.5 Stage-wise qualitative results, with the fourth row corresponding to AUDA = UDA + SP + SA. Each step of AUDA improves our final outputs. The inclusion of SP with UDA massively reducing both false positives and false negatives as compared to predictions with just UDA. Further inclusion of SA helps sweep-up objects formerly missed out, as we cam see with the car in the first example. 44
4.6 Stage-wise qualitative results, with the fourth row corresponding to AUDA = UDA + SP + SA. Each step of AUDA improves our final outputs. The inclusion of SP with UDA massively reducing both false positives and false negatives as compared to predictions with just UDA. Further inclusion of SA helps sweep-up objects formerly missed out, while also refining predictions of both 'people' and 'vehicle' classes, as well as reducing erroneous widely off-the-mark false positive predictions of 'vehicle'. 45

4.7 We can see significant improvement in these representative examples from CS → DZ after AUDA. Our predictions are generally far smoother, performance particularly in background classes is far improved, and our model looks to be capable of operating in such night time environments. 46
4.8 We can see significant improvement in represenative examples from CS → MFNT after SP+UDA, with our output, even prior to SA, being very usable in thermal domains. While background prediction is significantly improved, it is not reflected in our quantitative results owing to lack of ground truth labels corresponding to these classes. Additionally foreground objects, once missed or crudely segmented, are captured much more effectively. 47

5.1 Representation of how these scenes appear with a regular camera. . . 53
5.2 Representative samples from PittIntensified, with object detection (bbox) and instance segmentation labels. 54
5.3 Representative examples from PittIntensified, with corresponding segmentation labels, where blue is used to represent the 'vehicle' class, red to represent the 'people' class, *white* to represent the background class, and gray to represent the ignore label, which corresponds to a fixed area in the image blocked by the intensifier module. 55
5.4 Representative examples from PittIntensified, with corresponding segmentation labels, where blue is used to represent the 'vehicle' class, red to represent the 'people' class, *white* to represent the background class, and gray to represent the ignore label, which corresponds to a fixed area in the image blocked by the intensifier module. 56
5.5 Number of annotated pixels in each labeled class in PittIntensified. . 57
5.6 Number and distribution over sizes of annotated labels of objects with detection (bbox) and instance segmentation labels for people and vehicles. There are a total of 241 instances of the 'people' and 393 instances of the 'vehicle' class in the 393 labeled images of PittIntensified. 58

List of Tables

2.1 Summarizing and highlighting how our proposed framework, Almost Unsupervised Domain Adaptation (AUDA), differs from existing classes of Domain Adaptation methods, which are Unsupervised Domain Adaptation (UDA), Semi-Supervised Domain Adaptation (SSDA), and Few-Shot Semi-Supervised Domain Adaptation (FSSDA). It's important to note that, in addition to this, AUDA, can also work in an unsupervised setting similar to UDA methods, by not running the final Supervised Alignment step, while still enchancing UDA with SP. Here, the column corresponding to Effectiveness refers to the relative ability of the class of methods to maximize performance of our domain adapted model in realistic robotic scenarios where some labeled target domain data can often be obtained. Label-Efficiency corresponds to whether the method can operate with little ($\sim 10s$) to no labeled target samples. . . 8

3.1 Class-wise distribution of ground-truth labels in our training (11,223 images) and test set (604 images). 19

3.2 Class-wise AP50 and AP50-95 on our test-set. These results indicate that our model's performance is very strong, and is capable of night-time object detection. The last row, highlighted in bold, shows results averaged over all classes, i.e mAP. 21

3.3 AP50 for 'person' class over different sizes (following COCO definitions). With our setup 'small' is roughly equivalent to more than 40 meters from the camera. Our results indicate a strong model performance across sizes. 21

3.4 AP50 for 'person' class with OTS COCO-pretrained YOLOV5s, and our night-time YOLOV5s model on images from the high-sensitivity camera taken during day and night time in the same location. Our in-domain night-trained model outperforms its OTS counterpart in both its performance during the day and at night. 21

4.1 Comparison (mIoU) on respective validation sets after UDA from Cityscapes to DarkZurich, MFNet Thermal, PittIntensified with different Source Preparation techniques. In each case we can improve the potency of UDA with the right kind of Source Preparation. 33
4.2 Comparison (mIoU) on DarkZurich Val under various additional real-world shifts (and another style-shift) in the target domain after UDA from Cityscapes with different Source Preparation (SP) techniques. SP-stacked refers to MixStyle+mixup+Blur (all included in source model training with equal weight-age). SP performs better across all shifts, indicating increased robustness in the target model. 36
4.3 Comparison (mIoU) on respective validation sets after limited Supervised Alignment (SA) of the models with and without best performing Source Preparation (SP), and UDA from Cityscapes. Incorporating SA after both SP and UDA yields the best-performing models in the target domain, particularly when labeled target samples are scarce. The highlighted improvements correspond to improvements along the column. 36
4.4 Summarizing the contribution of each stage of AUDA, with 50 labeled target samples of SA, with their improvements (mIoU) highlighted. Results shown over respective validation sets. 37
4.5 Comparison (mIoU) on respective validation sets with the best performing SP technique for each dataset applied as a separate SP step before UDA or together with UDA. Results indicate that a separate SP generally yields superior target models. 37
4.6 Comparison (mIoU) between AUDA and PixDA. Results shown for validation sets of respective datasets. These indicate that AUDA can adapt more effectively across larger domain gaps. 38
4.7 Comparison (mIoU) to showcase label-efficiency of AUDA vs other SSDA approaches. AUDA not only performs better under label scarcity, but the degradation in performance as we approach label scarcity is also reduced. 39
4.8 Comparison (mIoU) to showcase the effect of SP on a different Domain Adaptation technique, USSS. SP shows that it can boost performance across both datasets and different levels of label scarcity. 40
4.9 Comparison (mIoU) to showcase the effect of chaining our SP schemes vs best individual SP scheme for each dataset after UDA. 41

Chapter 1

Introduction

Visual perception in diverse environments such as low light scenarios poses an extremely challenging problem. Animals are adept at perception in such situations, due to structural adaptations in their perception mechanism [3] or novel sensing mechanisms that let them sense radiant heat beyond the visible spectrum [42]. Can we bestow such capabilities to our robots by employing emerging sensing and imaging modalities like thermal and specialized night-vision sensors?

Challenges in robotics in low-light scenarios (such as Figure 1.1) can be addressed by employing such sensors and adapting models to operate on these new modalities, which comprise of new "domains" for our machine learning models. More formally, a new domain refers to input data distributions which differ significantly from the data we have seen during training.

Through years of painstaking effort we've collected, curated, and labeled massive amounts of data with regular cameras, particularly in well lit environments, allowing us to create great models for these conditions. We don't however have the same magnitude or quality in labeled data in other visual domains. This becomes particularly challenging for certain tasks where the ability to generate labels in new target domains, such as semantic segmentation, is prohibitively expensive. For other tasks like classification, and object detection, while labeling may still be expensive, it is far cheaper, and so is often feasible. To operate in new environments, we likely need to perform an array of such tasks, all differing in their labeling expense, leaving us with a very challenging problem overall.

However, all is not lost, as we are still viewing the same world, only through different lenses! Now, the key challenge becomes leveraging existing labeled data to help train models in new domains. In this thesis, we provide a pathway for designing robots that can operate in new visual domains, across different tasks of varying difficulty in labeling. While our work is directed towards operating artificial agents at night, it can be extended to other new environments as well.

Our contributions can be stated as follows,

- We extend the operating range of an object detection system to enable on-robot low-light operations. This corresponds to a scenario where obtaining labels in the target domain is difficult, but feasible.
- For the more challenging strictly label limited scenario, we propose a new label-efficient and effective framework for domain adaptation, and show its efficacy across large domain gaps on the representative task of semantic segmentation.
- We introduce a first of its kind low-light dataset comprising of images from an intensifier and a high-sensitivity camera.

To address the first challenge, we incorporate a high-sensitivity RGB camera, and use the fact that generating labels for object detection in new target domains is feasible to train and deploy an object detection system on-robot to enable low-light night-operations. Our system performs strongly at night as detailed in Chapter 3.

Obtaining lots of labeled data in new domains for tasks like semantic segmentation is however often infeasible, making the development of robotic systems with multi-modal capabilities difficult. This necessitates the effective use of data and labels from 'similar' mainstream domains, like regular RGB in well-lit scenarios, to support the training of models in new visual domains. Our efforts must therefore be focused on generalizability, but even without a lot of labeled target samples we can extract some critical signal about our target domain from its samples alone, with little to none of them being labeled. With this we can constrain our problem and focus our efforts from generalizing to all possible domains, to adapting to specific new target environments. We can do this with Domain Adaptation [2].

Domain Adaptation (`DA`) allows us to leverage similarities across domains without having access to many hard-to-obtain labels while also relying on existing labeled data available in mainstream visual domains. In many robotic scenarios (such as

off-road autonomous driving, and zone exploration at night) it is realistic to assume the availability of limited labeled data in addition to unlabelled data from the target domain.

Existing approaches to Domain Adaptation don't appropriately capture such scenarios, particularly when domain gaps. Unsupervised Domain Adaptation (UDA), [11, 38, 46] methods attempt to adapt models without utilizing any labeled target domain data, while Semi-Supervised Domain Adaptation (SSDA) [6, 7, 49, 59] methods require hundreds of labeled target samples for complex tasks like semantic segmentation, and existing approaches to Few-Shot Supervised Domain Adaptation (FSSDA) [40, 50, 64, 66] are generally designed to adapt across small domain gaps.

Moreover, while recent domain adaptation techniques [5, 15, 22, 47, 57] adapt models trained on labeled data in the source domain to perform well in a different target domain, they fail to consider the characteristics of the source domain itself, that the source model becomes biased towards. Based on this observation, we ask a key question, *can we assume that all the features learned by the model trained on the source domain be adapted to other domains?*

To address these issues, we take a holistic view of Domain Adaptation and propose a label-efficient three-stage Semi-Supervised framework called Almost Unsupervised Domain Adaptation (AUDA). Firstly, we propose Source Preparation (SP) as an alternative to conventional source model training, to improve the adaptability of source models (Figure 1.1). With SP, we test our hypothesis and attempt to mitigate biases towards source domain-specific characteristics by minimizing overfitting in the source domain while implicitly encouraging learning domain-invariant features, i.e. features which are more likely relevant to the target domain. Then, we employ Unsupervised Domain Adaptation (UDA) to exploit available unlabelled target domain images. If we are provided with absolutely no labeled samples from the target domain, we can stop here. If not, we exploit the even the few labeled target images ($\approx$20-50) available to us to perform a limited Supervised Alignment (SA) to the target domain.

As AUDA employs a far lower number of labeled samples and operates in a different label regime compared to existing SSDA approaches for semantic segmentation [6, 7], it can be applied to label-scarce domains, while still being able to adapt across larger domain gaps than FSSDA approaches [50], by exploiting unlabeled target data more effectively with SP and UDA.

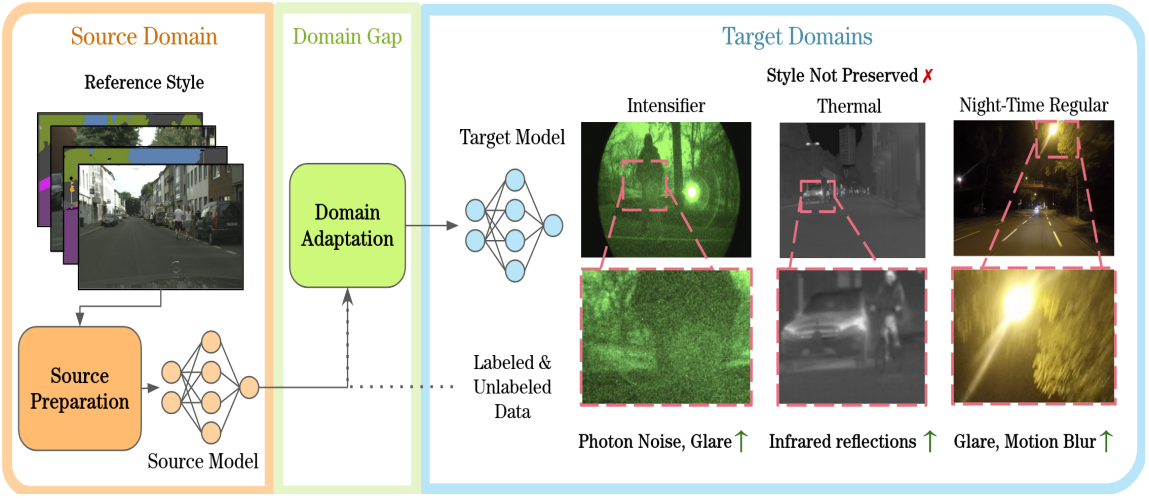

Figure 1.1: Target domains exhibit characteristics distinct from the source domain, such as high photon noise in intensifier images and infrared reflections in thermal camera images. Similarly, source domain-specific characteristics exist, and a source model overfitting to these characteristics, such as high-frequency detail which corresponds primarily to noise in intensifier images, can hinder Domain Adaptation. To mitigate this, we propose Source Preparation as an alternative to conventional source model training. Source Preparation enhances domain adaptation by minimizing overfitting in the source domain while implicitly encouraging learning domain invariant features, i.e. features more relevant to the target domain.

To rigorously evaluate `AUDA` and understand the implications of `SP`, we introduce PittIntensified, a first-of-its-kind dataset comprising temporally aligned image pairs captured from a high-sensitivity camera and an intensifier camera, with semantic and instance labels, in various low-light scenarios (Chapter 5). While thermal sensors can be used even when it's completely dark, low-light scenarios often have some light to be exploited which regular RGB cameras cannot sufficiently do. We address this gap in existing public datasets for low-light vision tasks and provide paired High-Sensitivity RGB and Intensifier images to enable Domain Adaptation to images captured by an intensifier camera.

Our results show that `AUDA` and critically, `SP` improves model performance in various target domains (See Section 4.2.1), while also enhancing robustness to realistic shifts within the target domain (Section 4.2.1.1). Our experiments also confirm the efficacy of `AUDA` for label-efficient `DA` across challenging domains, with access to as few as 20-50 labeled target samples (Section 4.2.1.2, 4.2.2). We also provide design

principles for selecting or developing SP methods for new target domains.

Our work thus provides a pathway for designing robots that can operate in new visual domains. We show our approach on the representative tasks of object detection and semantic segmentation in low-light scenarios. Going beyond this, since these represent tasks where label generation is relatively easier and harder respectively, we believe our approaches can be extended to other tasks with similar label scarcity and label-generation complexity as well.

Chapter 2

Related Work

In our work, we provide a pathway to improve the models we can train in new visual domains, which may or may not have a limited availability of labeled data. In subsection 2.1, we compare our work to existing approaches in a limited target domain label setting, and in subsection 2.2 we do so with existing approaches to training models in target domains with no labeled data. We describe the unique position of our new dataset, PittIntensified, in low-light vision scenarios for robotics in subsection 2.3. A short summary of how our approach, `AUDA`, differs from existing methods can be seen in Table 2.1. This is further elaborated in sections 2.1-2.2.

2.1 Domain Adaptation with Limited Supervision

Semi-Supervised Domain Adaptation (`SSDA`) [6, 7, 49, 59] and Few-Shot Supervised Domain Adaptation (`FSSDA`) [40, 50, 64, 66] are two lines of work that assume limited availability of labeled samples from the target domain, similar to `AUDA`. While most `SSDA` algorithms are proposed for image classification, few proposed for segmentation operate in different label regime, requiring hundreds of labeled target domain instances [6, 7], compared to tens used by `AUDA`. With the high-cost and difficulty in the generation of labels for semantic segmentation, this can these methods prohibitively expensive. On the other hand, `FSSDA` techniques like [50, 65] aim to adapt using *only* a limited number (1-5) of labeled samples from the target domain, and generally do not leverage unlabeled target domain data. These are also not

Table 2.1: Summarizing and highlighting how our proposed framework, Almost Unsupervised Domain Adaptation (AUDA), differs from existing classes of Domain Adaptation methods, which are Unsupervised Domain Adaptation (UDA), Semi-Supervised Domain Adaptation (SSDA), and Few-Shot Semi-Supervised Domain Adaptation (FSSDA). It's important to note that, in addition to this, AUDA, can also work in an unsupervised setting similar to UDA methods, by not running the final Supervised Alignment step, while still enchancing UDA with SP. Here, the column corresponding to Effectiveness refers to the relative ability of the class of methods to maximize performance of our domain adapted model in realistic robotic scenarios where some labeled target domain data can often be obtained. Label-Efficiency corresponds to whether the method can operate with little ($\sim 10s$) to no labeled target samples.

Method	Effective?	Label-Efficient?
Unsupervised Domain Adaptation	✗	✓
Few Shot Supervised Domain Adaptation	✗	✓
Semi Supervised Domain Adaptation	✓	✗
AUDA	✓	✓

designed to scale with more available labeled target data. This makes it difficult to adapt across large domain gaps, with these methods usually focusing on adaptation across smaller gaps like adapting across cities in CityScapes [8] [50]. Moreover, in both these cases, there exist far more solutions to image classification than semantic segmentation. These solutions cannot be directly and trivially be applied to semantic segmentation, leaving this problem under-explored.

In contrast, AUDA leverages all unlabeled data alongside limited labeled target data, thereby combining the strengths of both SSDA and FSSDA, enabling label-efficient adaptation across large domain gaps.

While there exist some methods for SSDA and FSSDA, very few have code releases. We compare our work for effective, label-efficient domain adaptation with one representative baseline for each above class of methods, while introducing another baseline for SSDA (See Section 4.2.2). [28] proposes USSS, a universal segmentation model which can be jointly trained across datasets with different label spaces, making use of the large amounts of unlabeled data available by training a common encoder and separate decoders. This set-up naturally allows the usage of labeled and unlabeled target domain data in addition to source data and source trained model to perform

SSDA to our requisite target domain. We use PixDA [50] as a comparative baseline for FSSDA. It attempts to improve DA by targeting pixel-wise class imbalance that leads to ignoring the underrepresented classes and overfitting the well represented ones.

2.2 Unsupervised Domain Adaptation and Domain Generalization

In Unsupervised Domain Adaptation (UDA) [56], data from a labeled source domain and an unlabeled target domain are available. These algorithms employ labeled source data for task supervision, and target data to assist alignment [5, 22, 47, 57]. Generally, they employ an adversarial framework [17, 18, 52, 53] based on [12] and/or propose self-training [29, 34, 55, 68] approaches which generate and use pseudo-labels [32] for the target domain. These works focus on improving UDA given source data and a model trained on it. We take a holistic view of the problem, and enhance Domain Adaptation by focusing on creating more adaptable models through Source Preparation. Our proposal is agnostic to specific algorithms and focuses on providing a platform for improving existing UDA methods. We select Refign [5] added upon DaFormer [21] with HRDA [20], one of the state-of-the-art methods at the time of our study, as one of our baselines. Here, DAFormer forms the baseline UDA method, which is improved upon by HRDA and then Refign. DAFormer is a transformer based approach to Domain Adaptation which uses self-training based on a teacher-student framework, where labeled source data is used for task supervision, and unlabeled target data is used for semi-supervised learning and UDA. HRDA applies a multi-resolution training approach for UDA atop DAFormer, that combines the strengths of small high-resolution crops to preserve fine segmentation details and large low-resolution crops to capture long-range context dependencies with a learned scale attention. Refign then leverages reference source images to improve target predictions by spatially aligning reference predictions with the target, and refining target predictions with adaptive label corrections. In the subsequent sections, we refer to this baseline as Refign-HRDA or simply Refign.

Another class of methods, Domain Generalization [4] assume that target domain is unknown, and aim to perform well under arbitrary domain shifts. However, in most robotic scenarios, target domain is known, and utilizing this as a signal can

help maximize performance, especially with large domain gaps. Thus we do not explore this class of methods. Reducing source domain-specific overfitting has inspired some recent works in domain generalization [24, 62]. However, these methods do not connect this idea with preparing more suitable source models for domain adaptation.

2.3 Robot Vision in Low Light

Vision in low light can be tackled using active or passive sensors and every such sensor represents a new target domain. Active sensors (like LiDAR) are often not applicable due to cost and operating constraints, necessitating the use of passive sensors. Unfortunately, regular cameras are not sensitive enough. While many datasets contain night-time images captured with standard cameras [9, 47, 61] in structured environments (roads with street lights). However, they are unable to capture darker environments important for many robotic tasks such as off-road driving. Our dataset, PittIntensified (Chapter 5), addresses this gap, and to the best of our knowledge, is the first to capture images from high-sensitivity and intensifier cameras in structured and off-road scenarios.

While most aforementioned methods in Section 2.1 and 2.2 focus on adaptation from synthetic-to-real images, or across different conditions with a regular RGB camera, we demonstrate our performance on a wider, more challenging variety of domains across changes in time and lighting [47], and modalities like Thermal [16] and Intensifier Cameras via PittIntensified dataset. These domain shifts include changes across style, noise-patterns, illumination conditions, and more as shown in Figure 1.1.

2.4 Real-time on-robot Object Detection Systems

Object Detection systems in computer vision can generally be separated into one-stage and two-stage detectors [36]. Typically one-stage detectors trade some accuracy for speed as compared to two-stage detectors. This is often crucial for on-robot operations, where real-time updates become critical. `YOLO` (You Only Look Once) [45], `SSD` (Single Shot Detector) [37], and `FCOS` (Fully Convolutional One Stage Object Detection) [51] form popular classes of one-stage detectors, with `YOLO` being the most

popular of them and comprising some of the current State-of-The-Art solutions to real-time object detection [26, 54]. More recently, some transformer based real-time object detection networks such as `RT-DETR` [39] have emerged, seeming to out-perform all prior real-time methods. For our robot, we choose `YOLOV5` [25] as our object detection network. We do so because it offers the right balance between performance, and stability and reliability of its code-base over newer versions of `YOLO(v7-v8)` and `RT-DETR` which may offer superior performance at the cost of a less-stable code-base.

Chapter 3

Extending the Operating Range of Object Detection Systems to Enable On-Robot Low-Light Operations

Today search in both disaster response and zone reconnaissance predominantly relies on humans on the ground, or teleoperation of robots [1, 10, 41]. However, the former places the search teams at great risk, while the latter does not scale as coordination between agents and communication of information over the course of the search is rate limited by the human operators [14]. Since these operations are usually time-critical, the development and use multi-robot autonomous systems that can help humans stay off the ground, while requiring only critical support from human operators, can massively aid these searches.

In order to perform this function, our system needs to be able to identify and locate objects of interest it sees (via object detection), as well as understand the environment it finds itself in (via semantic segmentation). This is important for both continued active search and planning, as well as to alert human agents in-the-loop of potential areas of interest. The circumstances in which such systems need to be deployed can widely vary, necessitating the effective use of an array of new sensors. Additionally, in scenarios where our agent might need to operate in an environment

where there may be other hostile agents, stealth is critically important. The use of active sensors such as LiDAR is akin to shining a flash-light in the infrared spectrum, and makes our agent, the source of the illumination, visible to other agents in the environment. In such cases, our system can be constrained to using only passive sensors. But even under these circumstances, sensors such as thermal and night-vision / high-sensitivity cameras can help enable night-time operations.

3.1 System Setup and Testing Environment

Our system comprises of a custom sensor platform built upon a Grizzly [19] robotic base. The sensor most relevant to our experiments is the high sensitivity camera mounted at the top of our robot. This is pictured in Figure 3.1.

We collect data, and test our system in an off-road environment on the outskirts of Pittsburgh at night, in the absence of any active illumination. In this location, light-pollution from the city still exists, due to which the illumination conditions can be approximated as those one might encounter during a full-moon. Images from our testing environment can be found in Figure 3.2.

Figure 3.1: Our Robotic System with high-sensitivity cameras, Canon ME20F-SH (pictured bottom-right), for night operations mounted at the top.

Figure 3.2: Images of our robot in our operating environment. The first corresponds to an image taken prior to start of operations, the second at time of operations illuminated with a flash-light to illustrate the surrounding darkness, and the third captured with another high-sensitivity camera during operation.

Our operating environment consists of the following object-classes that are relevant to us.

- Person (includes humans and mannequins)
- Pickup-truck
- Friendly (includes ego-vehicle look-alikes, and similarly colored humvee) (also referred to as Grizzly)
- VehicleOther (includes all vehicles that aren't a part of the above classes)

Examples of each of these classes can be found in Figure 3.3. For safety reasons, we use mannequins in place of humans in certain scenarios where the robot is running autonomously.

(a) Person

(b) Pickup-truck

(c) Friendlies

(d) OtherVehicle

Figure 3.3: Illustrative examples of instances of each class. Note that person class includes mannequins.

3.2 Our Solution

Our solutions reflects the nature of our operating environment in which despite appearing completely dark to regular cameras, there still exists some light due to moon/star light and atmospheric light-pollution. This allows us the use of a high-sensitivity RGB camera, such as Canon ME20F-SH. The added benefit of using this modality, is that even though the noise pattern is significantly different with this camera, the images are still somewhat visually similar to regular RGB images.

Our solution also reflects that fact that labeling data from new target domains is often feasible, as was is in our case. This allows us to use a greater amount of supervision for training our object-detector to obtain the best possible model.

In addition to this, size of the region we need to search over can be very large, and in such scenarios, the ability the spot objects in the distance can be critical. This encourages the use of high-resolution images, which can make inference slow. Additionally, since our object-detection sub-system needs to operate on-robot alongside various other processes in real-time, we employ a light-weight single-stage network, `YOLOV5-s` [25].

We train this model using a training scheme slightly modified from the original repository over our data, after starting with a network pretrained on the COCO dataset [35] to bootstrap feature extraction. For this purpose we collect images over three different nights in the outskirts of Pittsburgh, at an off-road location, from the high-sensitivity camera. To create our training set, we annotate 6,000 of these images with objects of classes 'person', 'pickuptruck', 'friendly', and 'otherVehicle'. More details on number of objects can be found in table 3.1.

Table 3.1: Class-wise distribution of ground-truth labels in our training (11,223 images) and test set (604 images).

Class / Instances	Train Set	Test Set
person	13603	1540
pickuptruck	2786	19
vehicleother	3406	268
friendly	3235	407
all	23030	2234

We train our model with this data alongside more data from similar environment during the day-time. Our overall training set comprises of 11,223 images, with 6,000 of these coming in from high-sensitivity camera at night-time, and the rest from a regular RGB camera used during day-time operations. While the noise pattern, colors, and other visual features in our high-sensitivity night time images may be different, the inclusion of day-time data and labels in training aids in our model's robustness, while also supplementing the limited labeled target sample set. Our solution is summarized in Figure 3.4.

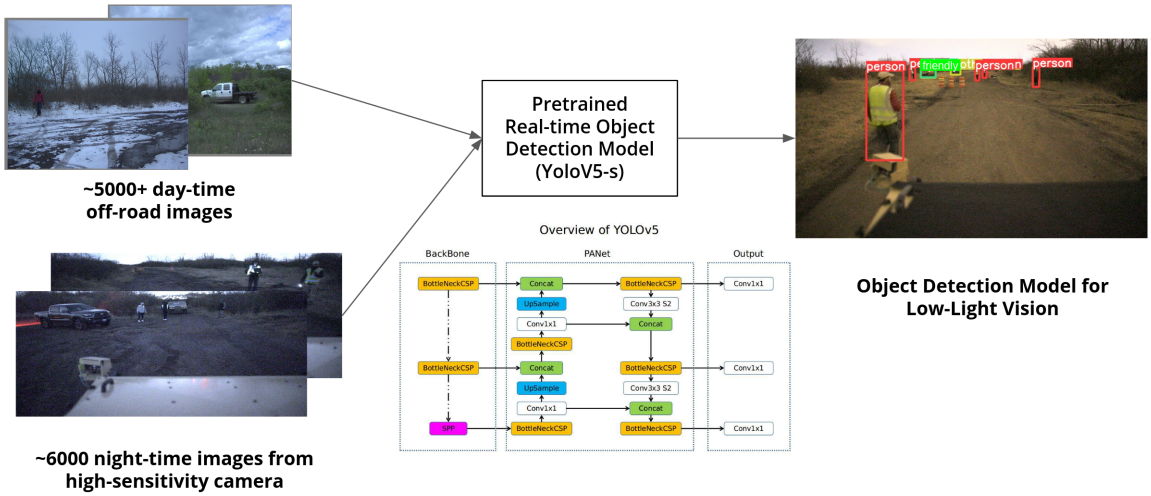

Figure 3.4: We collect and label 6000 night-time images captured with a High-Sensitivity RGB camera, in addition to using roughly 5000 day-time regular camera images to train `YoloV5s`, a real-time object detection model, to obtain our low-light night-ops capable object detection model.

3.3 Results

We demo our system on another night in a similar environment within the same off-road site, and evaluate our performance offline to obtain the following metrics. We test our model on 604 images from the high-sensitivity camera, consisting of 2234 ground truth object labels. The distribution of ground-truth labels is elaborated upon in Table 3.1.

Table 3.2: Class-wise AP50 and AP50-95 on our test-set. These results indicate that our model's performance is very strong, and is capable of night-time object detection. The last row, highlighted in bold, shows results averaged over all classes, i.e mAP.

Class	AP50	AP50-95
person	84.6	55.9
pickuptruck	98.5	88.6
vehicleother	66.0	43.6
friendly	90.6	67.3
all	**84.9**	**63.9**

Table 3.3: AP50 for 'person' class over different sizes (following COCO definitions). With our setup 'small' is roughly equivalent to more than 40 meters from the camera. Our results indicate a strong model performance across sizes.

Class	AP50-s	AP50-m	AP50-l
person	69.3%	95.5%	91.6%

3.3.1 Quantitative

Our results are summarized in Table 3.2. We show strong performance across all our classes, and obtain a mAP50 of **84.9%**. We display the class-wise and aggregate precision-confidence (P-curve), recall-confidence (R-curve), and precision-recall (PR-curve) curves obtained after evaluating our results on our test set in figure 3.5.

We also compare our trained model with existing, off-the-shelf (OTS) pretrained models, i.e. models trained on daytime images captured with regular RGB cameras, in figure 3.6. We do this specifically for the people class, as other classes don't completely overlap with the class definitions in OTS models. We can see that our model

Table 3.4: AP50 for 'person' class with OTS COCO-pretrained YOLOV5s, and our night-time YOLOV5s model on images from the high-sensitivity camera taken during day and night time in the same location. Our in-domain night-trained model outperforms its OTS counterpart in both its performance during the day and at night.

Class	OTS-day	OTS-night	Ours-night
person	70.0%	51.3%	84.6%

significantly outperforms OTS models, with ∼+30% in AP50 over corresponding OTS model, which emphasizes the value-addition corresponding to in-domain supervision. This is also reinforced with our observations in Table 3.4, which additionally also demonstrates the performance degradation observed in OTS models when the input domain shifts to night-time images from a high-sensitivity camera. Our system also works well with objects of different sizes, with results for 'person' class shown in table 3.3.

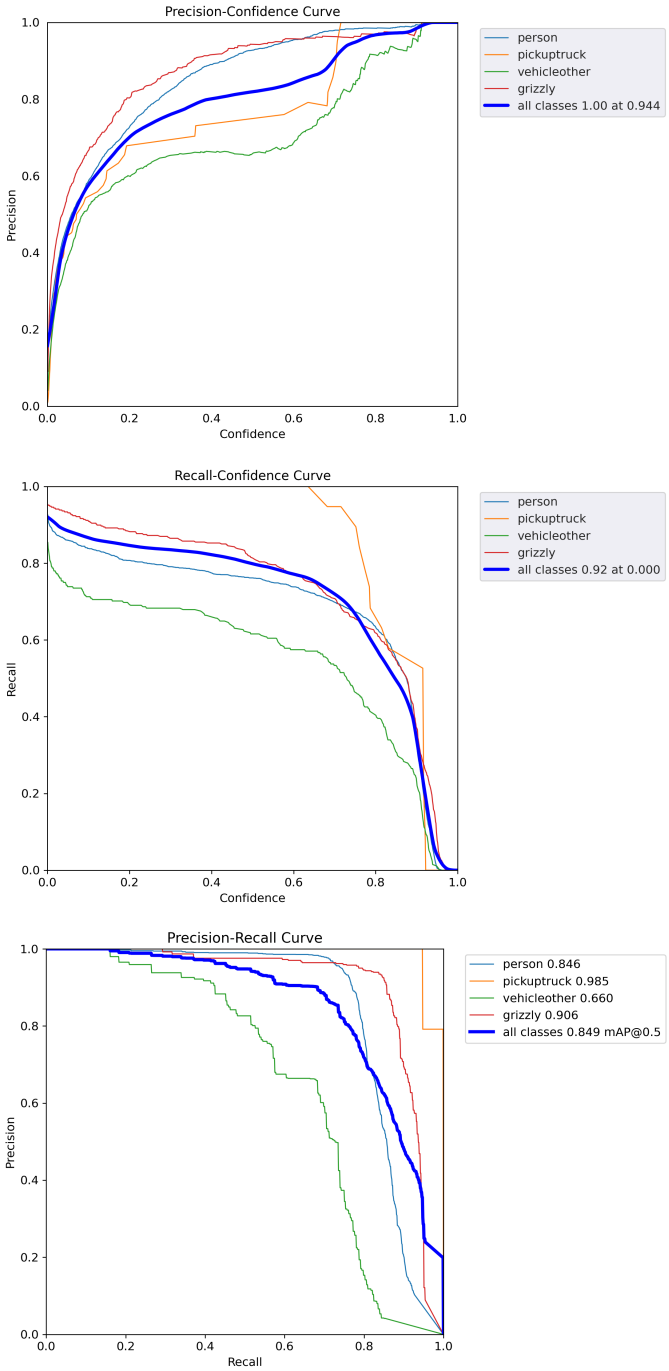

Figure 3.5: `P`, `R`, and `PR` curves of our model on our test set. Plot associated with PR curve contains class-wise `AP50` in labels.

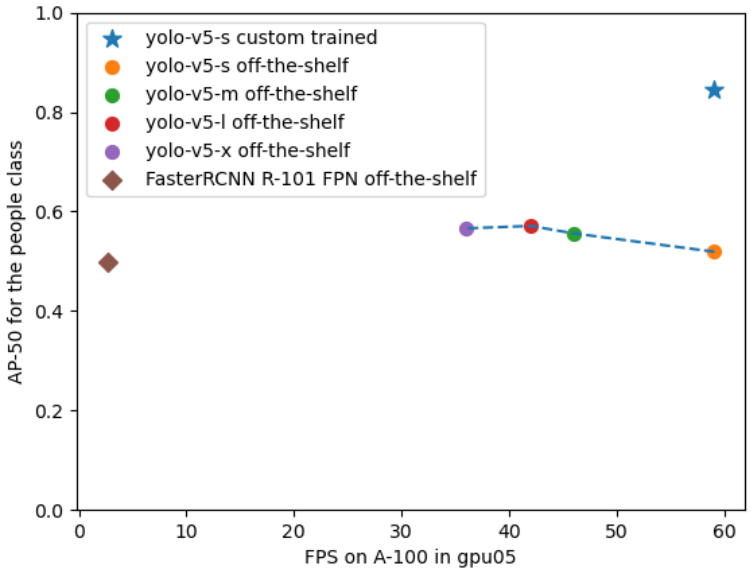

Figure 3.6: `AP50` - person vs Frames Per Second (FPS) for different object-detection algorithms on our set-up.

3.3.2 Qualitative

In Figure 3.7, we can see some examples of detections our system produced on inputs from demo runs. We also show some select cases where our output was subpar in figure 3.8.

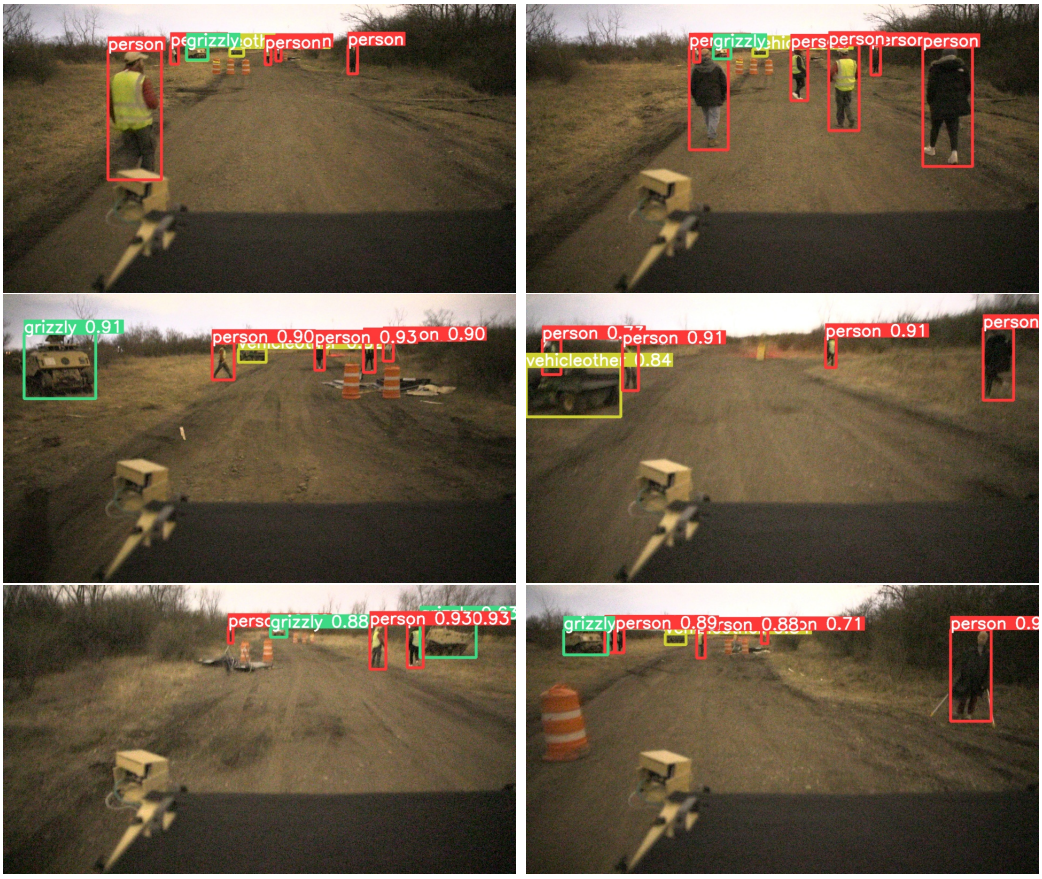

Figure 3.7: Some qualitative outputs of our object-detection model from our demo runs. Our results consistently detect all relevant objects with high quality bounding boxes. "grizzly" is an alias to class "friendly".

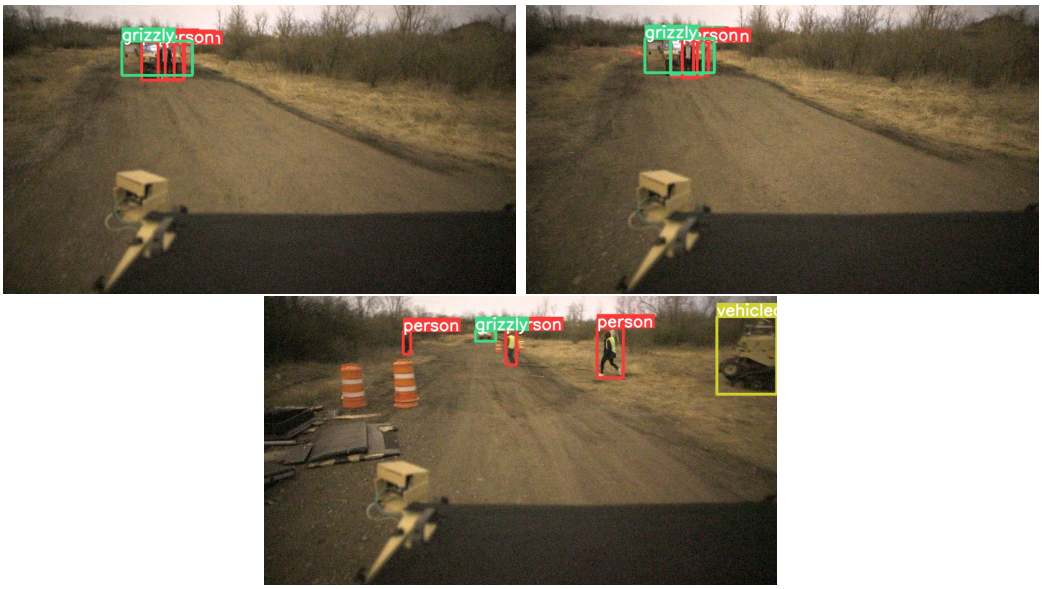

Figure 3.8: Select 'failure-cases' of our object detection model. The first two images correspond to over-counting leading to spurious detections, while the last shows a frame where an object is misclassified when it appeared towards the edge of the frame.

Chapter 4

Training Models on New Domains Effectively in a Label Efficient Manner

4.1 Methodology

4.1.1 Problem Setup

Domain adaptation involves a source domain S abundant in labels, and a target domain T with limited to no labels. Given this setup, our goal is to create a model that performs well on T. To show the efficacy of our proposals, we focus on improving semantic segmentation for realistic robotic scenarios with existing `UDA` approaches, though our work can be extended to other tasks, and forms of `DA`.
Let $D_s = \{x_s^{(i)}\}_{i=1}^{N_s}$ be the set of images from the source domain, where $x_s^{(i)} \in \mathbb{R}^{H_s \times W_s \times 3}$.
Let $L_s = \{y_s^{(i)}\}_{i=1}^{N_s}$ be the set of corresponding one-hot labels for the source domain images, where $y_s^{(i)} \in \{0,1\}^{H_s \times W_s \times C}$, and C is the number of classes. D_t is defined similarly for the target domain.
Let f signify the process of training a segmentation model, ψ_s, on S, where f includes both input data processing and network architecture. Let g be the method performing `UDA` that aims to adapt ψ_s to the T to obtain ψ_t. Traditionally, ψ_s is trained on the labeled source domain data (D_s, L_s) with f, while ψ_t is obtained by applying

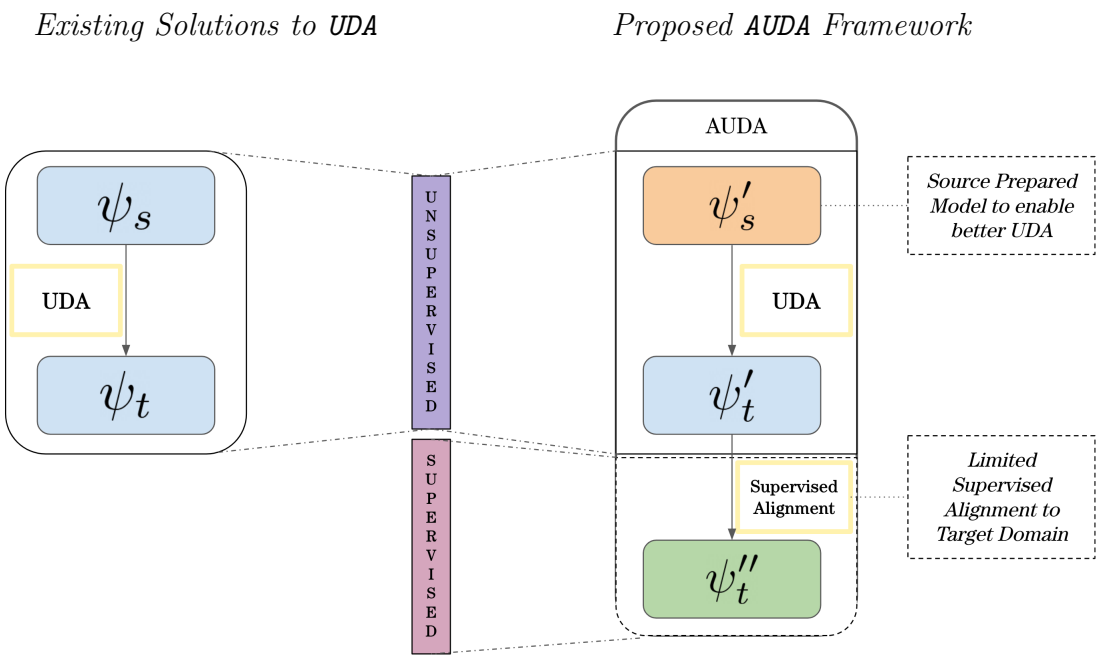

Figure 4.1: This figure illustrates our proposed framework, AUDA, for realistic robotic scenarios where some labeled target samples can be obtained. In contrast to traditional UDA, our approach includes Source Preparation (SP) to create a more 'adaptable' model for UDA, thus enhancing it, while still being fully unsupervised. This is then following by Supervised Alignment (SA) to leverage the limited labeled data available in the target domain.

the method performing UDA, g, to ψ_s using the source domain data (D_s, L_s) and the unlabeled target domain data D_t. For simplicity and ease of understanding, we represent these steps from here on out as $\psi_s = f(D_s, L_s)$, $\psi_t = g(\psi_s, D_s, L_s, D_t)$.

4.1.2 Overview of Proposed AUDA Framework

Our proposed framework for label-efficient DA to T given S can be separated into 3-stages as follows:-

- **Source Model Preparation** for Domain Adaptation using only D_s and L_s.
- **Unsupervised Domain Adaptation** from S to T, using D_s, L_s, and D_t.
- **Supervised Alignment** with limited labeled data in T to improve final performance in T.

Concretely, our Source Preparation step introduces f' in place of f in the original

problem setup. The newly formulated setup now looks like $\psi'_s = f'(D_s, L_s)$, $\psi'_t = g(\psi'_s, D_s, L_s, D_t)$. In Section 4.1.3 we detail how we design f', but it's key to note that we do not propose adding any additional parameters or significantly changing the network architecture. Our final step, Supervised Alignment, performs the following update to obtain our final target model $\psi''_t = h(\psi'_t, D'_t, L'_t)$, assuming we have a labeled target set $\{D'_t, L'_t\}$ where $|D'_t| \ll |D_t|$ and L'_t is the set of labels corresponding to D'_t, defined similar to L_s. Our framework is illustrated in Figure 4.1, wherein we highlight the modifications made to build `AUDA` atop existing `UDA` approaches.

4.1.3 Source Preparation

Our Source Preparation (`SP`) step aims to create source models with features more suitable for domain adaptation. We do so by trying to reduce biases in the source model towards source domain-specific characteristics by addressing overfitting in the source domain. We propose and evaluate the efficacy of 3 schemes across different approaches: (1) explicitly targeting style-based biases in subsection 4.1.3.1, (2) regularization in subsection 4.1.3.2, and (3) high-frequency detail reduction in subsection 4.1.3.3. These schemes however do not form an exhaustive set of all possible approaches to `SP`, rather just aim to demonstrate the promise of `SP`. We explain the motivations behind specific `SP` schemes detailed in subsections 4.1.3.1-4.1.3.3 below. These are elaborated further in the supplemental.

Prior works like [67] suggest that visual domain is closely related to image style. We hypothesize that a source model not overfit to style should be easier to transfer to new domains with varying styles. With this, we develop a scheme detailed in 4.1.3.1.

We hypothesize that increased regularization during the source model's training can help us learn features more robust to domain-specific noise. We propose a `SP` scheme from this intuition which is further detailed in 4.1.3.2.

In domains such as low-light environments, we attribute high-freq noise to be a key component of domain-specific noise. Low-light photon noise and glare are examples of domain-specific noise with high-frequency components, as can be seen in Figure 1.1. While rough shapes are usually preserved across domains such as regular day-night images, and thermal images, the details often vary. In 4.1.3.3, we target this directly.

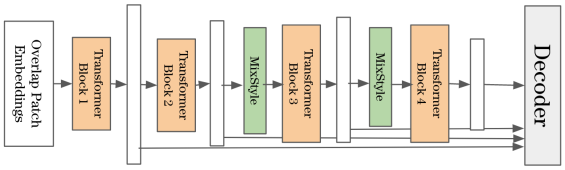

Figure 4.2: `MixStyle` is used for Source Preparation (`SP`) by making the highlighted modification in `SegFormer's` encoder [58]. These modifications are used only for `SP`, and not `UDA` or `SA`.

We use `SegFormer` [58] (`MiT-B5`) as our segmentation model, and explain any modifications made to this network during `SP` below. Note that, we don't add any learnable parameters in any of these modifications, and the unmodified original network architecture is used in subsequent steps. While some of the methods we use in `SP` have been proposed in other settings, we contextualize them in the `AUDA` paradigm and intend to exploit their properties to aid the creation of more 'adaptable' source models for DA.

4.1.3.1 MixStyle

Prior works show that instance-level feature statistics like mean and variance capture style in neural networks [13, 23, 33], including transformers for vision [30]. `MixStyle` [67] is a `DG` approach based on probabilistically mixing these statistics of training samples from source domain(s), to learn features more robust to variations in image-style. For every sample (image) x, MixStyle randomly chooses another sample ($\tilde{x}$) from the same batch, and $\lambda \sim Beta(\alpha, \alpha)$ to give the following update equation,

$$MixStyle(x) = \gamma_{mix}\frac{x - \mu_x}{\sigma_x} + \beta_{mix}$$

Where,

$$\gamma_{mix} = \lambda\sigma(x) + (1 - \lambda)\sigma(\tilde{x})$$

$$\beta_{mix} = \lambda\mu(x) + (1 - \lambda)\mu(\tilde{x})$$

And μ, σ correspond to mean and std dev. of corresponding samples

We follow the original paper on all specifications for implementation. We include `MixStyle` after block-2 and block-3 in `SegFormer's` encoder (`MiT-B5`) to train our

source prepared model, ψ'_s, as illustrated in Figure 4.2.

4.1.3.2 Mixup

`mixup` [63] is a data-augmentation technique that regularizes neural networks to favor simple linear behavior in-between training examples by training it on convex combinations of pairs of samples and their labels. Its formulation is stated below,

$$\tilde{x} = \lambda x_i + (1 - \lambda) x_j$$

$$\tilde{y} = \lambda y_i + (1 - \lambda) y_j$$

Where (xi, yi) and (xj , yj) are two examples drawn at random from our training data (batch), and $\lambda \in [0, 1]$ and $\lambda \sim Beta(\alpha, \alpha)$. Recent work [44] suggests that Out-of-Distribution performance and accuracy can be improved by choosing a high α (10). Since this favors the selection of highly interpolated samples, we perform MixUp for a sample at a probability of 0.5, so that we also see unaltered samples from S during training, leading to improved training stability.

4.1.3.3 Blur

GaussianBlur and other kernel-based blurring methods are commonly employed in data augmentation to enhance the robustness of neural networks against variations in high-frequency details and noise [48]. We propose using a strong blurring scheme during source model training where we blur out images with a 50% chance, using a Gaussian kernel of size uniformly sampled from (5, 5) to (19, 19).

4.1.4 Unsupervised Domain Adaptation

Unsupervised Domain Adaptation (`UDA`) aims to align ψ'_s to T to create ψ'_t, usually with task supervision from $\{D_s, L_s\}$ and supervision for alignment from D_t. While we demonstrate our results, further detailed in Section 4.2, with `Refign` [5], one of the best performing methods at the time of experimentation, our framework is independent of specific `UDA` methods. During `UDA`, our model receives some supervisory signal from both S and T leaving it less likely to be biased towards characteristics

specific to the source domain as compared to the prior step, where only supervisory signal from S is available. This means that we don't require the application of SP techniques as much during UDA. Our framework allows us to stop at this step if we don't have any labels in T, remaining fully unsupervised, while still obtaining the benefits of SP.

4.1.5 Supervised Alignment

Supervised Alignment (SA) aims to account for realistic robotic scenarios where a small amount (20-50 samples) of labeled data in T can be obtained. While we align the model we obtain after UDA with supervision using finetuning, other methods, such as linear probing, can be used here. SA, as a part of AUDA, is more label-efficient than SSDA approaches [6, 7], which typically use 100s of labeled images for from the target domain for tasks like semantic segmentation, while still leveraging all unlabeled data in the first two steps, unlike FSSDA approaches [50, 64], to perform better in the target domain (Section 4.2.2).

4.2 Experiments and Results

Task. We situate this work on the illustrative task of semantic segmentation. However, our approach can however be used for other tasks like panoptic segmentation.

Dataset. We test our proposals across different target domains illustrated in Figure 1.1. With Cityscapes [8] (CS) as our source domain, we adapt to target domains across time and lighting to DarkZurich [47] *(CS→DZ)*, across modalities to MFNetThermal [16] *(CS→MFNT)* and PittIntensified (Section 5) *(CS→PI)*. These also differ across different aspects, with stylistic variations foremost in *(CS→DZ)* and *(CS→MFNT)*, and noise patterns being of great importance in addition to style in *(CS→PI)*. Some differences are highlighted in Figure 1.1. We evaluate our solutions on labels common to both source and target domains. While *PI* and *MFNT* have train sets for SA, *DZ* does not.

Implementation Details. We choose Refign-HRDA* [5] and SegFormer (MiT-B5) [58] as our UDA method, and segmentation network respectively. We train Refign with a scheme similar to the original paper, in exception to increasing iterations

Table 4.1: Comparison (mIoU) on respective validation sets after UDA from Cityscapes to DarkZurich, MFNet Thermal, PittIntensified with different Source Preparation techniques. In each case we can improve the potency of UDA with the right kind of Source Preparation.

DA-Method	SP Method	CS→DZ	CS→MFNT	CS→PI
None	None	29.3	55.3	4.6
Refign	None	49.0	63.5	32.5
Refign	MixStyle	**49.5**	**65.0** (+1.5)	50.8
Refign	mixup	47.4	62.7	71.9 (+39.4)
Refign	Blur	49.4	60.4	**73.1** (+40.6)
Refign	MixStyle+mixup+Blur	47.9	64.8	**73.1**(+40.6)

1.5×, and SegFormer as the original paper does. In SA, we finetune Segformer for 4000 iterations, scaling down warm-up iterations of the 'poly' scheduler to 150. Our approach is independent of specific UDA methods and should extend to others.

4.2.1 Effect of Source Preparation

In this section, we compare different SP methods and their impact on target domain performance after UDA. We also show that SP can make the models we obtain after UDA more robust in 4.2.1.1, and that it can improve the models we obtain after Supervised Alignment in 4.2.1.2.

In Table 4.1, we show the performance of models obtained after UDA and different SP schemes we had proposed. We analyze and explain our results based on target domain characteristics below.

MixStyle. Regularizing over styles with MixStyle improves performances in both cross-modal and cross-time tasks, with the highest improvement of all tested SP schemes in *CS→MFNT* (+1.6% mIoU) and *CS→DZ* (+0.5% mIoU). It also significantly improves *CS→PI* by +18.3% mIoU.

Mixup. Regularization from mixup boosts the cross-modal task, *CS→PI*, with +39.4% mIoU, which is 2.2× what we obtain without SP. We hypothesize that regularizing during SP helps prevent the model from biasing toward photon noise and glare in low-light images captured by an intensifier.

Blur. We improve performance across all our source-target pairs with low-light

noise. In *CS→PI* we obtain a boost of +40.6% mIoU, which is 2.25× what we obtain with no SP. This can be attributed to high frequency patterns in the source domain corresponding to surface texture whereas in this particular target domain, noise contributes very significantly to high frequency patterns, i.e. overfitting to high-frequency detail in the source domain can lead to the target model being biased towards noise.

PittIntensified proved to be very challenging for the baseline source model prior to UDA, at 4.6% mIoU, indicating that of the features learnt by the source model, few were relevant across these domains, i.e. atleast some important features were source domain-specific. This resulted in limited improves with UDA. We hypothesize that our SP techniques greatly enhanced UDA because our source models are more adaptable. Since our SP mainly focuses on mitigating overfitting in the source domain, all results above validate our hypothesis that a source model less biased toward different kinds of source domain-specific characteristics is more suitable for adaptation.

Additionally, combining SP schemes, i.e. training our source model while using multiple SP schemes concurrently, can give us a source model that can transfer very well across multiple domains. More details on effectively combining SP methods are given in subsection 4.9.

Selecting the right SP method. From the trends we observe in our experiments, we can extract guidelines for selecting or designing the right SP method for a specific T. We refer the reader back to figure 1.1 to understand how the different domains we've selected differ from the source domain, and how we can extend our 'intuitions' to other similar target domains. If T has a lot of high-frequency noise, techniques that aim to reduce sensitivity to such noise, like blurring and regularization with mixup, appear enhance DA best. If there is a significant difference in style between the source and target domains, regularizing over style, as with MixStyle, is an effective SP technique. Often source and target domains will differ along different axis, and combining multiple SP schemes could further enhance DA across these domains.

4.2.1.1 Effect of Source Preparation on Robustness

SP not only enhances performance in the target domain but also increases the robustness of the adapted model to possible real-world changes 'within' it. These

changes correspond to an additional domain-shift, such as the onset of adverse weather conditions like rain, fog, and snow while operating in our original target domain, such as night time images. It is important to note that while our target model has been trained to operate across the first domain-shift, it has not been trained on any additional domain-shifts. Our results are detailed in Table 4.2, and examples of augmentations, generated using imgaug [27], are shown in Figure 4.3. We modify the images in the DarkZurich Val (DZv) set to add rain, fog, snow, and increased motion blur. We also 'cartoonify' the images to test across another stylistic variation. In all cases, models obtained after `UDA` with `SP` beat models obtained without `SP`, with +5.5% mIoU in DZv-rainy, +1.6% mIoU in DZv-snowy being examples. We attribute this to reduced sensitivity to noise and stylistic variations. Using multiple forms of `SP` together shows a great increase in robustness as well, supporting aforementioned attribution, as this aims to reduce sensitivity to both noise and stylistic variations.

Figure 4.3: Examples from DarkZurich augmented with rain, snow, fog, increased motion blur, and cartoonification.

4.2.1.2 Improving Supervised Alignment with Source Preparation

We evaluate the performance of models obtained after `UDA`, with and without `SP`, after performing `SA` in the form of finetuning with a very limited number of labeled samples from T. We show our results in Table 4.3 on $CS \rightarrow MFNT$ and $CS \rightarrow PI$ across labeled target train sets of different sizes. In each case, barring PI with 100 (comprising its entire train set), we perform four rounds of finetuning on the same randomly subsampled portions of the train set for each method, and subsequently average the results. `SP` improves performance in T after `SA`, particularly in cases with very few labeled samples from T, such as +4.5 mIoU on $MFNT$ with 20 samples, +4.7 mIoU on PI with 50 samples as compared to finetuning the model without `SP`, while also generally giving results with less variation.

Table 4.2: Comparison (mIoU) on DarkZurich Val under various additional real-world shifts (and another style-shift) in the target domain after UDA from Cityscapes with different Source Preparation (SP) techniques. SP-stacked refers to MixStyle+mixup+Blur (all included in source model training with equal weight-age). SP performs better across all shifts, indicating increased robustness in the target model.

SP Method	Original	Rainy	Snowy	Foggy	Motion Blur	Cartoonified
None	49.0	28.8	35.2	36.0	42.4	18.3
MixStyle	**49.5**	26.7	34.7	**36.3**	40.7	18.6
mixup	47.4	33.0 (+4.3)	34.8	36.2	41.5	18.0
Blur	49.4	**34.3** (+5.5)	**36.8**	35.7	**42.5**	**20.1**
SP-Stacked	47.9	34.0 (+5.2)	34.8	**36.8**	**43.1**	19.1

Table 4.3: Comparison (mIoU) on respective validation sets after limited Supervised Alignment (SA) of the models with and without best performing Source Preparation (SP), and UDA from Cityscapes. Incorporating SA after both SP and UDA yields the best-performing models in the target domain, particularly when labeled target samples are scarce. The highlighted improvements correspond to improvements along the column.

Dataset	SP?	UDA?	Number of labels for SA			
			0	20	50	100
CS→MFNT	✗	✗	55.3	66.0 ±1.4	77.2 ±1.2	79.1 ±4.3
	✗	✓	63.4	74.9 ±8.7	84.2 ±3	85.4±1.3
	✓	✓	**65.0**	**79.4** (+4.5) ±2.3	**84.5**±2.1	**85.7** ±0.8
CS→PI	✗	✗	4.6	49.7 ±7.3	58.4 ±3.1	69.8
	✗	✓	32.5	73.4 ±1.7	77.0 ±2	80.8
	✓	✓	**73.1**	**74.3** ±0.8	**81.7** (+4.7)±1.5	**81.9**

4.2.2 AUDA for Effective Label Efficient Domain Adaptation across large Domain Gaps

Stage-wise contributions in AUDA. We present experimental results of our proposed framework, AUDA, detailing the contributions of each step in Table 4.4 , demonstrating their positive impact. Across different source-target pairs, different stages are most effective. In *CS→DZ*, UDA improves target performance the most, at +19.7% mIoU,

Table 4.4: Summarizing the contribution of each stage of AUDA, with 50 labeled target samples of SA, with their improvements (mIoU) highlighted. Results shown over respective validation sets.

Method	CS→DZ	CS→MFNT	CS→PI
Baseline	29.3	55.4	4.6
UDA	49.0 (+19.7)	63.5 (+8.1)	32.5 (+27.9)
SP + UDA	49.5 (+0.5)	65.0 (+1.6)	73.1 (+40.6)
SP + UDA + SA	N/A	84.5 (+19.5)	81.7 (+8.6)

Table 4.5: Comparison (mIoU) on respective validation sets with the best performing SP technique for each dataset applied as a separate SP step before UDA or together with UDA. Results indicate that a separate SP generally yields superior target models.

SP?	SP modification during UDA?	CS→DZ	CS→MFNT	CS→PI
✗	✗	49.0	63.5	32.5
✗	✓	48.4	65.3	40.0
✓	✗	**49.5**	65.0	**73.1**
✓	✓	47.6	**65.5**	51.9

while SA does so in *CS→MFNT*, with +19.5% mIoU, and SP in *CS→PI*, SP increases target domain performance by +40.6% mIoU.

Necessity of SP. We compare SP techniques applied directly during UDA and in a separate SP step to investigate the necessity of a preparatory step. In each case we use the SP method that performs best for the domain-shift. Results in Table 4.5 show that a separate SP step consistently yields superior outcomes, supporting our hypothesis that source models need to be made more adaptable before UDA. Additionally, while SP technique applied directly during UDA can be helpful in some cases, it has little or negative effect when applied with separate SP.

Comparisons with FSSDA. In Table 4.6, we compare AUDA with an instantiation of FSSDA, PixDA [50], on *CS→PI* and *CS→MFNT*. We provide both approaches access to the same set of labeled data from the target domain (20 labeled samples in *CS→MFNT*, 50 in *CS→PI*) during training, and report the best of 1-shot and 5-shot performance during evaluation. The backbone segmentation networks are however different with PixDA using DeepLabv2, and AUDA using SegFormer(MiTB5).

AUDA performs significantly better in both these cases, indicating having a greater ability to adapt across larger domain gaps, which we attribute to SP, and exploitation of unlabeled target samples. SSDA approaches typically use hundreds of labeled target samples, and cannot be utilized in these scenarios.

Table 4.6: Comparison (mIoU) between AUDA and PixDA. Results shown for validation sets of respective datasets. These indicate that AUDA can adapt more effectively across larger domain gaps.

Method	CS→MFNT	CS→PI
PixDA	17.1	16.5
AUDA	75.5	82.6

Comparison With SSDA. While we previously stated that SSDA approaches typically use hundreds of labeled target domain samples for domain adaptation, in this section we show that their performance degrades rapidly in the limited label scenarios we are working with.

We compare AUDA with two different SSDA approaches. First, we modify Refign-HRDA [5] such that we add labeled target images along with the labeled source images as a part of the task-supervision, in addition to providing unlabeled target samples for alignment just as before. We term this Refign-SS and train it with the same scheme and hyper-parameters we use to train Refign as a UDA method in AUDA. For the second, we modify Universal Semi-Supervised Semantic Segmentation (USSS) as introduced in [28] by replacing DRNet [60] (as used in the original paper) with SegFormer [58] to ensure a fair comparison, and improve it by doing so. While USSS assumes partial annotations in both (source and target) domains, we provide all available source domain annotation, in addition to limited target annotations, and all unlabeled samples in both domains. We train USSS with its official code release, using all associated hyper-parameters. In all our comparisons we provide access to the same randomly selected sets of labeled target samples of different sizes. MFNetThermal [16] (*MFNT*) with 1567, and PittIntensified (*PI*) with 100 target samples correspond to utilizing the entirety of their respective train sets.

Our results over the validation set of each dataset, shown in Table 4.7, clearly demonstrate that AUDA performs much better in label scarce scenarios than other

approaches in our comparison, with up to +34.75% mIoU in *MFNT* with 20 target labels. Moreover, it is important to note that with an increase in target label scarcity, the degradation in performance is far steeper in existing SSDA approaches as compared to AUDA. In *MFNT*, AUDA reaches 92.5% of its performance with the full-training set, with just 20 labeled target domain samples, as compared with Refign-SS reaching just 63.4% of its performance under full-supervision. Similarly, in *PI*, AUDA reaches 89.2% of its performance under full-supervision as compared to Refign-SS's 82.1%, both with 20 labeled target samples. This is in addition to AUDA's ability to be used in a target label-free scenario as well, with only the use of the first two of the three steps, i.e. SP and UDA. This is important for environments and tasks where iterative development is critical, as this provides us with the ability to first train a model in the new target domain without any supervision, deploy it, and iteratively improve it with SA, without having to retrain it entirely.

Table 4.7: Comparison (mIoU) to showcase label-efficiency of AUDA vs other SSDA approaches. AUDA not only performs better under label scarcity, but the degradation in performance as we approach label scarcity is also reduced.

Dataset	Method	Number of labels for SA				
		0	20	50	100	1567
CS→MFNT	USSS	-	29.1	35.3	43.2	59.1
	Refign-SS	63.5	46.1	61.4	68.3	72.6
	AUDA	**65.0**	**80.8** (+34.7%)	**84.1** (+22.7%)	**85.0**	**87.3**
CS→PI	USSS	-	66.9	71.0	73.7	-
	Refign-SS	32.5	69.4	80.7	**84.5**	-
	AUDA	**73.1**	**74.3**	**82.6**	83.2	-

4.2.3 SP with an Alternate Domain Adaptation Approach

To test the ability of SP in improving other techniques and approaches to domain adaptation, we run the modified USSS algorithm from above (an approach to SSDA), with and without a source prepared source model trained on Cityscapes [8] (*CS*). We report the outcome of these experiments in Table 4.8. Our results indicate that SP can significantly improve performance across these semi-supervised domain adaptation

Table 4.8: Comparison (mIoU) to showcase the effect of SP on a different Domain Adaptation technique, USSS. SP shows that it can boost performance across both datasets and different levels of label scarcity.

Dataset	SP?	Number of labels for SA			
		20	50	100	1567
CS→MFNT	✗	**29.1**	**35.3**	43.2	59.1
	✓	26.6	30.9	**50.0** (+6.8%)	**67.4** (+8.3%)
CS→PI	✗	66.9	**71.0**	73.7	-
	✓	**67.6**	70.3	**76.3** (+2.6%)	-

techniques as well, with +8.3% and +6.8% in mIoU in *CS* → *MFNT* with 1567 and 100 labeled target samples respectively, and +2.6% mIoU in *CS* → *PI* with 100 labels.

4.2.4 Towards Appropriately Stacking Different SP Schemes

While approaching source preparation with the intention to reduce different forms of source domain biases at the same time may be effective, naively performing all of our SP schemes together, i.e. naively stacking our SP schemes without allowing the model to see un-altered source prep images regularly, does not work very well. We show the results of our experiments in Table 4.9, in which we compare the performances of the model obtained after the best single SP method for each dataset with SP-stacked, and SP-stacked-naive after UDA. Here, SP-stacked differs from SP-stacked-naive in that we reduce the likelihood at which our SP schemes are enacted from p=0.5 to p=0.3. This ensures that stacking different SP methods together does not drown out signal from the un-altered source image, i.e. probability of training on original sample with just one SP method is 0.5, but when stacked keeping the same likelihood for each of the three SP methods in stacking, this goes down to 0.125 (corresponding to $(0.5)^3$). Across all domain adaptations, we can see that our performance degrades upon naively stacking SP methods, indicating that some consideration on this front is necessary while designing new SP schemes. Moreover while our experiments focus on combining SP schemes while equally weighing the individual methods equally, combining them with more nuance based on the axis where the source and target domain differ the

most will very likely create an even better model for each target domain. It is also critical to note here that while we select the best performing SP method across each domain pairs for SP-Single, we use the same stacked combination for our models trained with stacked SP methods.

Table 4.9: Comparison (mIoU) to showcase the effect of chaining our SP schemes vs best individual SP scheme for each dataset after UDA.

Dataset	SP-Single	SP-Stacked-Naive	SP-Stacked
CS→DZ	49.5	43.5	47.9
CS→MFNT	65.0	60.9	64.8
CS→PI	73.1	49.7	73.1

4.2.5 Analyzing Qualitative Results

Figures 4.4, 4.5 and 4.6 show qualitative results after different stages of AUDA on *PI*, along with results after just UDA without SP to understand and show the efficacy of SP. In all figures, all results corresponding to a particular image are added to the same column as the image. The first row corresponds to the query image, the second to the output we obtain after UDA without SP, the third to the output we obtain after UDA with SP, the fourth after SP, UDA, and SA, i.e. complete AUDA, and the last corresponds to the ground truth labeling. While we show predictions that go into region of the image blocked by the frame of the intensifier module, these regions are marked to belong to the 'invalid' class, and so don't affect any quantitative metrics.

From these figures, we can see that UDA with SP greatly helps with the reduction of both false positives and false negatives. This happens particularly in images, or regions of images with high noise, such as the images captured in a dark park, which has a lot of low-light noise, or in regions with bright lights on streets. Both of these have high-frequency components. With SP with our blur-based scheme, we make our source model more robust to variations in such features, which leads to enhanced domain adaptation as we had hypothesized.

We can also see that SA generally refines, and further improves the outputs we obtain after UDA and SP, and gives us the results closest to the ground truth (last row).

Our qualitative results thus support our hypotheses of making source models more adaptable to enhance domain adaptation with SP, and of using limited SA to improve the models we can train in limited target label settings. The captions under Figures 4.4, 4.5 and 4.6 highlight some changes to behavior which we can observe upon using AUDA in CS→PI. Additionally, Figure 4.7 shows the performance at night-time with a regular RGB camera after AUDA on CS→DZ, and Figure 4.8 highlights some of the improvements in performance we obtain after SP+UDA on CS→MFNT on background classes, which is not reflected in our quantitative results as the ground truth labels for these don't exist in MFNT.

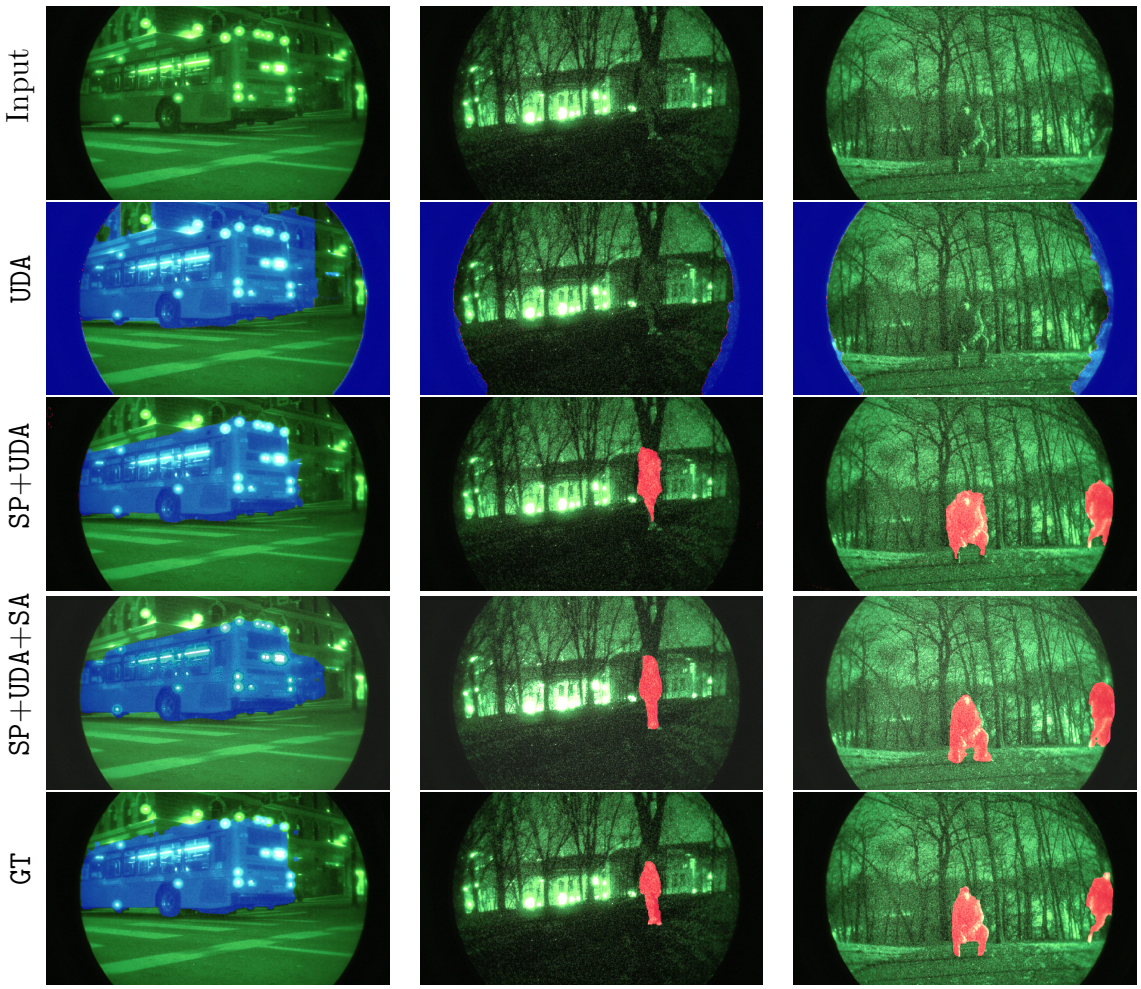

Figure 4.4: Stage-wise qualitative results, with the fourth row corresponding to `AUDA` = `UDA` + `SP` + `SA`. Each step of `AUDA` improves our final outputs. The inclusion of `SP` with `UDA` massively reducing both false positives and false negatives as compared to predictions with just `UDA`. This is particularly true for 'vehicle' class predictions. Further inclusion of `SA` refines our labels, as can be seen in the case of 'person' class predictions.

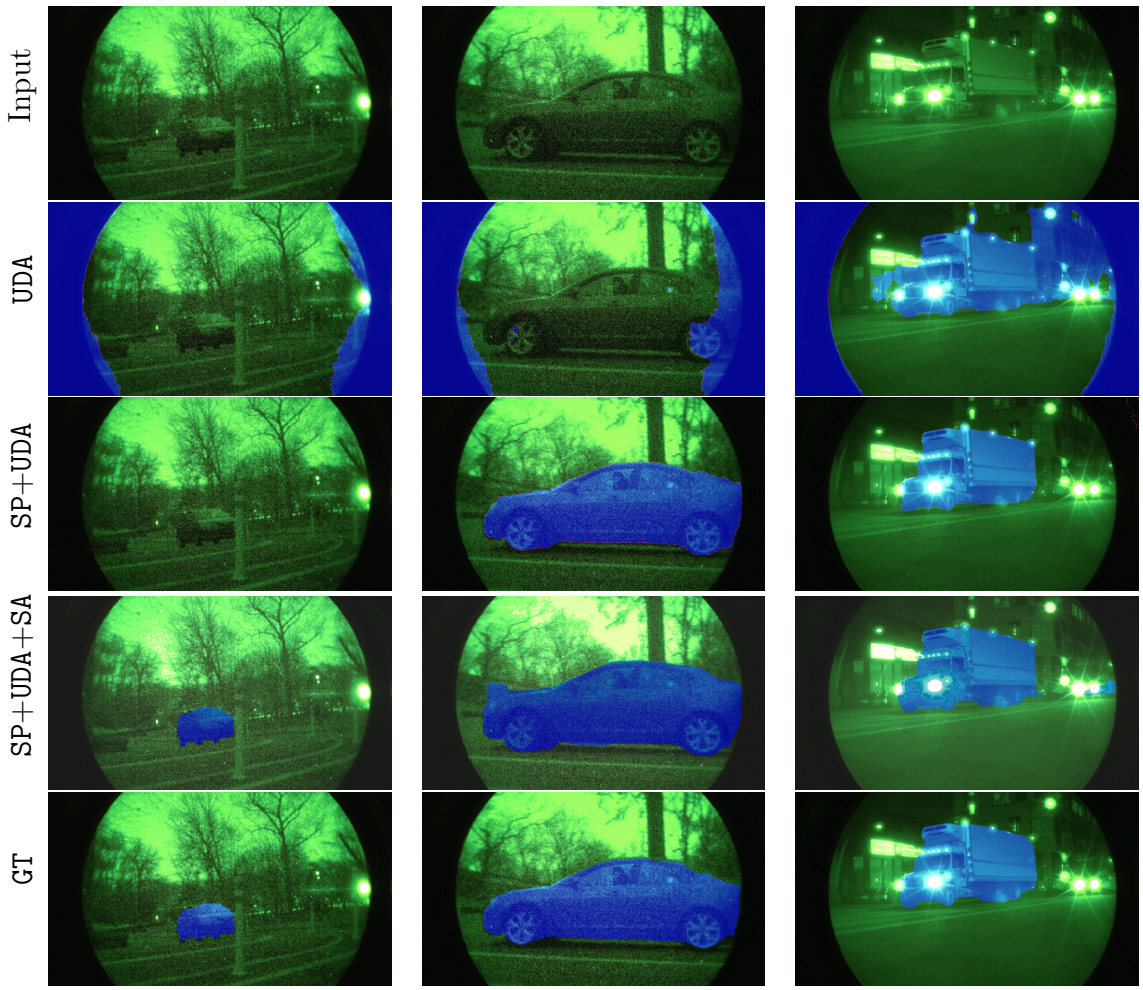

Figure 4.5: Stage-wise qualitative results, with the fourth row corresponding to AUDA = UDA + SP + SA. Each step of AUDA improves our final outputs. The inclusion of SP with UDA massively reducing both false positives and false negatives as compared to predictions with just UDA. Further inclusion of SA helps sweep-up objects formerly missed out, as we cam see with the car in the first example.

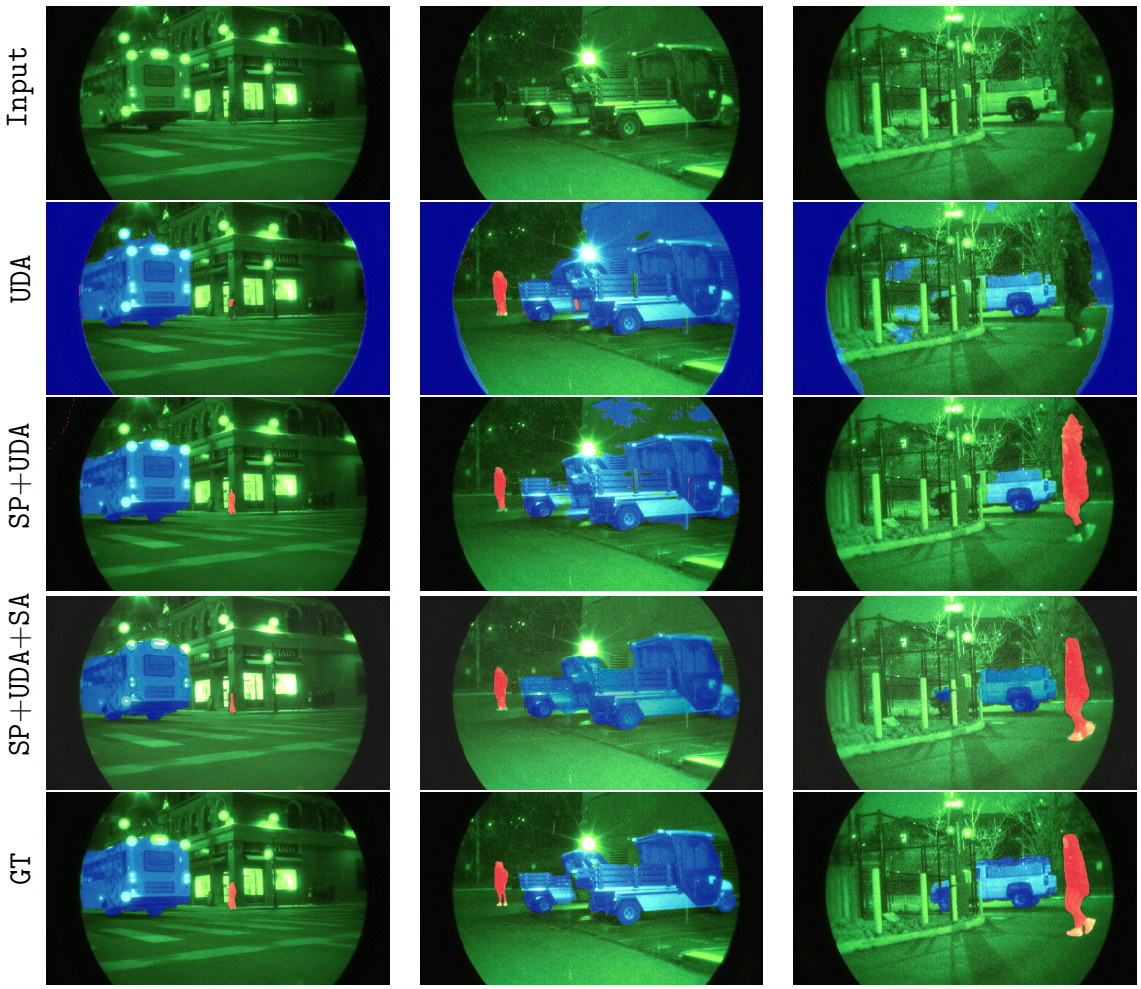

Figure 4.6: Stage-wise qualitative results, with the fourth row corresponding to AUDA = UDA + SP + SA. Each step of AUDA improves our final outputs. The inclusion of SP with UDA massively reducing both false positives and false negatives as compared to predictions with just UDA. Further inclusion of SA helps sweep-up objects formerly missed out, while also refining predictions of both 'people' and 'vehicle' classes, as well as reducing erroneous widely off-the-mark false positive predictions of 'vehicle'.

Figure 4.7: We can see significant improvement in these representative examples from CS → DZ after AUDA. Our predictions are generally far smoother, performance particularly in background classes is far improved, and our model looks to be capable of operating in such night time environments.

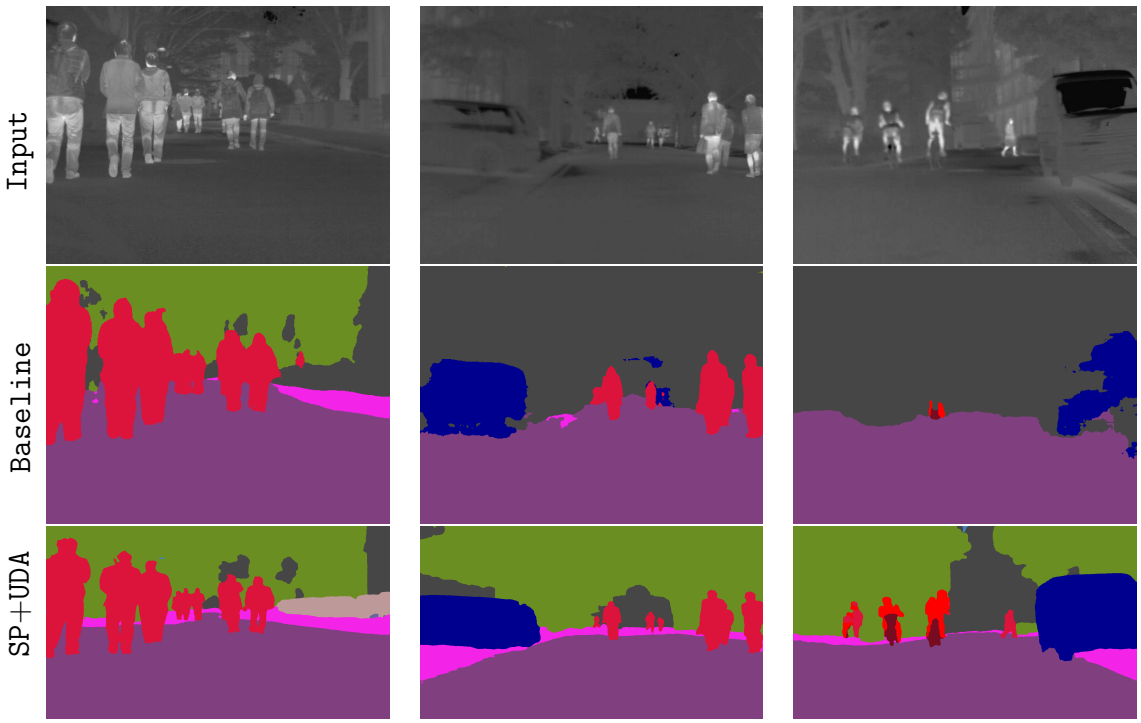

Figure 4.8: We can see significant improvement in represenative examples from `CS` → `MFNT` after `SP+UDA`, with our output, even prior to `SA`, being very usable in thermal domains. While background prediction is significantly improved, it is not reflected in our quantitative results owing to lack of ground truth labels corresponding to these classes. Additionally foreground objects, once missed or crudely segmented, are captured much more effectively.

4.2.6 Additional Details on Methods, Experimental Set-up, and Compute Use

Selection of SP strategy for blur. While we experiment over a different range of kernel sizes while formulating our gaussian blur based `SP` scheme, our key decision choice was the nature of the random sampling over our range of possible kernels of sizes (5,5) to (19,19). We compared sampling uniformly over this range against sampling with a normal distribution centered around 12, with a standard deviation of 3.5. After `UDA` with `Refign-HRDA*`, the mIoU with a uniformly sampled Gaussian blur, 73.14% was more than what we observed with the normally sampled counterpart, 68.2%. This suggested that sampling more evenly from a range of sizes among blur kernels provided a more useful signal for training. This however indicates that the use of a more complex and varied blurring scheme may further improve our `SP` scheme. In this particular case, where photon noise is mainly poisson noise, a blurring strategy that more closely approximates this could also be a viable alternative, but we choose to use gaussian blur due to its ease of incorporation as its generally built into to all popular computer vision and deep learning pipelines.

Selection of Classes used in Evaluation for different datasets. Since the algorithms we use for `UDA` maps across domains while assuming a common set of labels in both domains, we evaluate our algorithm based on performance across only common classes, i.e. to obtain mIoU we take the average of IoU over these select classes. In the case of *Cityscapes→DarkZurich* [47], this includes all 19 classes used for Cityscapes evaluation. For *CS→MFNT*, we evaluate over cars, person, and bike classes of the *MFNT* dataset. To account for similar classes in Cityscapes, we remap predictions for both motorcycle and bicycle to *MFNT* class bike, and person and rider to *MFNT* class person. Similarly, since *PI* contains classes for 'people' and 'vehicles', the latter of which comprises cars, buses, and trucks, we remap predictions for person and rider to *PI* class people, and car, truck, and bus to *PI* class vehicle. For consistency, we compute mIoU across these select classes in all aforementioned experiments.

Compute Costs. All our experiments have been run in a single GPU setting, with NVIDIA A100 40GB [43] GPUs. While training times would defer based on the choice of specific architectures and methods in our framework, a single complete

training of AUDA using SegFormer and Refign with SP, UDA, and SA takes approximately 2.5 days on a single GPU. Once a source model is trained with SP, it can however be used to train UDA to different target domains without any additional training costs. SA takes a fraction of the time the other two components take since we run it for just a few iterations.

Chapter 5

PittIntensified : Intensified Images for Vision in Low-Light Scenarios

We introduce PittIntensified, a new dataset designed for low-light robotic scenarios, where the utilization of light-sensitive sensors allows for maximally exploiting available light. To the best of our knowledge, no such dataset exists in the public domain. By employing an intensifier as a sensor for low-light vision, PittIntensified functions as an additional sensing modality that fills the void between regular RGB cameras, which lack the required sensitivity for such scenarios, and thermal cameras, which operate in a different wavelength range and don't aim to maximally utilize any available visible light. It comprises pairs of images from a High-Sensitivity RGB camera and an Intensifier camera, with semantic and instance level labels for a select subset of Intensifier images. While High-Sensitivity RGB cameras let us see the world in color-balanced RGB even at night, they are typically significantly more expensive than Intensifier cameras, which makes their incorporation in robots, particularly smaller robots, difficult. We can however use the high-sensitivity RGB camera to help bridge the domain gap to the cheaper intensifier cameras. We aim to faciliate this through paired images across cameras.

5.1 Collection and Labeling Set Up

The dataset comprises 4792 image pairs captured at night, featuring a high-sensitivity RGB camera (Canon ME20F-SH) and an intensifier camera (Canon ME20F-SH with AstroScope 9350-EOS-PRO Gen 3), in diverse low-light settings encompassing public streets and parks.

We provide semantic and instance-level labels, obtained by manually correcting labels generated with SegmentAnything [31], for people and vehicles in 393 intensifier images. This is split into a validation set consisting of 293 images, and a train set for limited Supervised Alignment with 100 images. With our paired dataset, we hope to facilitate building a bridge between RGB images, which are captured by the high-sensitivity camera, to images from the intensifier.

We collected the data that comprises PittIntensified on two separate nights in Pittsburgh, Pennsylvania in the United States, with all images of size 960×540. It has a total of 11 sequences, 5 of which are taken within a park to capture scenes with minimal city light from sources such as buildings and street lights, 5 on-road, and 1 in a parking lot. Figure 5.1 captures our recording set-up with a regular phone camera and gives an idea of how dark these scenes appear before intensification. More examples from PittIntensified can be found in Figures 5.2, 5.3, 5.4, where we show image pairs, and corresponding segmentation, detection, and instance labels for people and vehicles.

Figure 5.1: Representation of how these scenes appear with a regular camera.

5.2 Qualitative Examples

Figure 5.2 holds some representative examples of our object detection, and instance segmentation labels for intensifier images. We provide illustrative samples for segmentation-labeled pair-wise images in Figure 5.3 and 5.4. The first column corresponds to High-Sensitivity RGB images, the second to Intensified Images, and the third to ground truth segmentation labels.

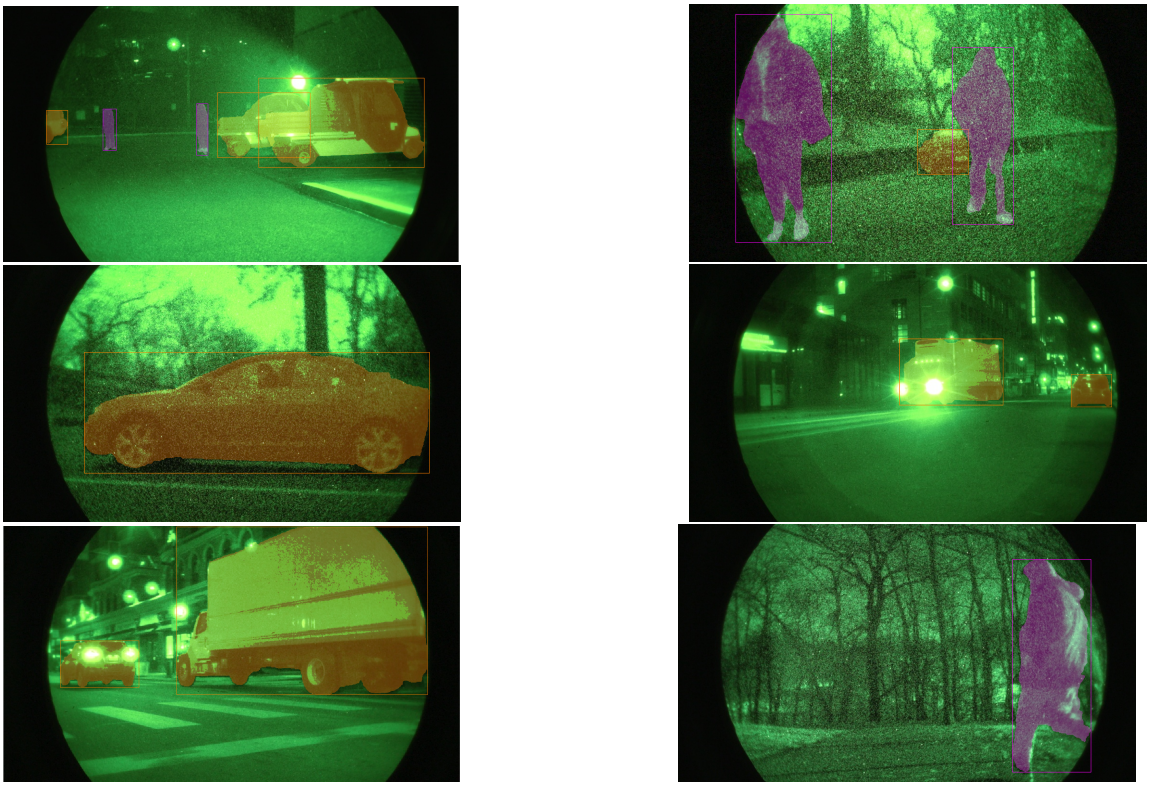

Figure 5.2: Representative samples from PittIntensified, with object detection (bbox) and instance segmentation labels.

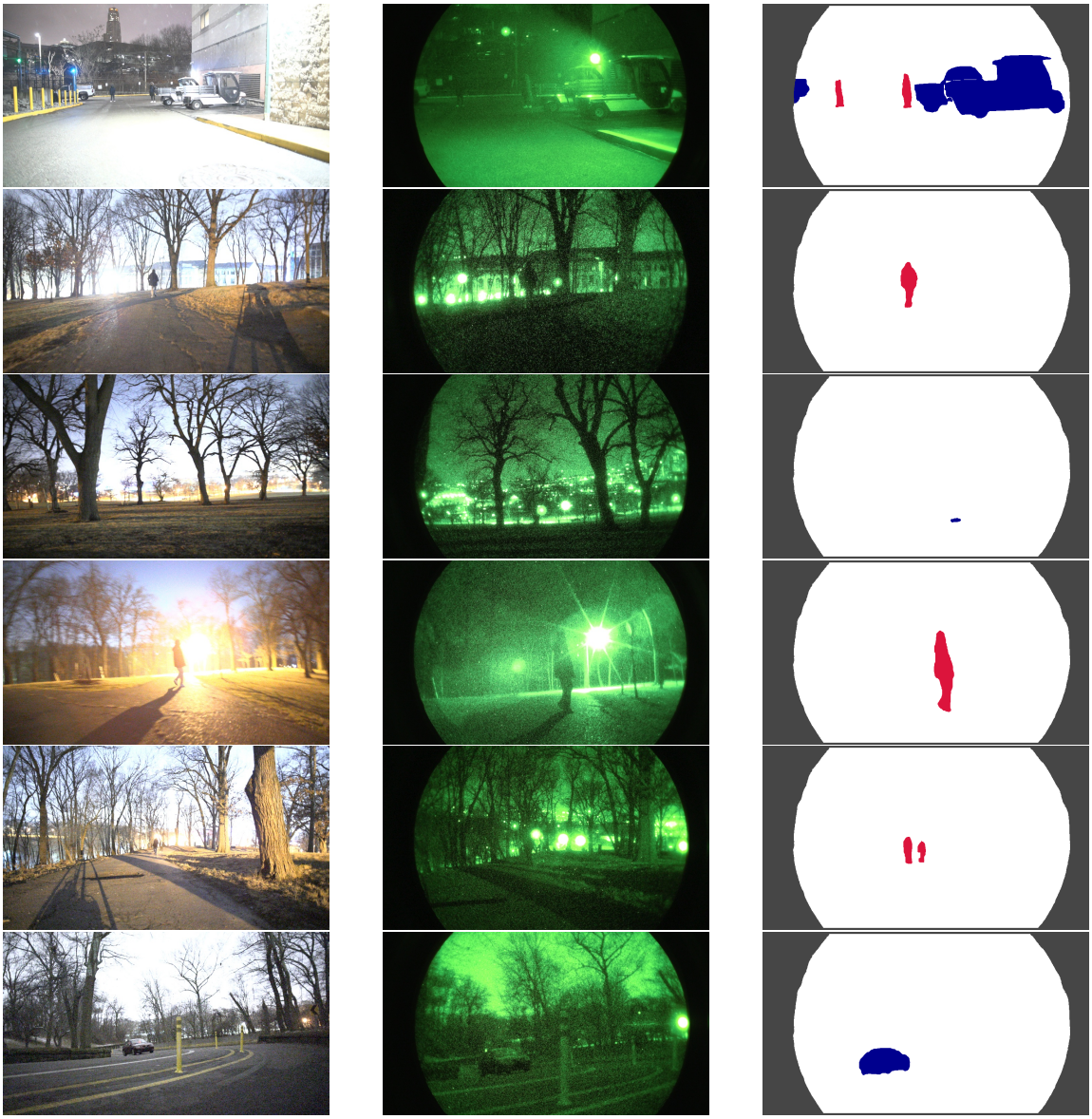

Figure 5.3: Representative examples from PittIntensified, with corresponding segmentation labels, where blue is used to represent the 'vehicle' class, red to represent the 'people' class, *white* to represent the background class, and gray to represent the ignore label, which corresponds to a fixed area in the image blocked by the intensifier module.

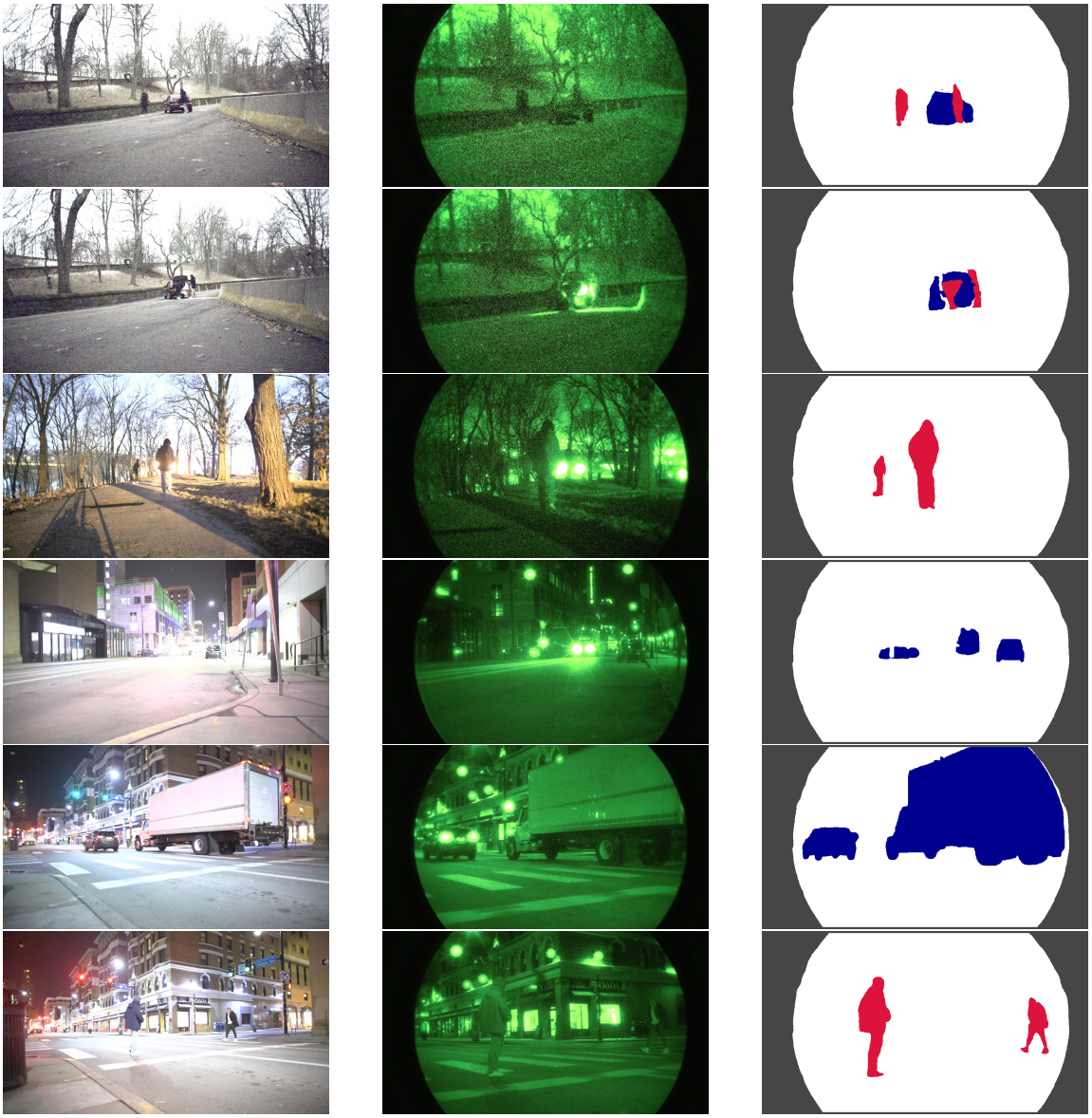

Figure 5.4: Representative examples from PittIntensified, with corresponding segmentation labels, where blue is used to represent the 'vehicle' class, red to represent the 'people' class, *white* to represent the background class, and gray to represent the ignore label, which corresponds to a fixed area in the image blocked by the intensifier module.

5.3 Quantitative Analysis

We manually refine the coarse annotations generated by Segment Anything [31] to provide semantic and instance-level labels for a subsampled set of 393 images from the intensifier camera. Figure 5.5 illustrates the number of pixels annotated per class, and the percentage of valid pixels belonging to each class.

As a part of our instance-level labels, we provide bounding-box annotations for people, and vehicles. We show their distribution over different sizes in Figure 5.6. Across all images, we have 241 bounding boxes corresponding to 'people' and 393 corresponding to 'vehicle'.

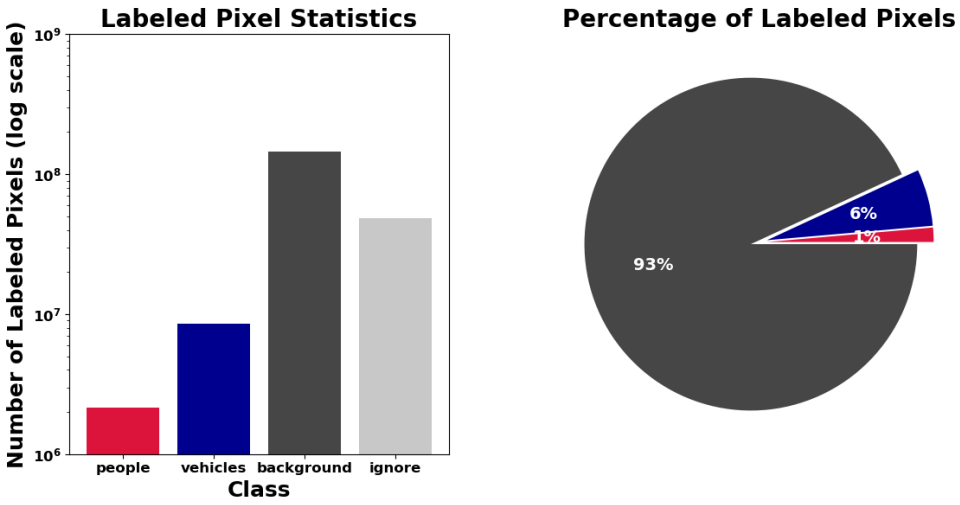

Figure 5.5: Number of annotated pixels in each labeled class in PittIntensified.

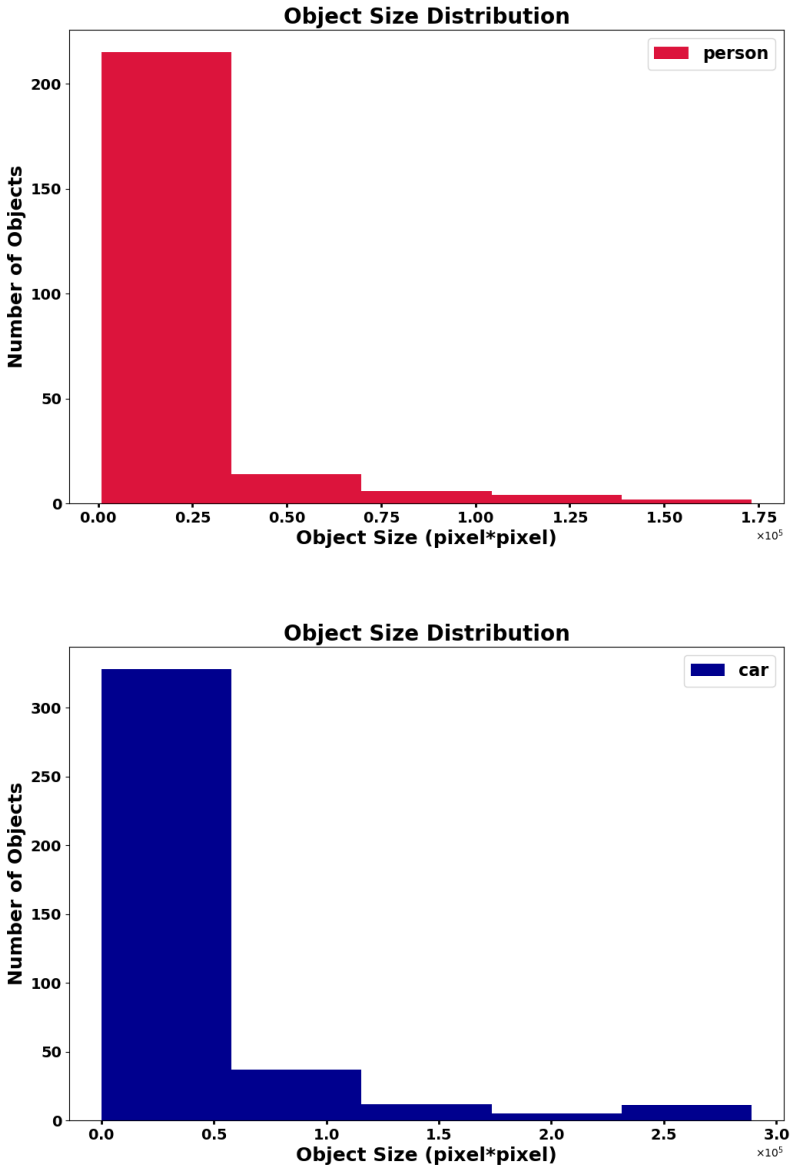

Figure 5.6: Number and distribution over sizes of annotated labels of objects with detection (bbox) and instance segmentation labels for people and vehicles. There are a total of 241 instances of the 'people' and 393 instances of the 'vehicle' class in the 393 labeled images of PittIntensified.

Chapter 6

Conclusions

In this work, we provide a pathway for designing robots that can operate in new visual domains, across different tasks of varying difficulty in labeling. While our work is directed towards operating artificial agents at night, it can be extended to other new environments as well.

First, we extend the operating range of an object detection system to enable on-robot low-light operations. This corresponds to a scenario where obtaining labels in the target domain is difficult, but feasible. We use this fact to train and deploy a very strong object detection system for low-light vision, thus extending the operating range of our robot to function 24/7.

Second, for the more challenging scenarios in which our ability to generate labels in new target domains can be strictly limited, we introduce a new framework for *effective, label-efficient Semi-Supervised Domain Adaptation*, called **Almost Unsupervised Domain Adaptation**. This comprises a novel preparatory step called **Source Preparation**, a method to account for source domain-specific characteristics and enhance Domain Adaptation by preparing an 'adaptable' source model, Unsupervised Domain Adaptation, and limited Supervised Alignment.

Source Preparation improves the performance of models across diverse domains, while also improving robustness to real-world shifts within each domain. Through Supervised Alignment we account for robotic scenarios, such as off-road environments in our PittIntensified dataset, where limited labeled target data can be obtained, and should be used.

We demonstrate our approach in improving domain adaptation across varying domains across time-lighting and modality, in scenarios with and without limited labeled target domain data during training on the representative task of semantic segmentation.

By providing a solution for two critical tasks in visual robotics in new target domains, which correspond to tasks of varying difficulty in label generation, we provide a pathway for how these tasks and other similar tasks could be solved in the future.

Limitations and Future Work. While we propose some design principles for designing new Source Preparation techniques, automatically learning or selecting the optimal source preparation technique from the data itself remains an open challenge. While our work shows great promise in domain adaptation to challenging new domains, the pursuit of creating more adaptable features continues. Further exploration in multi-modal joint-learning setups could be promising in this regard.

Bibliography

[1] Joseph L Baxter, EK Burke, Jonathan M Garibaldi, and Mark Norman. Multi-robot search and rescue: A potential field based approach. In *Autonomous robots and agents*, 2007. 3

[2] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 2010. 1

[3] Randolph Blake. The visual system of the cat. *Perception & Psychophysics*, 1979. 1

[4] Gilles Blanchard, Gyemin Lee, and Clayton Scott. Generalizing from several related classification tasks to a new unlabeled sample. In *NeurIPS*, 2011. 2.2

[5] David Brüggemann, Christos Sakaridis, Prune Truong, and Luc Van Gool. Refign: Align and refine for adaptation of semantic segmentation to adverse conditions. In *WACV*, 2023. 1, 2.2, 4.1.4, 4.2, 4.2.2

[6] Shuaijun Chen, Xu Jia, Jianzhong He, Yongjie Shi, and Jianzhuang Liu. Semi-supervised domain adaptation based on dual-level domain mixing for semantic segmentation. In *CVPR*, 2021. 1, 1, 2.1, 4.1.5

[7] Ying Chen, Xu Ouyang, Kaiyue Zhu, and Gady Agam. Semi-supervised dual-domain adaptation for semantic segmentation. *ICPR*, 2022. 1, 1, 2.1, 4.1.5

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 2.1, 4.2, 4.2.3

[9] Dengxin Dai and Luc Van Gool. Dark model adaptation: Semantic image segmentation from daytime to nighttime. In *ITSC*, 2018. 2.3

[10] Daniel S Drew. Multi-agent systems for search and rescue applications. In *Current Robotics Reports*, 2021. 3

[11] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, 2015. 1

[12] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 2016. 2.2

[13] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, 2016. 4.1.3.1

[14] Faine Greenwood, Erica L Nelson, and P Gregg Greenough. Flying into the hurricane: A case study of uav use in damage assessment during the 2017 hurricanes in texas and florida. In *PLoS one*, 2020. 3

[15] Licong Guan and Xue Yuan. Iterative loop method combining active and semi-supervised learning for domain adaptive semantic segmentation. *arXiv preprint arXiv:2301.13361*, 2023. 1

[16] Qishen Ha, Kohei Watanabe, Takumi Karasawa, Yoshitaka Ushiku, and Tatsuya Harada. Mfnet: Towards real-time semantic segmentation for autonomous vehicles with multi-spectral scenes. In *IROS*, 2017. 2.3, 4.2, 4.2.2

[17] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. In *arXiv preprint arXiv:1612.02649*, 2016. 2.2

[18] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. CyCADA: Cycle-consistent adversarial domain adaptation. In *ICML*, 2018. 2.2

[19] HoweAndHowe. Grizzly, 2019. 3.1

[20] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. Hrda: Context-aware high-resolution domain-adaptive semantic segmentation. In *ECCV*, 2022. 2.2

[21] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. Daformer: Improving network architectures and training strategies for domain-adaptive semantic segmentation. In *CVPR*, 2022. 2.2

[22] Lukas Hoyer, Dengxin Dai, Haoran Wang, and Luc Van Gool. Mic: Masked image consistency for context-enhanced domain adaptation. In *CVPR*, 2023. 1, 2.2

[23] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. 4.1.3.1

[24] Xin Jin, Cuiling Lan, Wenjun Zeng, and Zhibo Chen. Style normalization and restitution for domain generalization and adaptation. *IEEE Transactions on Multimedia*, 2021. 2.2

[25] Glenn Jocher. Yolov5 by ultralytics, 2020. URL https://github.com/ultralytics/yolov5. 2.4, 3.2

[26] Glenn Jocher. Yolov8 by ultralytics, 2023. URL https://github.com/

ultralytics/ultralytics. 2.4

[27] Alexander B. Jung, Kentaro Wada, Jon Crall, Satoshi Tanaka, Jake Graving, Christoph Reinders, Sarthak Yadav, Joy Banerjee, Gábor Vecsei, Adam Kraft, Zheng Rui, Jirka Borovec, Christian Vallentin, Semen Zhydenko, Kilian Pfeiffer, Ben Cook, Ismael Fernández, François-Michel De Rainville, Chi-Hung Weng, Abner Ayala-Acevedo, Raphael Meudec, Matias Laporte, et al. imgaug. https://github.com/aleju/imgaug, 2020. 4.2.1.1

[28] Tarun Kalluri, Girish Varma, Manmohan Chandraker, and CV Jawahar. Universal semi-supervised semantic segmentation. In *ICCV*, 2019. 2.1, 4.2.2

[29] Myeongjin Kim and Hyeran Byun. Learning texture invariant representation for domain adaptation of semantic segmentation. In *CVPR*, 2020. 2.2

[30] Soohyun Kim, Jongbeom Baek, Jihye Park, Gyeongnyeon Kim, and Seungryong Kim. Instaformer: Instance-aware image-to-image translation with transformer. In *CVPR*, 2022. 4.1.3.1

[31] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023. 5.1, 5.3

[32] Dong-Hyun Lee. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. *ICML Workshop*, 2013. 2.2

[33] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Demystifying neural style transfer. In *IJCAI*, 2017. 4.1.3.1

[34] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. In *CVPR*, 2019. 2.2

[35] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 3.2

[36] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. Deep learning for generic object detection: A survey. *IJCV*, 2020. 2.4

[37] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016. 2.4

[38] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. *NeurIPS*, 2016. 1

[39] Wenyu Lv, Shangliang Xu, Yian Zhao, Guanzhong Wang, Jinman Wei, Cheng Cui, Yuning Du, Qingqing Dang, and Yi Liu. Detrs beat yolos on real-time

object detection. *arXiv preprint arXiv:2304.08069*, 2023. 2.4

[40] Saeid Motiian, Quinn Jones, Seyed Iranmanesh, and Gianfranco Doretto. Few-shot adversarial domain adaptation. In *NeurIPS*, 2017. 1, 2.1

[41] Keiji Nagatani, Yoshito Okada, Naoki Tokunaga, Seiga Kiribayashi, Kazuya Yoshida, Kazunori Ohno, Eijiro Takeuchi, Satoshi Tadokoro, Hidehisa Akiyama, Itsuki Noda, et al. Multirobot exploration for search and rescue missions: A report on map building in robocuprescue 2009. In *Journal of Field Robotics*, 2011. 3

[42] Eric A Newman and Peter H Hartline. Integration of visual and infrared information in bimodal neurons in the rattlesnake optic tectum. *Science*, 1981. 1

[43] NVIDIA. Nvidia a100 gpu. https://www.nvidia.com/en-us/data-center/a100/. 4.2.6

[44] Francesco Pinto, Harry Yang, Ser-Nam Lim, Philip H. S. Torr, and Puneet K. Dokania. Regmixup: Mixup as a regularizer can surprisingly improve accuracy and out distribution robustness. In *NeurIPS*, 2023. 4.1.3.2

[45] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 2.4

[46] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, 2018. 1

[47] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Map-guided curriculum domain adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation. *TPAMI*, 2020. 1, 2.2, 2.3, 4.2, 4.2.6

[48] Connor Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 2019. 4.1.3.3

[49] Ankit Singh. Clda: Contrastive learning for semi-supervised domain adaptation. *NeurIPS*, 2021. 1, 2.1

[50] Antonio Tavera, Fabio Cermelli, Carlo Masone, and Barbara Caputo. Pixel-by-pixel cross-domain alignment for few-shot semantic segmentation. In *WACV*, 2022. 1, 1, 2.1, 4.1.5, 4.2.2

[51] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *ICCV*, 2019. 2.4

[52] 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. In *CVPR*, 2018. 2.2

[53] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in

semantic segmentation. In *CVPR*, 2019. 2.2

[54] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *CVPR*, 2023. 2.4

[55] Haoran Wang, Tong Shen, Wei Zhang, Ling-Yu Duan, and Tao Mei. Classes matter: A fine-grained adversarial approach to cross-domain semantic segmentation. In *ECCV*, 2020. 2.2

[56] Garrett Wilson and Diane J Cook. A survey of unsupervised deep domain adaptation. *TIST*, 2020. 2.2

[57] Xinyi Wu, Zhenyao Wu, Hao Guo, Lili Ju, and Song Wang. Dannet: A one-stage domain adaptation network for unsupervised nighttime semantic segmentation. In *CVPR*, 2021. 1, 2.2

[58] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *NeurIPS*, 2021. (document), 4.2, 4.1.3, 4.2, 4.2.2

[59] Jeongbeen Yoon, Dahyun Kang, and Minsu Cho. Semi-supervised domain adaptation via sample-to-sample self-distillation. In *WACV*, 2022. 1, 2.1

[60] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *CVPR*, 2017. 4.2.2

[61] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *CVPR*, 2020. 2.3

[62] Junkun Yuan, Xu Ma, Defang Chen, Kun Kuang, Fei Wu, and Lanfen Lin. Domain-specific bias filtering for single labeled domain generalization. *IJCV*, 2023. 2.2

[63] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *ICLR*, 2017. 4.1.3.2

[64] Junyi Zhang, Ziliang Chen, Junying Huang, Liang Lin, and Dongyu Zhang. Few-shot structured domain adaptation for virtual-to-real scene parsing. In *ICCV-Workshop*, 2019. 1, 2.1, 4.1.5

[65] Junyi Zhang, Ziliang Chen, Junying Huang, Liang Lin, and Dongyu Zhang. Few-shot structured domain adaptation for virtual-to-real scene parsing. In *ICCV*, 2019. 2.1

[66] An Zhao, Mingyu Ding, Zhiwu Lu, Tao Xiang, Yulei Niu, Jiechao Guan, and Ji-Rong Wen. Domain-adaptive few-shot learning. In *WACV*, 2021. 1, 2.1

[67] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. In *ICLR*, 2021. 4.1.3, 4.1.3.1

[68] Yang Zou, Zhiding Yu, B. V. K. Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*, 2018. 2.2