Learning-Based Perception for Robotics in Suboptimal Data Landscapes

Thesis by
Connor Tinghan Lee

In Partial Fulfillment of the Requirements for the
Degree of
Doctor of Philosophy in Space Engineering

Caltech

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2024
Defended May 22, 2024

ACKNOWLEDGEMENTS

I'd like to thank my advisor, Dr. Soon-Jo Chung, whose belief in my potential and unwavering support brought me back to Caltech to embark on this path. Dr. Chung fully supported every one of my various research goals throughout the years and has been instrumental in shaping my research direction and fostering my growth as a roboticist.

Next, I'd like to extend my thanks to the professors who provided guidance and support throughout my academic journey. I thank Dr. Georgia Gkioxari for her guidance during the final stages of my PhD. Her eagerness to share her computer vision insights and perspectives have been pivotal in refining my research. I also thank Dr. Fred Hadaegh and Dr. Michael Watkins for their service on my defense committee and for providing insightful feedback on my work. Lastly, I thank Dr. Joel Burdick, Dr. Pietro Perona, and Dr. Larry Matthies, whose support during my undergraduate years at Caltech helped cultivate my passion for robotics and computer vision.

Throughout my graduate studies, I have been fortunate to have admirable mentors who imparted not only their scientific wisdom, but also their unique traits. Dr. Anthony Fragoso has been a beacon of guidance since my time as an undergraduate researcher at NASA Jet Propulsion Laboratory. Among his many wisdoms, he taught me to pursue my ideas with unwavering confidence and to boldly push past challenges. Dr. Lu Gan enriched our endeavors in robot perception with her innovative ideas and supported me towards my research and career goals. Above all, she taught me how to execute research plans swiftly and decisively in the face of uncertainty. Lastly, Dr. Matthew Anderson played a pivotal role in developing the robotics hardware that brought my perception algorithms to life. His passion for robotics and eagerness to assist me in tackling unfamiliar challenges inspired me to become a more versatile researcher.

Collaboration has been a cornerstone of my research and I thank my collaborators, including Nikhil Raganathan, Dr. Xingxing Zuo, and Dr. Anthony Clark, whose collective efforts have enriched my projects. I extend my appreciation to all my undergraduate collaborators I've had the privilege of working with. Lastly, I'd like to thank my external collaborators throughout the years from Aerovironment, Boeing, Ford, SRI, Office of Naval Research, US Naval Research Laboratory, and

the US Army Engineer Research and Development Center, for their support and contributions to my research endeavors.

My academic journey at Caltech has been enriched by the vibrant community of researchers and scholars with whom I’ve had the privilege of interacting. I thank my lab mates in the Autonomous Robotics and Control Lab and my fellow 2018 GALCIT cohort. Your camaraderie and shared pursuit of knowledge have made my time in graduate school truly memorable.

Lastly, I am deeply thankful to my friends and family for their support and encouragement throughout my academic journey. My parents Lawrence and Joyce have been always been supportive and encouraging during my time at Caltech. To Marta: you motivate me to be excellent, and I am thankful for the countless adventures and moments we’ve shared along the way.

ABSTRACT

Autonomous robots are increasingly present in the world today, being used across a variety of settings and applications. In order to interact with their surroundings, robots typically use cameras to see the world, employing computer vision algorithms to comprehend rich, visual information. While contemporary, learning-based computer vision models provide robots with an accurate and robust understanding of their surroundings, most off-the-shelf methods rely on supervised deep learning techniques, requiring abundant labeled data in order to train and prevent overfitting. However, in many robotic applications and settings, the data landscape is characterized by data scarcity and/or the lack of apparent supervisory signals. Since custom perception solutions are often required for robotic applications, direct adoption of common computer vision methods proves challenging.

In this thesis, we develop robotic perception approaches across three different applications that overcome the challenges of such data landscapes. First, we develop learning-based visual terrain-relative navigation (VTRN) approaches for high-altitude aerial vehicles. This is a problem for which relevant data is available, but made difficult by the lack of obvious supervisory signals related to the high-level navigation objective. In the first chapters of the thesis, we show the power of self-supervised learning approaches to increase VTRN robustness to seasonal and temporal variations that would otherwise debilitate such systems.

Next, we address the challenge of developing thermal semantic perception algorithms for aerial field robotics. Due to the specialized nature of field environments and the sensing modality, development of thermal vision algorithms under these conditions is often characterized by the lack of relevant data. We show how we develop various thermal semantic segmentation in response to the evolving data constraints inherent in field robotic projects. In the final part of the thesis, we develop data-efficient, multispectral deep learning algorithms for autonomous driving applications where the lack of data arises from the need for custom, multispectral datasets that are synchronized and coregistered.

PUBLISHED CONTENT AND CONTRIBUTIONS

[1] C. Lee, S. Soedarmadji, M. Anderson, A. Clark, and S.-J. Chung. “Semantics from Space: Satellite-Guided Thermal Semantic Segmentation Annotation for Aerial Field Robots”. In: *arXiv preprint arXiv:2403.08997* (2024). Available at `https://arxiv.org/abs/2403.14056`.
C.L. conceived the project, developed the algorithm, implemented the software, conducted experiments, and wrote the manuscript.

[2] C. Lee*, M. Anderson*, N. Raganathan, X. Zuo, K. Do, G. Gkioxari, and S.-J. Chung. “CART: Caltech Aerial RGB-Thermal Dataset in the Wild”. In: *arXiv preprint arXiv:2403.08997* (2024). Available at `https://arxiv.org/abs/2403.08997`.
C.L., M.A. contributed equally to this work. C.L. led the project and the data annotation efforts, conducted the experiments and benchmarking, and wrote the manuscript.

[3] S. A. Deevi*, C. Lee*, L. Gan*, S. Nagesh, G. Pandey, and S.-J. Chung. “RGB-X Object Detection via Scene-Specific Fusion Modules”. In: *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2024, pp. 7351–7360. DOI: `10.1109/WACV57701.2024.00720`.
C.L., S.D., and L.G. contributed equally to this work. C.L. contributed to algorithm development, conducted experiments, and participated in the writing of the manuscript.

[4] L. Gan, C. Lee, and S.-J. Chung. “Unsupervised RGB-to-Thermal Domain Adaptation via Multi-Domain Attention Network”. In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. 2023, pp. 6014–6020. DOI: `10.1109/ICRA48891.2023.10160872`.
C.L. contributed to algorithm development, conducted experiments, and participated in the writing of the manuscript.

[5] C. Lee, J. G. Frennert, L. Gan, M. Anderson, and S.-J. Chung. “Online Self-Supervised Thermal Water Segmentation for Aerial Vehicles”. In: *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2023, pp. 7734–7741. DOI: `10.1109/IROS55552.2023.10342016`.
C.L. conceived the project, developed the algorithm, implemented the software, conducted experiments, and wrote the manuscript.

[6] C. Lee, E. Mesic, and S.-J. Chung. “Self-Supervised Landmark Discovery for Terrain-Relative Navigation”. In: *ICRA 2023 Workshop on Unconventional spatial representations: Opportunities for robotics*. Available at `https://usr2023.github.io/papers/Landmark.pdf`. 2023.
C.L. conceived the project, developed the algorithm, implemented the software, conducted experiments, and wrote the manuscript.

[7] A. Fragoso, C. Lee, A. McCoy, and S.-J. Chung. “A seasonally invariant deep transform for visual terrain-relative navigation”. In: *Science Robotics* 6.55 (2021). DOI: 10.1126/scirobotics.abf3320.
C.L. contributed to algorithm development, implemented the software, conducted the experiments, and participated in the writing of the manuscript.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Number *Page*

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 Motivation

Over the past three decades, autonomous robots have surged in prominence, evolving from guided cruise missiles in the late 20th century to the forefront of embodied artificial intelligence (AI) initiatives in 2024. Today, they play pivotal roles across diverse domains, from scientific endeavors like space exploration [18], subterranean exploration [32, 14], and coastal mapping [6] to commercial applications such as precision agriculture [27], infrastructure inspection, drone delivery, and the burgeoning realm of self-driving cars. Additionally, they serve critical functions in search and rescue missions and military operations. This pervasive integration of robotics in society underscores their versatility and potential to enhance our world.

Central to the operation of these autonomous systems is their ability to perceive and interpret their surroundings. Imaging sensors, specifically cameras, serve as one of the primary mechanisms for robots to sense the physical world. When coupled with advanced algorithms for processing and interpreting visual data, robots can localize themselves, navigate environments, and execute complex tasks effectively.

Cameras can also come in multiple modalities, operating across the different wavelengths of the electromagnetic spectrum. In settings where conventional visual (RGB) cameras face challenges such as low light or adverse weather conditions, alternative modalities offer advantages. Long-wave infrared (thermal) cameras excel in detecting heat signatures and are invaluable for enabling autonomy in darkness and adverse weather conditions like snow and fog [16]. In contrast, other sensors like lidar and radar offer 3D measurement capabilities which are commonly used for 3D scene perception for autonomous driving [15] and terrain contour matching for missile guidance [17]. However, they lack the information density and richness that cameras provide [15], and are less useful in situations where high spatial precision is desired.

As such, different imaging sensors provide complementary data streams, enabling robots to navigate and make decisions robustly in wildly diverse environments. Consequently, the development of computer vision algorithms capable of leveraging different data streams is imperative. Such algorithms enable robots to interact

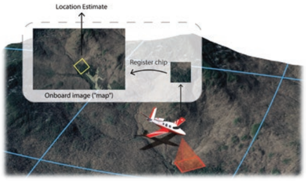
Application Area / Setting	Robust Visual Navigation for Uninhabited Aerial Systems (UAS)	Thermal Semantic Perception for Aerial Field Robotics in Littoral Environments	RGB-T Deep Sensor Fusion for Autonomous Driving
			
Problem	Visual terrain-relative navigation (VTRN) systems are crucial for precision navigation in GPS-denied settings, but struggle with seasonal differences in imagery.	Thermal imaging is useful for autonomy in low-light, but there are no relevant field datasets that can be used to develop such algorithms.	Multimodal deep sensor fusion models offer robust perception across diverse conditions but need large, coregistered datasets to develop.
Data Landscape + Constraints	• Terabytes of raw, high-res. aerial imagery • Lack of clear supervisory signal for objective	• Lack of relevant thermal datasets • Difficult to collect diverse training data due to red tape and geographic distribution	• Scarcity of coregistered RGB-T datasets • Lack of annotations for target objectives
Main Contributions	1. A deep transform that injects seasonal invariance into existing registration-based VTRN methods 2. A method to discover and re-identify seasonally-invariant landmarks for more compute-efficient localization.	1. An online adaptation method for water segmentation that does not need thermal data prior to test time. 2. A general RGB-T domain adaptation method that trains via labeled RGB and unlabeled thermal data collected from the field. 3. Caltech Aerial RGB-Thermal Dataset: the first RGB-T dataset tailor-made to create field robotic perception algorithms. 4. A fast and free method to auto-generate semantic segmentation annotations for aerial imagery using satellite data.	1. An RGB-T deep fusion framework that leverages pretrained single-modality networks to mitigate overfitting and speed up fusion training time.

Figure 1.1: This thesis focuses on developing and improving learning-based visual perception for robotics in three main application areas and settings, with each characterized by unique data landscapes that constrain the possible avenues for machine learning methods.

seamlessly in environments and fulfill their designated roles efficiently. In this work, we focus on developing learning-based perception solutions using rich image data from cameras, specifically within the RGB and thermal modalities.

Contemporary computer vision algorithms heavily rely on deep learning techniques. Unlike traditional computer vision methods that depend on handcrafted features and techniques, deep learning methods can easily scale to handle various settings through data-driven approaches [20, 13, 31, 35]. Consequently, deep learning methods require less manual tuning while being capable of handling different settings effectively. As such, learning-based methods are quite valuable for robot perception, allowing perception systems to generalize and perform well across different environments. An example where learning-based methods significantly improve robot perception is feature detection, which is commonly used for localization. Here, learned methods like SuperPoint [13] and R2D2 [30] can enable good feature matching in low-light conditions, across modalities [2], and in other settings where traditional feature detectors fail. This ability to easily handle varied situations makes learning-based approaches vital for robot perception.

Learning-based robotic perception methods typically harness the power of deep neural networks like convolutional neural networks (CNNs) and transformers to address visual tasks such as object detection, semantic segmentation, local feature detection, and place recognition [8, 29, 34]. Traditionally, these networks are trained using fully supervised learning, where labeled image data serve as training inputs. During training, the network processes images and compares its outputs against ground truth labels, optimizing network weights through backpropagation. However, this approach requires large amounts of labeled training data to prevent overfitting. In scenarios characterized by suboptimal data landscapes, conventional methodologies for developing vision algorithms falter. Examples of such conditions include data scarcity, lack of annotations, and lack of apparent supervisory signals.

In this thesis, we develop computer vision algorithms for robot perception across three different applications and settings, with each facing unique data constraints as outlined in Fig. 1.1:

1. Robust visual navigation for uninhabited aerial systems
2. Thermal semantic perception for aerial field robots
3. RGB-thermal deep sensor fusion for self-driving cars

All three settings that we focus on are unified by the challenge of navigating suboptimal data landscapes that prevent adoption of conventional fully-supervised approaches. In the rest of this chapter, we briefly introduce these main problem settings along with our respective contributions for each application.

1.2 Robust visual navigation for uninhabited aerial systems

Uninhabited aerial systems (UAS) traditionally rely on Global Navigation Satellite Systems (GNSS) for navigation. However, in GNSS-denied environments or instances of GNSS failure, aerial robots must resort to a method known as terrain-relative navigation (TRN). At its core, TRN matches source data (3D terrain information, 2D imagery, etc...) captured from the robot against georeferenced data cached onboard in order to provide precise geolocation information. Notable real-world instances of TRN include terrain contour matching [17] and the Digital Scene

Matching Area Correlator [7] for cruise missile guidance and the Lander Vision System for planetary entry, descent, and landing (EDL) in the Mars 2020 mission [18]. TRN approaches like [7] and [18] that rely on visual image matching are specifically known as visual terrain relative navigation (VTRN). These approaches leverage onboard cameras to capture and match rich visual imagery against high-resolution imagery. As a result, VTRN methods offer higher localization precision than TRN approaches based on sensors like radar [17].

Existing VTRN approaches employ image registration techniques for data association, which typically fall into two categories: area-based template matching or local feature-based homography estimation [18]. While proven through consistent real-world deployments, these methods are susceptible to temporal variations between source and georeferenced target data, and may result in poor localization matches. In space exploration applications, such variations manifest as severe illumination changes. For terrestrial applications, they manifest as varying light conditions at different times of day, cloud cover, and seasonal ground coverage.

For this problem, we primarily focus on improving VTRN in terrestrial settings, aiming to mitigate matching errors arising from seasonal and lighting variations by applying learning-based, data-driven methods. While high-resolution image data is abundant for deep learning methods in this problem setting, the challenge lies in determining the suitable supervisory signals to enable training that optimizes our VTRN objectives.

Contributions

In Chapter 2, we present a simple yet effective method that can inject seasonal-invariance into any existing VTRN system and increase the robustness of their image registration backend for localization. Specifically, we propose a CNN-based image transform as a preprocessing step to strip away unique seasonal content in source and target imagery, bringing both to a common domain. As the ambiguity of this objective prevents humans from easily creating optimal labels to enable a fully-supervised approach, we propose a completely data-driven and self-supervised method, where the supervisory signal is derived from the way in which we present training image samples to the CNN. We show that conventional image registration-based VTRN methods exhibit increased aerial geolocalization performance in the midst of seasonal variations when used in conjunction with our proposed deep transformation.

While the previous method improves the robustness of existing VTRN methods, it still inherits the computational and storage costs of image registration backends due to the use of large onboard maps. In Chapter 3, we introduce a self-supervised landmark discovery algorithm that diverges from current VTRN approaches. Our approach departs from the traditional reliance on large, georeferenced maps, opting for a memory-efficient paradigm based on sparse, seasonally-invariant landmarks ideal for geolocalization across large areas. Unlike recent methods that rely on potentially biased and suboptimal human guidance for supervised learning [25, 36, 14] or those that completely forsake landmarks [3, 9], leading to limited navigable areas, we propose the problem of discovering optimal landmarks for vision-based navigation. We demonstrate that a data-driven, self-supervised method can also solve this landmark discovery problem.

1.3 Thermal semantic perception for nighttime aerial field robots

Field robots currently rely on visual cameras and deep neural networks for semantic scene perception [15]. While RGB cameras offer rich visual information, they struggle in low-light settings [33]. In contrast, thermal cameras can provide rich visual data in such conditions, but with lower resolution and less fine-grained details [16]. Although thermal cameras are being increasingly integrated in autonomous robots to enable nighttime autonomy [21, 39, 32, 26, 10, 24, 12], integration in field robotics is difficult due to the lack of large-scale thermal datasets covering field settings of interest.

For this problem, we aim to develop thermal semantic segmentation models to enable aerial robots to understand their surroundings and conduct autonomous operations at night. Specifically, we aim to develop learning-based perception algorithms that can enable operations in littoral settings, such as rivers and coastlines, to enable downstream scientific missions [5].

Contributions

In Chapters 4 – 7, we present the development of various thermal semantic perception algorithms for an aerial field robot, starting completely from scratch. The content in these chapters is organized chronologically, and reflects the particular data limitations we had during the corresponding phase of the project. Collectively, the chapters illustrate the evolution of field robotic perception algorithms from deployment-time adaptation approaches to conventional fully-supervised approaches as our data constraints relaxed over time. Furthermore, we showcase our dataset collection and curation process on the path towards enabling fully-supervised learning, and present a novel method that reduces the time and costs of creating semantic segmentation annotations for imagery collected using aerial robots. The chapters in this series are summarized as follows.

Chapter 4 marks the beginnings of the project where we had only enough annotated data to evaluate but not train semantic segmentation algorithms. In this chapter, we introduce an online self-supervised algorithm designed to adapt to thermal data during deployment, with a specific focus on water segmentation to enable downstream scientific objectives like bathymetry [6]. Specifically, we demonstrate how our network successfully learns from the amalgamation of weak and noisy heuristic-based signals, highlighting a pathway for perception development in the early stages of field robotics projects where offline training data is non-existent. Unlike other online self-supervised methods [1, 11], we also propose a computational framework that enables continual segmentation inference at 10 Hz on an embedded device, all while undergoing online training.

Chapter 5 marks the phase of the project in which we have collected a significant amount of thermal data which have not yet undergone manual annotation. In this chapter, we develop an unsupervised domain adaptation (UDA) method that leverages large-scale annotated RGB datasets with unlabeled thermal data to train a thermal multiclass semantic segmentation network. Like many existing works, our method aims to achieve RGB-T UDA via domain confusion of intermediate network features [19]. We propose a new method to achieve this by using domain-specific attention modules to align domain-invariant features while preventing forced alignment of modality-specific features. Our approach outperforms other UDA methods across two RGB-T domain adaptation experiments and due to its simplicity, it can be easily adapted for use across various deep learning tasks.

In Chapter 6, we detail our efforts to curate and annotate the thermal imagery we collected over the course of this project and present the first RGB-thermal (RGB-T) dataset targeted towards the development of perception algorithms for aerial field robotic settings. Using our curated dataset, we establish new benchmarks for thermal and RGB-thermal semantic segmentation, RGB-T image translation, and thermal visual-inertial odometry (VIO) and simultaneous localization and mapping (SLAM).

In Chapter 7, we introduce a novel approach that integrates onboard robotic sensors, satellite-derived data products, and visual foundation models to automatically annotate aerial thermal imagery for semantic segmentation. We validate our proposed method using the curated dataset from Chapter 6 and demonstrate superiority over current zero-shot vision-language foundation models that are used to annotate RGB imagery [22, 38]. We also demonstrate a fully-supervised learning approach that successfully utilizes our automatically generated labels to train a lightweight, semantic segmentation network. Lastly, our method provides a quick and low-cost method to generate semantic segmentation annotations for image data captured from an aerial robot and is agnostic to sensor modality.

1.4 RGB-thermal deep sensor fusion for self-driving cars

Although RGB and thermal cameras are effective when used individually, using them jointly in a process known as multimodal deep sensor fusion, if computational budget allows, improves perception robustness by collectively compensating for the weaknesses attributed to each individual sensor [15]. While such a process sounds complex, deep sensor fusion algorithms operate similarly to their single-modality counterparts: image inputs are encoded by a neural network encoder before being passed to a decoder and task-specific head to generate the desired task-specific output. However, in deep sensor fusion models, network representations of each input modality are combined, i.e. *fused*, at various points in the model in order to create multimodal representations [40, 23, 24]. Determining optimal areas for information fusion is still an active area of research.

Although deep sensor fusion algorithms usually exhibit higher robustness over single-modality counterparts, they face a stricter data constraint: The various data inputs used in the model must be coregistered or at minimum, have known spatial transformations. Consequently, as the number of input modalities increase, the pool of publicly-available candidate datasets for training such a model drops, increasing the likelihood of needing custom dataset curation as in Section 1.3. In this problem setting, we aim to move away from standard end-to-end training approaches [4, 40, 24, 23, 41, 28] and instead, develop a data-efficient deep sensor fusion algorithm for RGB-T object detection, with a specific focus on applications in autonomous vehicles such as self-driving cars.

Contributions

In Chapter 8, we propose an RGB-X objection detection model that leverages scene-specific fusion modules to fuse intermediate data representations. Specifically, we develop our method in order to take full advantage of pretrained single-modality models, using lightweight scene-specific convolutional block attention modules [37] to perform fusion in the latter stages of the network before passing the fused representations to an object detector head. During training, we only update the weights of the fusion modules, restricting the total trainable parameters of the model to a minimal amount. In contrast to end-to-end approaches, this strategy enables us to perform fusion training using small, coregistered multimodal datasets and to do so very rapidly. We demonstrate our approach on RGB-T and RGB-gated data and quantify its performance against existing works that perform resource-intensive end-to-end training.

References

[1] S. Achar, B. Sankaran, S. Nuske, S. Scherer, and S. Singh. "Self-supervised segmentation of river scenes". In: *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE. 2011, pp. 6227–6232.

[2] F. Achermann, A. Kolobov, D. Dey, T. Hinzmann, J. J. Chung, R. Siegwart, and N. Lawrance. "MultiPoint: Cross-spectral registration of thermal and optical aerial imagery". In: *Conference on Robot Learning*. PMLR. 2021, pp. 1746–1760.

[3] M. Bianchi and T. D. Barfoot. "Uav localization using autoencoded satellite images". In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 1761–1768.

[4] M. Bijelic, T. Gruber, F. Mannan, F. Kraus, W. Ritter, K. Dietmayer, and F. Heide. "Seeing Through Fog Without Seeing Fog: Deep Multimodal Sensor Fusion in Unseen Adverse Weather". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[5] K. L. Brodie, B. L. Bruder, R. K. Slocum, and N. J. Spore. "Simultaneous mapping of coastal topography and bathymetry from a lightweight multicamera UAS". In: *IEEE Transactions on Geoscience and Remote Sensing* 57.9 (2019), pp. 6844–6864.

[6] K. L. Brodie, B. L. Bruder, R. K. Slocum, and N. J. Spore. "Simultaneous Mapping of Coastal Topography and Bathymetry From a Lightweight Multicamera UAS". In: *IEEE Transactions on Geoscience and Remote Sensing* 57.9 (2019), pp. 6844–6864. DOI: 10.1109/TGRS.2019.2909026.

[7] J. R. Carr and J. S. Sobek. "Digital Scene Matching Area Correlator (DSMAC)". In: *Image Processing for Missile Guidance*. Ed. by T. F. Wiener. SPIE, 1980. DOI: 10.1117/12.959130.

[8] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. "Deeplab: Semantic image segmentation with deep convolutional nets, atrous

convolution, and fully connected crfs". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.4 (2017), pp. 834–848.

[9] S. Chen, X. Wu, M. W. Mueller, and K. Sreenath. "Real-time Geo-localization Using Satellite Imagery and Topography for Unmanned Aerial Vehicles". In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2021, pp. 2275–2281. doi: 10.1109/IROS51168.2021.9636705.

[10] Y. Choi, N. Kim, S. Hwang, K. Park, J. S. Yoon, K. An, and I. S. Kweon. "KAIST multi-spectral day/night data set for autonomous and assisted driving". In: *IEEE Transactions on Intelligent Transportation Systems* 19.3 (2018), pp. 934–948.

[11] S. Daftry, Y. Agrawal, and L. Matthies. "Online Self-Supervised Long-Range Scene Segmentation for MAVs". In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2018, pp. 5194–5199.

[12] J. Delaune, R. Hewitt, L. Lytle, C. Sorice, R. Thakker, and L. Matthies. "Thermal-inertial odometry for autonomous flight throughout the night". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2019, pp. 1122–1128.

[13] D. DeTone, T. Malisiewicz, and A. Rabinovich. "Superpoint: Self-supervised interest point detection and description". In: *Proc. IEEE Conf. Comput. Vis. Pattern Recog. workshops*. 2018, pp. 224–236.

[14] L. Downes, T. J. Steiner, and J. P. How. "Deep learning crater detection for lunar terrain relative navigation". In: *AIAA SciTech 2020 Forum*. 2020, p. 1838.

[15] D. Feng, C. Haase-Schütz, L. Rosenbaum, H. Hertlein, C. Glaeser, F. Timm, W. Wiesbeck, and K. Dietmayer. "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges". In: *IEEE Transactions on Intelligent Transportation Systems* 22.3 (2020), pp. 1341–1360.

[16] R. Gade and T. B. Moeslund. "Thermal cameras and applications: a survey". In: *Machine vision and applications* 25 (2014), pp. 245–262.

[17] J. P. Golden. "Terrain contour matching (TERCOM): a cruise missile guidance aid". In: *Image processing for missile guidance*. Vol. 238. SPIE. 1980, pp. 10–18.

[18] A. E. Johnson, S. B. Aaron, H. Ansari, C. Bergh, H. Bourdu, J. Butler, J. Chang, R. Cheng, Y. Cheng, K. Clark, et al. "Mars 2020 Lander Vision System Flight Performance". In: *AIAA SciTech 2022 Forum*. 2022, p. 1214.

[19] Y.-H. Kim, U. Shin, J. Park, and I. S. Kweon. “MS-UDA: Multi-Spectral Unsupervised Domain Adaptation for Thermal Image Semantic Segmentation”. In: *IEEE Robotics and Automation Letters* 6.4 (2021), pp. 6497–6504. DOI: 10.1109/LRA.2021.3093652.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems* 25 (2012).

[21] C. Li, W. Xia, Y. Yan, B. Luo, and J. Tang. “Segmenting objects in day and night: Edge-conditioned cnn for thermal image semantic segmentation”. In: *IEEE Transactions on Neural Networks and Learning Systems* 32.7 (2020), pp. 3069–3082.

[22] F. Liang, B. Wu, X. Dai, K. Li, Y. Zhao, H. Zhang, P. Zhang, P. Vajda, and D. Marculescu. “Open-vocabulary semantic segmentation with mask-adapted clip”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 7061–7070.

[23] M. Liang, J. Hu, C. Bao, H. Feng, F. Deng, and T. L. Lam. “Explicit Attention-Enhanced Fusion for RGB-Thermal Perception Tasks”. In: *IEEE Robotics and Automation Letters* (2023).

[24] J. Liu, X. Fan, Z. Huang, G. Wu, R. Liu, W. Zhong, and Z. Luo. “Target-aware Dual Adversarial Learning and a Multi-scenario Multi-Modality Benchmark to Fuse Infrared and Visible for Object Detection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2022, pp. 5802–5811.

[25] A. Nassar, K. Amer, R. ElHakim, and M. ElHelw. “A deep cnn-based framework for enhanced aerial imagery registration with applications to uav geolocalization”. In: *Proc. IEEE Conf. Comput. Vis. Pattern Recog. workshops*. 2018, pp. 1513–1523.

[26] S. Nirgudkar, M. DeFilippo, M. Sacarny, M. Benjamin, and P. Robinette. “MassMIND: Massachusetts Maritime INfrared Dataset”. In: *International Journal of Robotics Research* 42.1-2 (2023), pp. 21–32.

[27] A. Pretto, S. Aravecchia, W. Burgard, N. Chebrolu, C. Dornhege, T. Falck, F. Fleckenstein, A. Fontenla, M. Imperoli, R. Khanna, et al. “Building an aerial–ground robotics system for precision farming: an adaptable solution”. In: *IEEE Robotics and Automation Magazine* 28.3 (2020), pp. 29–49.

[28] F. Qingyun and W. Zhaokui. “Cross-modality attentive feature fusion for object detection in multispectral remote sensing imagery”. In: *Pattern Recognition* 130 (2022), p. 108786.

[29] S. Ren, K. He, R. Girshick, and J. Sun. “Faster r-cnn: Towards real-time object detection with region proposal networks”. In: *Advances in neural information processing systems* 28 (2015).

[30] J. Revaud, C. De Souza, M. Humenberger, and P. Weinzaepfel. “R2d2: Reliable and repeatable detector and descriptor”. In: *Advances in neural information processing systems* 32 (2019).

[31] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich. “Superglue: Learning feature matching with graph neural networks”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 4938–4947.

[32] S. S. Shivakumar, N. Rodrigues, A. Zhou, I. D. Miller, V. Kumar, and C. J. Taylor. “Pst900: Rgb-thermal calibration, dataset and segmentation network”. In: *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2020, pp. 9441–9447.

[33] J. Vargas, S. Alsweiss, O. Toker, R. Razdan, and J. Santos. “An overview of autonomous vehicles sensors and their vulnerability to weather conditions”. In: *Sensors* 21.16 (2021), p. 5397.

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. “Attention is all you need”. In: *Proceedings of the Advances in Neural Information Processing Systems Conference* 30 (2017).

[35] S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud. “DUSt3R: Geometric 3D Vision Made Easy”. In: *CVPR*. 2024.

[36] T. Wang, Y. Zhao, J. Wang, A. K. Somani, and C. Sun. “Attention-Based Road Registration for GPS-Denied UAS Navigation”. In: *IEEE Transactions on Neural Networks and Learning Systems* 32.4 (2021), pp. 1788–1800. DOI: 10.1109/TNNLS.2020.3015660.

[37] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon. “CBAM: Convolutional block attention module”. In: *Proceedings of the European Conference on Computer Vision*. 2018, pp. 3–19.

[38] J. Xu, S. Liu, A. Vahdat, W. Byeon, X. Wang, and S. De Mello. “Open-vocabulary panoptic segmentation with text-to-image diffusion models”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 2955–2966.

[39] S. Yun, M. Jung, J. Kim, S. Jung, Y. Cho, M.-H. Jeon, G. Kim, and A. Kim. “Sthereo: Stereo thermal dataset for research in odometry and mapping”. In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2022, pp. 3857–3864.

[40] H. Zhang, E. Fromont, S. Lefevre, and B. Avignon. “Multispectral fusion for object detection with cyclic fuse-and-refine blocks”. In: *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2020, pp. 276–280.

[41] H. Zhang, E. Fromont, S. Lefèvre, and B. Avignon. “Guided attentive feature fusion for multispectral pedestrian detection”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2021, pp. 72–80.

Chapter 2

A SEASONALLY INVARIANT DEEP TRANSFORM FOR VISUAL TERRAIN-RELATIVE NAVIGATION

[1] A. Fragoso, C. Lee, A. McCoy, and S.-J. Chung. “A seasonally invariant deep transform for visual terrain-relative navigation”. In: *Science Robotics* 6.55 (2021). DOI: 10.1126/scirobotics.abf3320.

2.1 Abstract

Visual Terrain-Relative Navigation (VTRN) is a localization method based on registering a source image taken from a robotic vehicle against a georeferenced target image. With high-resolution imagery databases of Earth and other planets now available, VTRN offers accurate, drift-free navigation for air and space robots even in the absence of external positioning signals. Despite its potential for high accuracy, however, VTRN remains extremely fragile to common and predictable seasonal effects, such as lighting, vegetation changes, and snow-cover. Engineered registration algorithms are mature and have provable geometric advantages, but cannot accommodate the content changes caused by seasonal effects and have poor matching skill. Approaches based on deep learning can accommodate image content changes, but produce opaque position estimates that either lack an interpretable uncertainty or require tedious human annotation. In this work, we address these issues with targeted use of deep learning within an image transform architecture, which converts seasonal imagery to a stable, invariant domain that can be used by conventional algorithms without modification. Our transform preserves the geometric structure and uncertainty estimates of legacy approaches and demonstrates superior performance under extreme seasonal changes, while also being easy to train and highly generalizable. We show that classical registration methods perform exceptionally well for robotic visual navigation when stabilized with the proposed architecture, and are able to consistently anticipate reliable imagery. Gross mismatches were nearly eliminated in challenging and realistic visual navigation tasks that also included topographic and perspective effects.

2.2 Introduction

Remotely-sensed database imagery is a common ground-truth map for Visual Terrain-Relative Navigation (VTRN). Onboard cameras are passive sensors ideal for size-, weight-, and power-constrained platforms, and extensive coverage of high-resolution imagery makes VTRN essential in the absence of Global Navigation Satellite System (GNSS) capability. Database imagery has been used to provide absolute position measurements in extraterrestrial robotic entry, descent, and landing (EDL) missions [20, 10], GNSS-denied defense applications [8], backup unmanned aerial vehicle (UAV) state estimation [9], and offline subpixel geolocation of remotely sensed data products that can be extremely sensitive to localization errors [28].

VTRN and geolocation against target images are applications of the more general image registration problem [19], in which images taken from different poses, at different times, or with different sensors are transformed into the same coordinate system. Under ideal conditions, image registration is well-studied and has a number of mature automatic solutions. Examples include intensity-based template matching with normalized cross-correlation (NCC) [24], mutual information (MI) similarity metrics [32], frequency-domain techniques [25], and feature matching [17]. Classical registration algorithms are also often equipped with geometric and radiometric invariances that greatly simplify the VTRN problem itself. For example, feature-based methods can accommodate non-rigid image transformations due to terrain, with scale-invariant feature transform (SIFT) descriptors [17] in particular being invariant to scale, 2-D rotation, and linear illumination changes.

In principle, an aerospace robot can be localized to within a few centimeters relative to a database using onboard imagery and a subpixel-accurate registration algorithm. High-quality georeferenced terrestrial imagery is updated on a regular basis and often available at resolutions of 10 centimeters per pixel or better—global coverage is available commercially at approximately 30 centimeters per pixel [16]. In practice, however, aerospace robots that rely on vision regularly encounter severe appearance changes, such as snow-cover or leaf-drop, that change the texture, illumination, and content of the landscape beneath them. These changes violate the heuristic radiometric assumptions of classical registration and lead to fragility. Indeed, manual selection and matching of structures and control points with a stable appearance remains a respected, if time-consuming practice for offline registration [4]. Autonomous platforms, however, must reliably perform matching without human intervention and have historically relied on comparing radar altimetry [11] to a topographic database. Although topographic data is more stable than visible data, terrain matching is less accurate than image registration, and exhibits poor performance at low altitudes or in flat areas [15].

In order to address the shortcomings of classical approaches, a natural option is to consider deep-learning approaches that fit stable, high-level features [35]. Although seasonal changes in aerial imagery have received minimal attention, some deep learning techniques have been successful for challenging fusion and registration tasks, particularly in medical imaging [13]. A common approach is to train a deep similarity metric to replace fragile classical metrics using a Siamese architecture [34]. For remote sensing, [14] learned a similarity score between synthetic aperture radar and optical images using a pseudo-Siamese architecture, with separate fully-convolutional networks for each fed into a common comparison network. Registration in this manner requires small rigid image *chip* patches to be repeatedly sampled from larger images and passed through the network consecutively, which precludes real-time use.

End-to-end networks accommodate non-rigid registration and avoid repeated evaluation of image chips by directly estimating the geometric transformation between two input images, but lack the engineered advantages of classical approaches. Pure end-to-end approaches require explicit exposure to all expected transformations in training, for which examples of real non-rigid transformations are exceptionally difficult to obtain. Reliability and uncertainty for end-to-end networks are also extremely difficult to interpret. Furthermore, deep learning does not necessarily outperform hand-engineered approaches for monomodal registraion, and is often complementary. [5] observed that monomodal registration of lung imagery under large deformations benefited from the hybrid use of hand-engineered and learning-based descriptors. Similarly, [33] augmented SIFT features with robust deep features taken from the intermediate activations of a pretrained VGG-16 network. Semantic features can also be extracted and matched using pixel-level segmentation, in which stable predefined structures such as road networks [12], lunar crater rims [10], or prescribed semantic elements [22] are labeled using a trained network and matched to a reference image. Segmentation techniques identify stable structures, but require extraordinarily tedious human annotation. Structure classes must also be unambiguous and consistently distributed in imagery to be useful, and are prescribed rather than themselves learned.

The registration robustness problem can also be posed using domain adaptation theory, in which a non-annotated target data domain supplements an annotated source domain. The source and target domains are assumed to share content relevant to a task but differ in their domain-specific statistics—for example, leaf-on and leaf-off imagery are both useful for navigation even though their vegetation patterns are different. Relevant content can be identified automatically by mapping two or more source domains to a common latent domain that is optimized to complete a task, but in which the original source domains can no longer be distinguished. The goal is usually to assimilate unlabeled data into a model as efficiently as possible, which has seen widespread use in remote sensing primarily to improve classifier robustness while limiting annotation load [29]. Automatic image registration, however, has received little attention in the domain adaptation literature despite its extreme sensitivity. The domain adaptation work most relevant to VTRN is based on image-to-image translation and comes from the automotive community, including adaptation for scene segmentation [21] and translation of degraded operating conditions into ordinary conditions [1].

Main Contribution

In light of these observations, we derive an image transform approach to VTRN that combines the success of deep learning for image translation and domain adaptation with the well-known engineered properties of classical image registration. Rather than attempt to generate an opaque positioning estimate using deep learning and extensive annotation, we rely on the fact that conventional registration techniques in principle have perfect performance when their radiometric input assumptions are exactly met. Accordingly, we use deep learning only to modify the appearance of input images, which is a narrowly-defined task at which it excels. The basis of our technique is a fully convolutional network (FCN) that serves as a preprocessing step and transforms input images to a seasonally-invariant domain. This network identifies and enhances stable structures, serves as an attention mechanism, and is optimized for robust performance over any well-posed classical registration algorithm. After sufficient training, a single transform allows leaf-on, leaf-off, and snow-cover test images to have an identical appearance and registration response, and can mitigate some higher-frequency appearance changes, such as deep shifting shadows, that were not explicitly anticipated or trained over in advance. The transform structure lends seasonal invariance to existing conventional code and techniques without further modification or manual annotation. The result is a VTRN pipeline that inherits

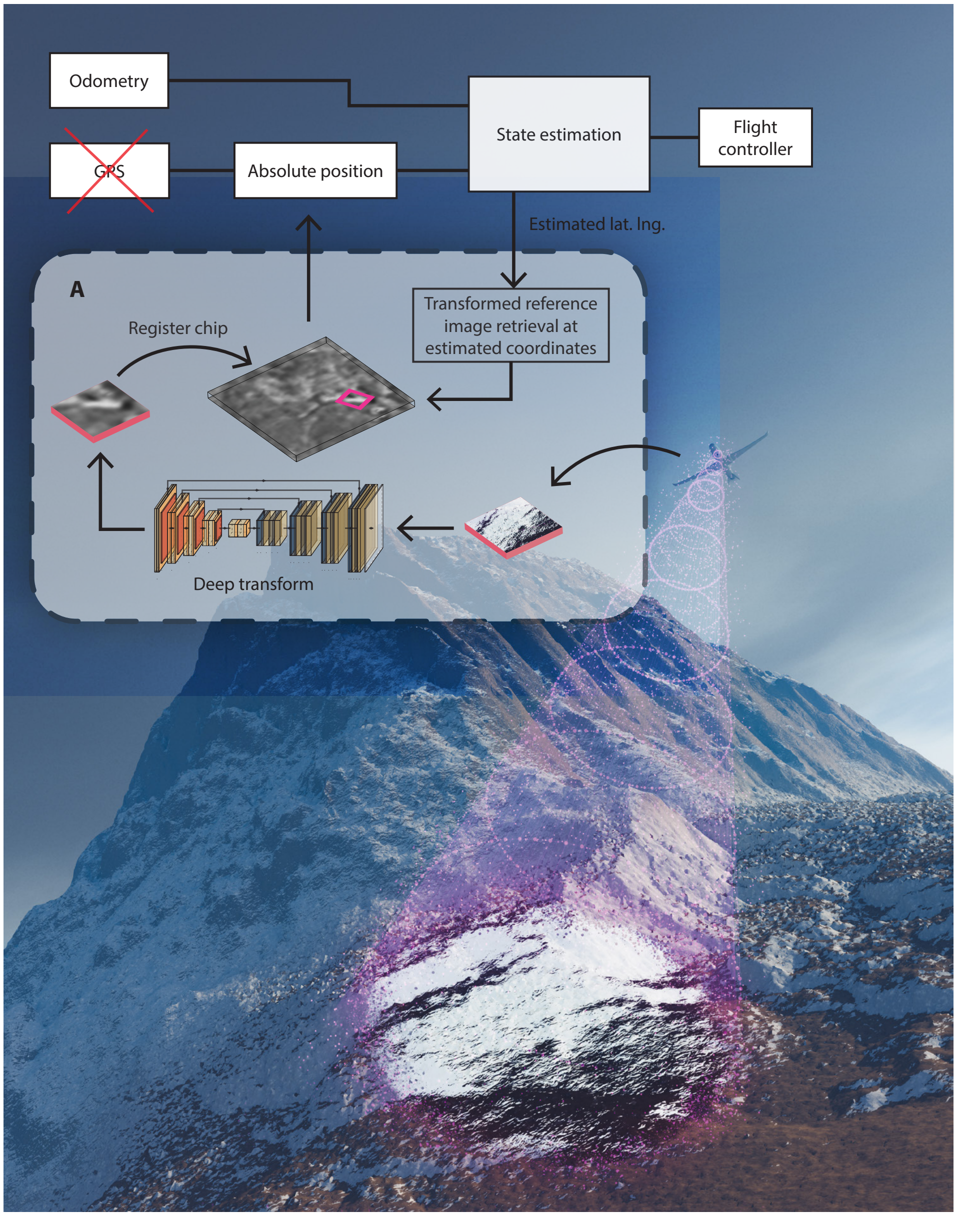


Figure 2.1: VTRN using the seasonally-invariant deep transform in a GNSS-denied environment. The UAV determines its absolute position by registering an online deep-transformed image of the ground into a previously deep-transformed georeferenced database image proposed using a running black-box odometry estimate. The deep transform module (**A**) removes ephemeral character from both images and forces them to satisfy the radiometric assumptions of a hand-engineered registration module that follows. As a result, the UAV can reliably recover its position from the geometric transformation between the two images using legacy techniques that fail otherwise.

geometric invariances from conventional registration without explicit exposure in training while also relaxing the widely-violated input assumptions that cause them to break.

2.3 Results

In this section, we demonstrate the effectiveness of our deep transform architecture for optimizing area-based and feature-based image registration in challenging seasonal conditions. After describing our architecture and the datasets used for training and testing, we provide experimental performance evaluation results.

Architecture

Our transform serves as an upstream preprocessing step that adds seasonal robustness to downstream VTRN or registration algorithms, which are themselves unmodified. The network is trained in advance using diverse seasonal imagery examples and deployed with locked weights, either aboard an aircraft or space robot (VTRN — Fig. 2.1) or for general image registration. Input and output image sizes are equal, but outputs are grayscale as is typically required for registration. We train our transform using publicly available data that is already co-registered and requires no further annotation. Suitable training imagery is widely available with global and extraterrestrial coverage at a high resolution, and captures years of regular appearance changes.

During training, we expose a U-Net [27] image transform model to matching cross-seasonal image pairs in Siamese fashion: a single transform is identically shared between two parallel streams, with registration performance between the outputs used as a loss function to optimize the transform weights (Fig. 2.2). The training process is discussed in more detail in Section 2.5.

At run-time, a single stream of incoming imagery, such as a navigation camera (NAVCAM) image or an unregistered scientific image product, is intercepted and replaced with transform output. This output is passed, along with a previously transformed reference image, to the rest of the registration pipeline to calculate the geometric transformation between the two images. For change detection and other scientific applications in which image appearance must be preserved, the original input may simply be warped by the now-known geometric transformation.

VTRN backend and reference image selection: The deep transform architecture is agnostic to the registration backend as long as the matching score is compatible

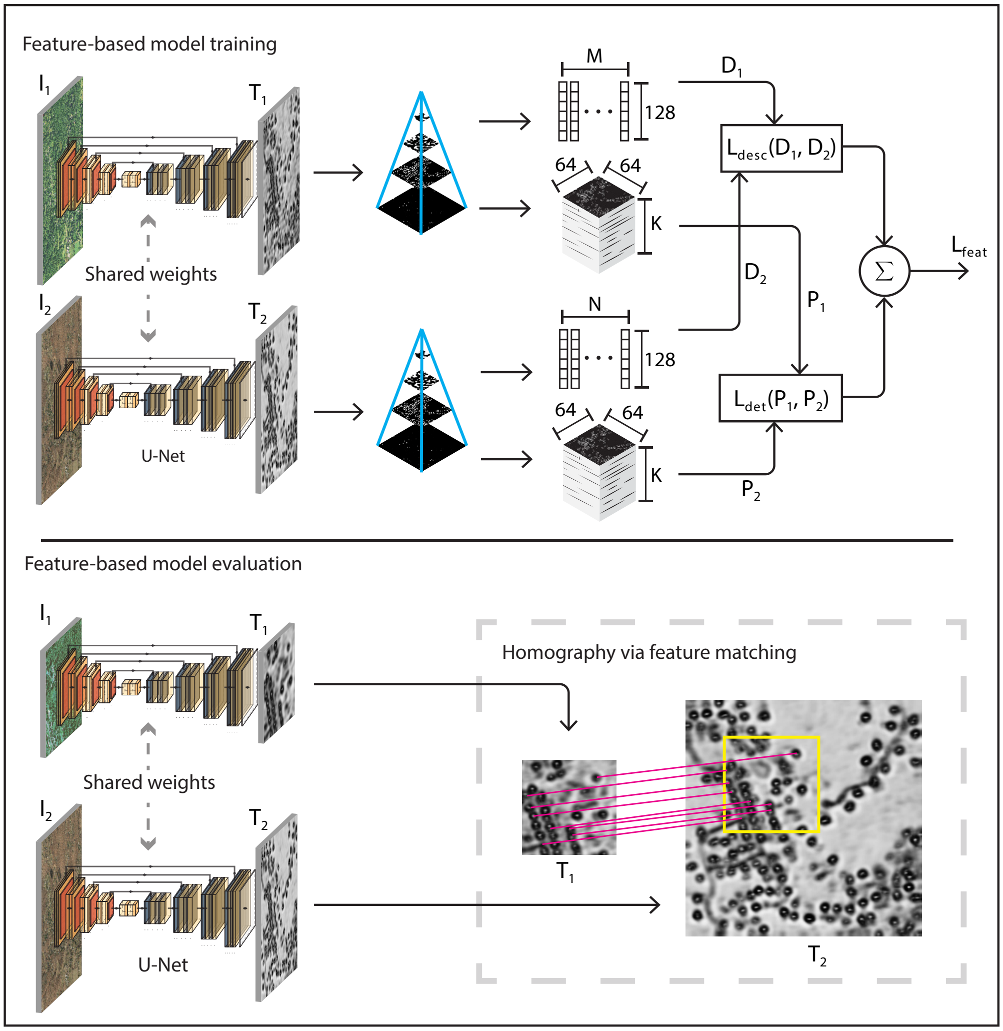


Figure 2.2: Training and evaluation pipeline for feature-based image registration. During training, the loss function guides the network to transform images such that corresponding keypoints align in scale and location and their feature descriptors match. In evaluation mode, the network serves as a preprocessing step to transform images in different domains to the common domain in which feature-based registration excels. See Sec. 2.5 for details.

with the one used in training. For patch matching tasks, we use a sliding window backend due to its simplicity and maturity as a VTRN technique. More efficient implementations can be accommodated with no changes to the transform, and in general the matching backend should be selected to maximize performance.

For any VTRN architecture, reference images must also be proposed before the registration step. For clarity, we assume that black-box visual-inertial odometry is available to an accuracy sufficient only for the selection of a large reference image. In order to exhibit the advantages of a deep transform, we also assume that odometry cannot locate the onboard image within the selected reference, and consider matching independently. At the scales (images are roughly 1 km on a side) and update frame rates (roughly 20 Hz and greater) considered, this is a highly conservative assumption with extreme noise and drift rates. A lost aircraft with greater uncertainty searches a larger reference image or database [3], aided by the increased stability and distinctiveness of deep transformed imagery.

Datasets

We train our transform using publicly available aerial orthoimagery from various regions of the United States (Fig. 2.3). Full-foliage (leaf-on) and snow-cover imagery were obtained from the National Agricultural Imagery Program (NAIP) of the United States Department of Agriculture [30], and absent-foliage (leaf-off) imagery was obtained from the geospatial data program of the state of Connecticut [7]. Training and test sets were generated by co-registering cross-seasonal images into matching pairs. The datasets include man-made structures ranging from urban settlement to complete absence, with landcover including dense forest, agricultural fields, barren ground, coastline, and alpine tundra. We include a rugged mountainous dataset over the states of Wyoming and Montana, where classical registration performs poorly due to intense contrast changes caused by snowfall and severe mountain shadows. In addition to orthoimagery, we also test non-rigid transformations due to topography and off-nadir perspective by warping orthoimages onto co-registered digital elevation models (DEMs).

We consider two training/test dataset partition strategies, temporal and geographic, corresponding respectively to performance evaluation for VTRN and general registration use cases. Practical VTRN missions *require* a reference imagery database aboard the aircraft in advance and can always operate over their training dataset footprint. To evaluate the performance of our transform for VTRN, a representative

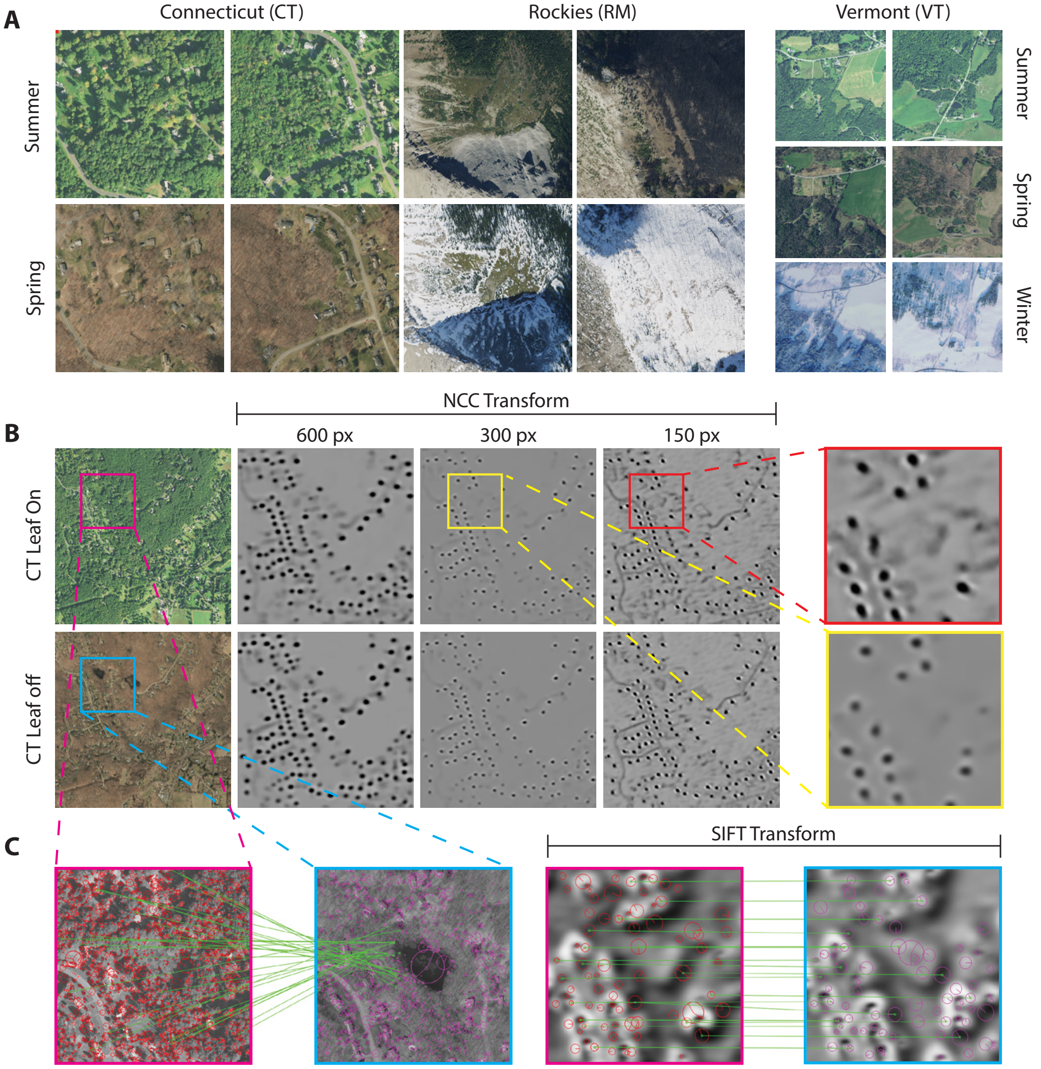


Figure 2.3: Representative images from the datasets along with samples of NCC- and SIFT-based transforms. (**A**) Example images from the Connecticut (CT), and Rockies (RM) datasets in their various domains. (**B**) NCC transformations of the CT "leaf-on/leaf-off" image pair. The different effects of training using 600×600, 300×300, and 150×150 image chips are shown. Zoomed-in images outlined in red and yellow highlight the details present when training with 150×150 image chips. (**C**) The presence of a large number of incorrectly-matched features in the grayscale image pair prevents RANSAC from finding the correct geometric transformation. The SIFT-optimized transform accentuates useful features for matching, while stripping away noisy, or unstable features that increase the number of outliers in matching.

test set is strictly temporally separated from the training set but overlaps with its geographic area. For general image registration we consider the case in which the area of interest is too large to be practically contained in a single training set. Under either assumption, seasonal effects may have been only observed outside of the operating region (for example, extreme snowfall rarely appears in public datasets), so the ability to abstract structures beyond their specific geographic position is critical. Accordingly, we also partition the dataset into training and testing sets with strict geographic separation. For both cases, we assume a 4:1 ratio of training to test data, with only training data used to optimize the network.

Without loss of generality, any of the domains can serve as a reference image at run-time depending on particular mission requirements and the operating area. The map accuracy of each dataset is 6 meters.

Connecticut set: The Connecticut set (CT) consists of 3638 co-registered leaf-on (summer 2016 and 2018) and leaf-off (early spring 2016) database image pairs randomly sampled over the US State of Connecticut (Fig. 2.3A, left). Leaf-off database images were resampled to match the 0.6 m/pixel NAIP resolution data, with dimensions 1270×1270. In order to evaluate performance when database images can be used in training, we use the 2016 leaf-on dataset collected two years before the training version (2018) for testing.

We also demonstrate a VTRN application with a simulated aircraft (Section 2.3) using 308 pairs taken over northwestern Connecticut. This imagery has an associated 10 cm DEM used to warp imagery and incorporate the effects of topography and off-nadir perspective. We note that this dataset is consistently more rugged and forested than the Connecticut set as a whole. The leaf-on images serve as NAVCAM imagery and are drawn from the same earlier edition used only for testing, and leaf-off images serve as a reference database.

Rockies set: The Rockies set (RM) is comprised of 90 co-registered NAIP quarter-quadrangle image pairs, taken between 2012 and 2018, capturing summer and snow-cover conditions in the Rocky Mountains of Wyoming and Montana (Fig. 2.3A, right). For training and testing, we subdivide the quarter-quadrangles into 1200×1200 non-overlapping tiles with a resolution of 1 meter per pixel. We note that this dataset is challenging even for manual registration due to extensive barren areas, a complete lack of man-made structure, snow and ice coverage, and severe mountain shadows. As with the Connecticut set, we also use earlier editions of the summer set solely for testing VTRN use cases.

Performance Evaluation

After training, we evaluate transform performance by comparing the accuracy of widely-used registration algorithms against a grayscale control. The test set consists of cross-seasonal pairs of source and reference images S and R that respectively produce registration queries and targets. We experimentally determine the best training procedure by restricting or combining different training sets and evaluating generality.

Because the typical failure mode of cross-seasonal registration is gross mismatch, we consider the correct match rate of image chips randomly drawn from each S and registered against the corresponding R as an estimate of the performance improvements afforded by a deep transform. This test is representative of typical VTRN operation, in which a NAVCAM images a small area within a database image, as well as non-rigid image registration in which translated chips are used to seed more complex transformations.

We use Intersection over Union (IoU) thresholds to identify match rates at varying levels of tolerance, as is standard for evaluating bounding box performance for detectors. If the IoU between a test chip and its predicted counterpart in the reference image is greater than the threshold, the registration is counted as successful. If the IoU is less than the threshold, it is counted as unsuccessful. We note that the limited map accuracy can prevent perfect IoU scores from being realized even with perfect matching.

For performance evaluation, we tested $K = 50$ randomly selected chips for each image pair in the test sets. As discussed above, we report match rate results for both temporally-separated and geographically-separated test sets.

Area methods: For area-based registration, we test a transform optimized for registration by NCC before discussing its relation to distribution-based methods. NCC is simply the linear correlation coefficient between two image patches, with a score of 1 if two images are perfectly positively correlated and a score of 0 if they are uncorrelated. The NCC score and training process are developed in more detail in Section 2.5.

We first subject the training process to three sets of experiments to determine best practices: we consider the effect of training chip size on detail recovered by the transform, the volume of training data required for a transform to perform well on areas it has not been exposed to, and the volume of additional data required to perform well on areas with different landcover. Because each of these experiments consider the generality of the transform, we use the geographically partitioned test set.

Although performance at test time always improves with larger test chip sizes, we observed that a smaller *training* chip size produces better results. Test chips were fixed at 300 pixels on a side throughout our experiments, but training chips that were 150 pixels on a side outperformed larger chips, generated sharply localized structures, and enhanced detail in challenging areas such as uniform deciduous forests (Figs. 3–4).

Additional training data improves performance in a geographically separated test set, but with returns that increase with IoU threshold due to to improved sharpness (Fig. 2.4). Exposure to the full training dataset, rather than a randomly-selected subset a tenth of its size, offers a 8 percentage-point improvement for the Connecticut set at an IoU threshold of 0.75, while increasing the IoU threshold to the range of 0.95 affords a 21 percentage-point improvement. Similarly, evaluation on the Rockies set yields a 4 percentage-point improvement at an IoU threshold of 0.75, but an 11 percentage-point improvement at a threshold of 0.95.

We also observe that additional training data with landcover *different* from the test set also improves matching rates (Fig. 2.4). On both the Rockies and Connecticut test sets, the best performance was achieved by training the network over all available Connecticut and Rockies training pairs rather than using each training set separately. For an IoU threshold of 0.9 and geographic partition of test and training sets, this network achieved match rates of 0.92 on the Connecticut set and 0.96 on the Rockies set, compared to Connecticut-only and Rockies-only values of 0.83 and 0.94 and grayscale control values of 0.50 and 0.66. We note that the merged training set offered a greater improvement on the Connecticut test set than on the Rockies test set, particularly at IoU thresholds above 0.85. This asymmetry is largely due to the complete absence of man-made structures in the Rockies set. The Rockies set provides relevant training samples away from built-up areas in Connecticut, but Connecticut training samples cause the deep transform to rely on man-made structures that are irrelevant in wilderness environments. Unsurprisingly, we also find that training sets must contain some landcover similar to the test set to offer increased performance over the grayscale control — training over Connecticut alone led to poor performance over the Rockies set and vice versa.

In order to consider VTRN use-cases in which the flight area is known in advance,

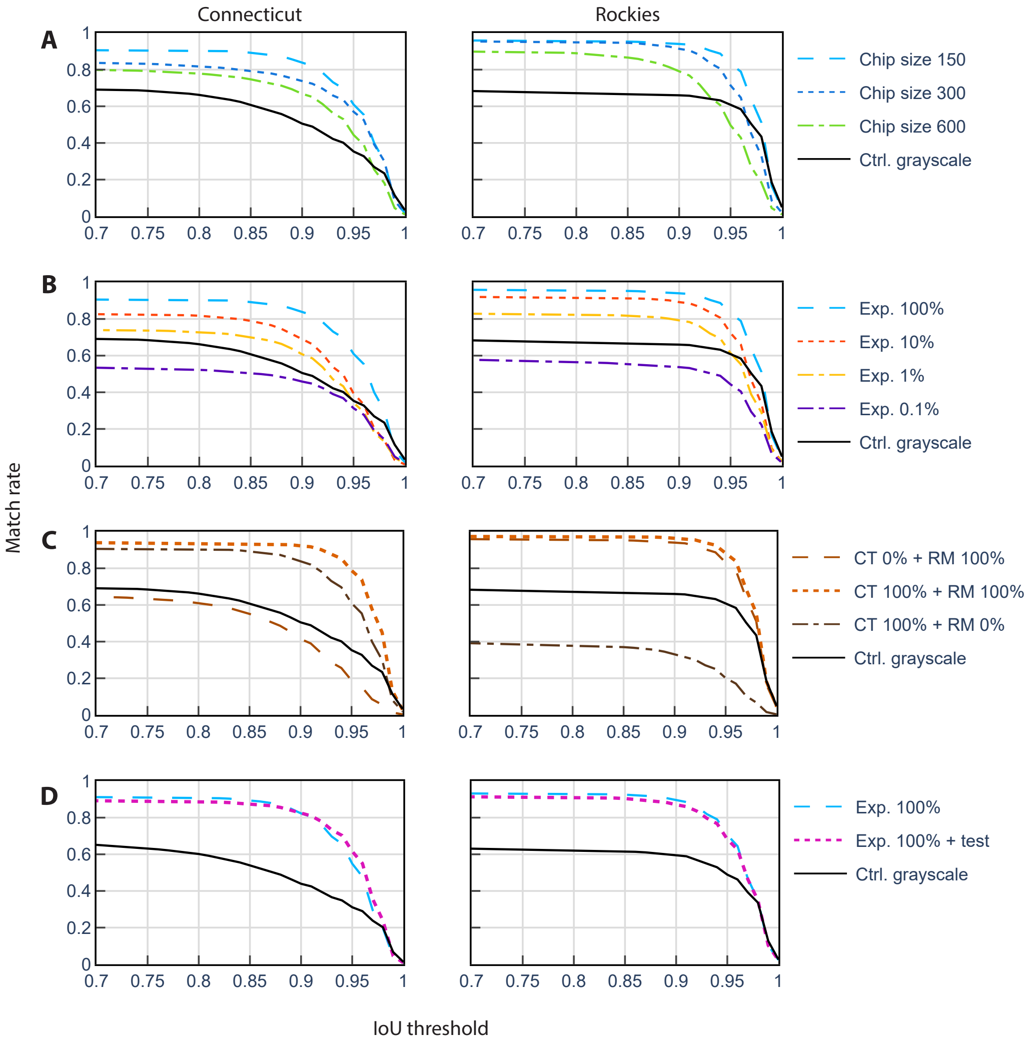


Figure 2.4: NCC image registration results. Models in the first three rows were evaluated with geographically-partitioned data while those in the last row were tested with the temporally-partitioned data. Unless otherwise noted, models were trained and tested using the datasets indicated by their column. (**A**) Plots in this row display the effect of varied chip sizes during training. (**B**) These plots show the positive match rate during testing while increasing the amount of training data seen by the transform model. (**C**) Transforms were trained on the entire Connecticut (CT) set, the entire Rockies (RM) set, and a merged set consisting of both. Transforms were evaluated separately on the Connecticut and Rockies test sets. (**D**) Models trained specifically for VTRN perform on-par with the best performing transforms from row A.

we merged the geographically-separated training and test sets into a single training set and considered performance on additional leaf-on test sets separated temporally from the original sets (Fig. 2.4). Overall, we observe little impact from including the geographic area of the test area in training, with a small increase in performance at high IoU thresholds and a small decrease in performance at lower IoU thresholds. With the exception of those containing permanent structural changes, pairs that fail only when exposed to the additional data typically have intra-domain landcover changes that were not covered in the training set. Because only one example of each domain was presented for each pair during training, this behavior is a symptom of overtraining a particular scene on a particular leaf-on appearance that can change severely at test time. On the other hand, the subtle increase at high IoUs appears to be due to increased sharpness afforded by inclusion of highly relevant training samples. The fact that overall matching performance is relatively insensitive to geographic coverage, however, suggests that features learned by the transform are abstract and high-level rather than tied to landmarks with a specific location.

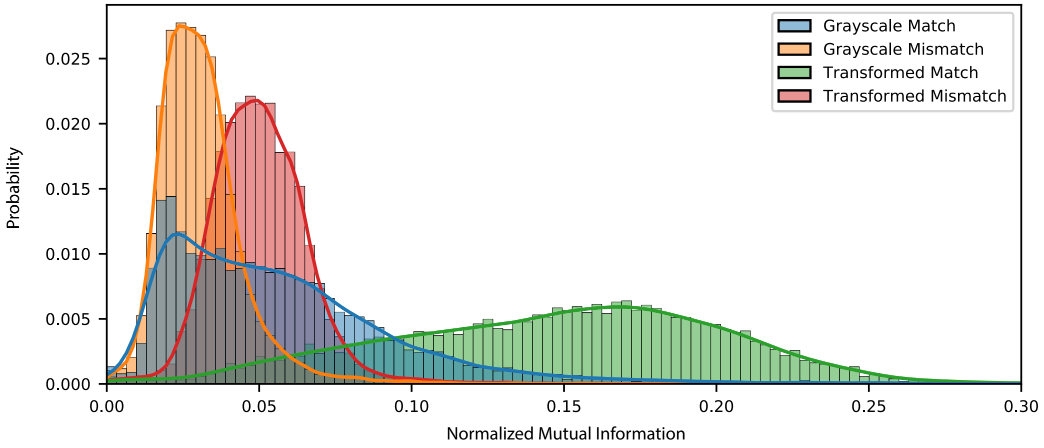


Figure 2.5: Distributions of mutual information values for mismatched and matched 300 × 300 image chips evaluated on a combined Connecticut and Rockies test set.

We also attempted to train a transform to directly optimize a normalized MI objective, but observed unstable training even with extremely small learning rates and deliberate overtraining over a single image pair also used for testing. This behavior is consistent with ill-posedness of MI as an optimization objective; the optimal transform is highly non-unique because MI and related distributional methods are invariant to any invertible deterministic function. Furthermore, MI is non-differentiable and cannot be used for backpropagation without the use of a smooth approximation. Fortunately, the NCC training objective can also be used to train transforms that improve MI-based registration. Although a full-resolution test of normalized MI performance over image chips is intractable (coarse-to-fine architectures are required even if heading and and altitude are known [2]), the NCC-trained transform enhances the ability of normalized MI to distinguish matching and non-matching pairs. On a set of 11600 matching and 11600 non-matching image chip pairs randomly drawn from the both test sets and exposed to our best-performing network, the Kullback-Leibler (KL) divergence between MI distributions for matching and mismatching grayscale image pairs was 0.22, while application of the NCC transform increases KL divergence to 1.58 (Fig. 2.5).

Feature methods: Unlike the straightforward loss functions available for optimizing area methods, adapting our transform for feature-based registration performance requires simultaneous optimization of detector and descriptor response. We illustrate our approach using SIFT features [17], to which our transform adds seasonal invariance to well-known rotation, scale, and linear-brightness invariance properties. SIFT features fail spectacularly in grayscale control tests across seasonal pairs because most features detected are small and associated with unreliable ephemeral landscape textures such as vegetation. Instead, we simultaneously optimize a deep transform for detector reproduciblilty, descriptor reproducibility, and descriptor distinctiveness to remove unreliable seasonal content (see Section 2.5 for details of the optimization objective and training procedure).

As in Section 2.3, we evaluate the performance of our deep transform using geographically-partitioned and temporally-partitioned training and testing sets. In order to evaluate VTRN for a nadir-looking NAVCAM with no other sensors available (area methods assume a known orientation and height), we consider 640 × 480 pixel test chips extracted from cross-seasonal pairs and subjected to randomly-varying translations, rotations, and scale transformations. We add the effect of topography and off-nadir perspective angles in Section 2.3.

We observe that our deep transform offers major performance advantages over grayscale control across all IoU thresholds (Fig. 2.6) and over all test sets and training strategies considered. Feature-based methods appear more sensitive to the landcover encountered in training than area methods, however, and exposure to impertinent samples can degrade performance. Merging the Rockies and Connecticut datasets improved match rates on the Connecticut test set at all IoU thresholds, largely due to improved feature density in forested areas, but worsened match rates on the Rockies test set due to a reliance on man-made structures that are not present at test time.

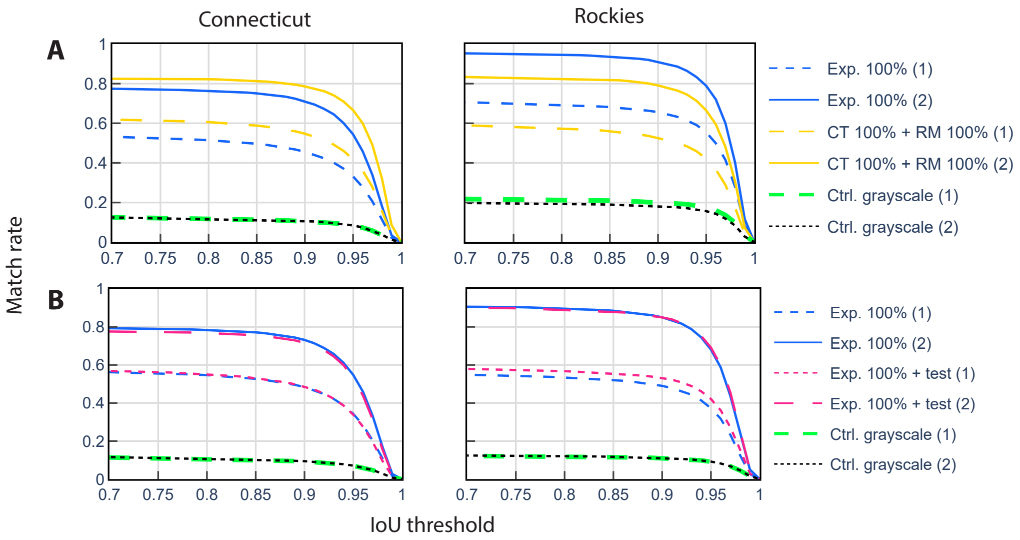


Figure 2.6: SIFT matching performance on geographically- and temporally-separated test sets evaluated using 640 × 480 test image chips. Test pairs that lack sufficient keypoints to calculate a camera pose are counted as mismatches in experiments denoted (1), but assumed useless for navigation in experiments denoted (2) and not counted. (**A**) For geographically-separated evaluations, we compare the matching rates of transformations trained on a single set (Exp. 100%, for CT or RM) against transformations trained on both sets (CT 100% + RM 100%). (**B**) To determine the effect of training over an anticipated flight area, we compare transformations trained on geographically-separated data (Exp. 100%) to transformations specifically trained over the same area as the test set, but from different years (Exp. 100% + Test).

In addition to improving match rates, the deep transform also provides a realistic assessment of the navigational utility of an image and fails far more gracefully than grayscale imagery. Images devoid of navigational cues, such as open water or uniform deciduous forest, simply do not have enough features available for registration after exposure to the transform and result in a failure. Grayscale imagery, on the other hand, often has enough spurious features available to return a registration estimate, and tends to generate mismatches when it should instead fail. Accordingly, we report experiments that both reject and retain failed test pairs in Fig. 2.6.

Finally, including the area of a temporally-separated test set within the training set slightly reduces the number of rejected pairs on the Rockies set, but with essentially no improvement on the Connecticut test set and a slight decrease in matching skill on accepted pairs in both sets. This decrease in performance is consistent with overtraining on the leaf-on appearance of the scene in training.

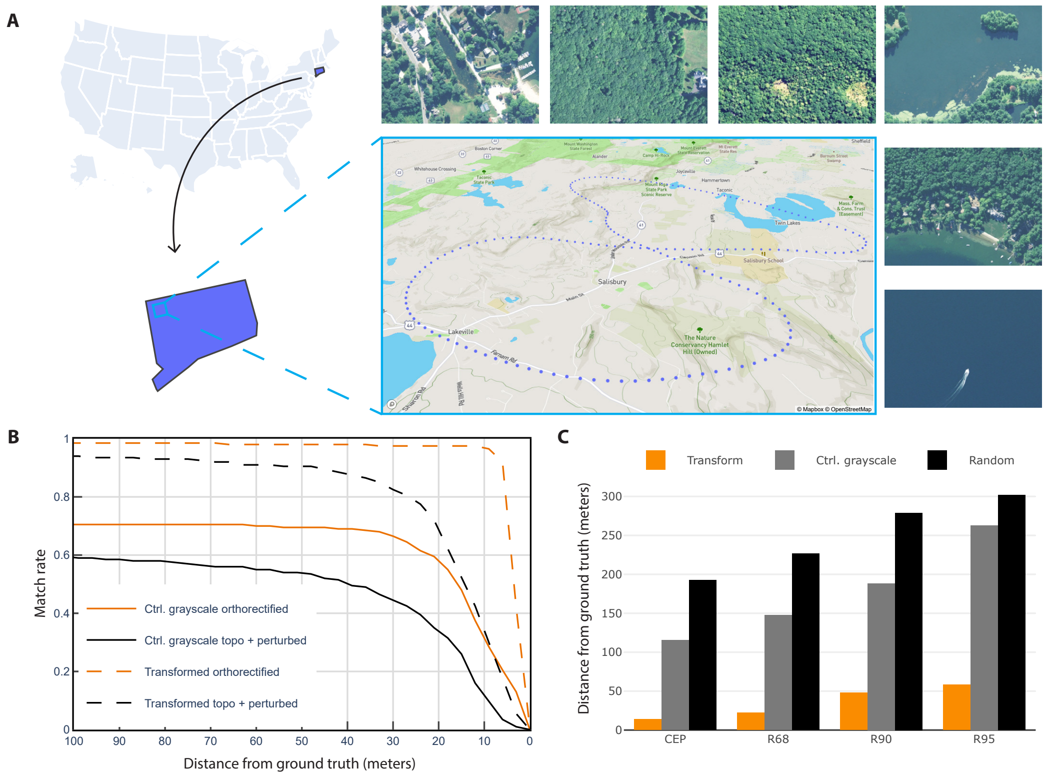


Figure 2.7: VTRN demonstration with a simulated flight over northwestern Connecticut. (**A**) The figure-eight flight trajectory is shown with a few select NAVCAM images (640 × 480) seen by the aircraft. The trajectory contains a mixture of small towns, agricultural areas, dense deciduous forest, and occasional open water. (**B**) NCC-based registration match rate at various distance thresholds from the ground truth positions. (**C**) SIFT-based registration distance-from-ground-truths at the 50th, 68th, 90th, and 95th percentiles.

Demonstration: Seasonally-Invariant VTRN

We evaluate our transform under realistic VTRN conditions during a simulated flight over a relatively undeveloped and rugged area of Northwest Connecticut (Fig. 2.7). The conditions encountered during the flight are considerably more challenging overall than the Connecticut set, and contain large uninterrupted expanses of deciduous forest with steeper terrain and sparser development. The flight area is covered by a DEM and a temporally separated test set, which are used to simulate imagery from a NAVCAM aboard a fixed-wing aircraft. As a result of the steep terrain and rolling motion of the aircraft, image registration involves a complex geometric transformation incorporating aircraft translation, attitude, and height changes along with and perspective effects due to a combination of camera optics, off-nadir viewing, and topography. Reference imagery is drawn from a database of leaf-off orthophotography, with the NAVCAM imagery taken in full leaf-on conditions separated by two years from the training data. Due to the complexity of this transformation and the intense seasonal changes, both the content invariance of deep transforms and the geometric advantages of conventional registration are needed to generate reliable absolute position updates.

We consider both NCC- and SIFT-based navigation along with associated deep transforms and a grayscale control set. SIFT can accommodate complex geometric transforms, in principle, as long as the number of features matched is adequate, while NCC assumes nadir-looking shots as well as an estimate of heading provided by a compass and an estimate of height provided by an altimeter. We do not supply SIFT with a heading estimate because of its rotational invariance, but NCC is supplied with perturbed (+/- 2 degrees) on-nadir shots and noisy heading estimates (+/- 5 degrees) to evaluate realistic operating conditions with instrument error. Contrary to typical practice, we also evaluate NCC without the benefit of online orthorectification. In order to isolate seasonal effects on performance from geometric effects, we also include a control NCC test with online orthorectification, perfect altimetry, and perfect compass measurements.

The test NAVCAM sequence consists of 200 image chips with dimensions 640×480, taken approximately every 180 meters along a 36 km figure-eight aircraft trajectory roughly 200 m above the terrain. Due to the frequent and sequential nature of the imagery, shots are somewhat correlated and cluster in challenging areas that were far less abundant in the full Connecticut set. We use a database image of 1270×1270 pixels for NCC and a smaller 800×800 pixel image for SIFT due to its sparser nature and use with poorly-constrained transformations. We also note that three of the 200 shots contained only open water, and were either mismatched or rejected for lacking navigational content by the registration algorithms in all experiments.

For NCC, the VTRN backend was performed with a patch-based sliding window method, while SIFT features were matched using standard brute-force sum-of-absolute-differences (SAD) scores. Extremely noisy onboard odometry was simulated and used to select 0.6 m/pixel reference images that contain the aircraft location somewhere within the prescribed map size. This odometry error assumption far exceeds the average interval between shots of 180 meters. Each shot was treated as independent, and odometry was not allowed to influence the initial pre-VTRN uncertainty beyond the size of the reference image.

Test images were subjected to the best-performing NCC and SIFT transforms determined in Sections 2.3 and 2.3, which were trained over Connecticut and Rockies training data consolidated into a single set. SIFT feature detection, extraction, and matching were performed using OpenCV with default settings, and random sample consensus (RANSAC) was used to estimate a full-affine geometric transformation without knowledge of the aircraft height or attitude. Off-nadir perspective and topography allow test images to differ in shape from ground truth and prevent perfect IoU scores even if localization is perfect, so navigation performance was evaluated using the centroid distance between registered images and ground truth. We also calculate several centroid distance statistics, including circular error probable (CEP), R68, R90, and R95 scores, which correspond respectively to the 50th, 68th, 90th, and 95th percentiles of centroid distance.

VTRN performance: When used with a deep transform, NCC proved to be a powerful and robust navigation tool, even in the presence of steep topography and viewing angle perturbations that violated its geometric assumptions. The deep transform had far fewer mismatches than grayscale and also localized close matches more accurately (Fig. 2.7). The CEP for the unmodified transformed image was 14 m, compared to 3 m for the idealized geometry, 40 m for unmodified grayscale, 16 m for the idealized grayscale geometry, and 168 m for a set of 100 randomly guessed centroids per frame. The deep transform had few gross mismatches, with transformed imagery having an R68 score of 19 m, an R90 score of 46 meters, and an R95 score of 115 m, compared to 4 m, 7 m, and 14 m for idealized conditions. Because each of these scores are considerably smaller than the size of the NAVCAM image even at high percentiles, this divergence suggests that much of the error was caused by the impossibility of matching distorted test images against rectangular, orthorectified ground truth using a rigid translation. Grayscale control imagery, on the other hand, had an R68 score of 150 m, an R90 score of 233 m and R95 score of 262 m, which improved to 32 m, 228 m, and 260 m under ideal imaging geometries. The departure in R68 but approximate convergence in R90 and R95 score between these two experiments suggests that grayscale was highly sensitive to topography and viewing angle. Furthermore, grayscale was also plagued by gross mismatches caused by seasonal content, as evidenced by comparison to random centroid guessing scores of 197 m, 241 m, and 258 m.

SIFT was able to provide accurate and reliable accurate position estimates from deep transformed imagery even with a camera as the sole navigation instrument, but at the price of a much lower update rate. Although the deep transform was able to stabilize

Table 2.1: Runtime and memory consumption of the deep transform for various input image sizes.

Image size	GPU I/O (s)	GPU (s)	CPU (s)	Memory (GB)
300×300	0.0003	0.0192	0.700	0.67
640×480	0.0004	0.0597	2.734	2.13
1280×720	0.0008	0.1722	8.139	6.25
1600×1200	0.0015	0.3623	17.250	12.95

imagery, it struggled to enhance and concentrate adequate numbers of strong features in the most challenging areas. The stabilizing effect, however, recovered high levels of accuracy among useful images identified using an empirical feature match count. 11 percent of images were accepted as reliable in the transformed imagery, with a CEP of 14 meters and a maximum error of 76 meters. In contrast, grayscale control imagery was unable to anticipate reliable or unreliable frames at any feature count threshold. For the same experiment, 53 percent of images were accepted, with a CEP of 116 m and an R95 of 263 m, for a minor advantage over random guess values of 192 m and 301 m respectively.

Hardware and Runtime Metrics

Our machines were configured with Intel Core i9-7900x processors with 128GB of memory. We trained the deep transforms using Nvidia Quadro RTX 8000 (48GB) and the Titan RTX (24GB) GPUs. Since VTRN requires image chips to be transformed in an online setting, we recorded runtime and memory usage for various navigation chip sizes, including those used in our VTRN demonstration. We benchmarked the deep transform using the Titan RTX and report these results in Table 2.1. We find that the GPU-accelerated deep transform can process 640 × 480 chips at 17 Hz and even larger 1600×1200 chips at 2.7 Hz. For small image chips, the deep transform can still operate at reasonable speeds even without a GPU, but at a cost of a slower position update rate. We do not report metrics for larger database images, which are not transformed in real-time but instead calculated offline in advance and cached for quick lookup.

2.4 Discussion

Overall, our transform learns highly general, abstract features that are semantic in nature rather than landmarks tied to specific geographic locations. Our transform need not be trained over the exact area where it will be used, and only small amounts of data were required to surpass performance on standard grayscale imagery. The advantages of isolating geometry to an engineered conventional back-end, rather than training, were also apparent in feature-based experiments. Using a SIFT back-end, deep transforms were able to accurately assess reliability and accommodate complex topographic and perspective transformations despite training only on orthophotography. We also find that NCC is an extraordinarily powerful tool when supplemented with a deep transform despite its simplicity. Deep transformed NCC exhibited robust performance against severe seasonal effects and perturbed imaging conditions that violated its translation-only, mutually orthorectified geometric assumptions.

The usual case of more data improving performance remains largely true, but targeted training set design can help achieve better results. Training sets must include at least some landcover similar to what will be encountered at test time to be effective, with increasing volumes improving the spatial precision of the transform. Including as much data as possible improved matching for area-based methods even if additional training examples had highly dissimilar landcover, but feature-based approaches required care to avoid crippling a deep transform with irrelevant samples. Although the inclusion of roadless areas from the Rockies training set improved feature matching away from built-up areas in Connecticut, for example, overexposure to buildings in training also hindered the ability of the deep transform to construct useful SIFT features in images where they were not present. Consequently, systems designed to operate entirely away from man-made structures, such as for extraterrestrial applications, should not use training sets dominated by them. Our transform also accommodated ephemeral changes that were not anticipated in advance, which suggests that general time-series data is useful for stabilizing VTRN even if seasonal disturbances are not expected. Deep mountain shadows that change throughout the day and year, for example, had improved matching as natural consequence of their presence in the set and the self-supervised architecture (Fig. 2.8). This tendency is further evidenced by the possibility of overtraining over specific leaf-on appearances, which caused inclusion of VTRN database imagery in training to *decrease* performance on a temporally-separated test set in some cases.

Although feature-based methods using SIFT features are more flexible in the geometric transformations that they can accommodate, the NCC training objective was much more robust in the most challenging and unstable areas. Unless operational considerations preclude the use of NCC and require feature-based approaches, such

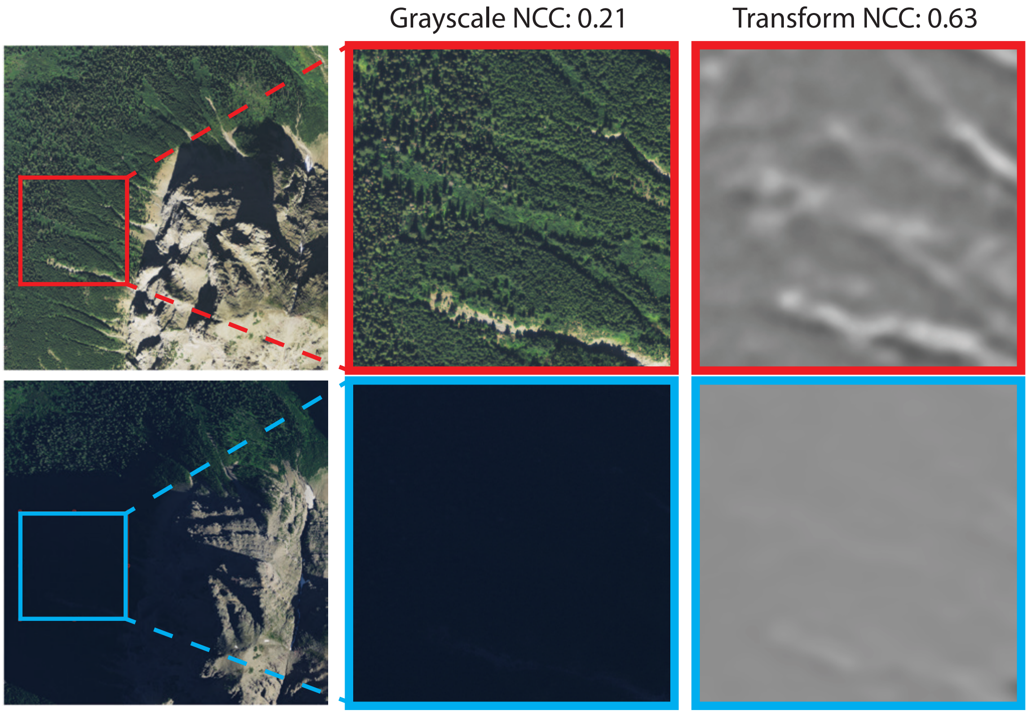


Figure 2.8: Example of improved robustness to severe mountain shadows in test data. Self-supervised training causes the transform to attempt to mitigate any unstable content without explicit annotation. Intense lighting changes, for example, were occasionally present in the Rockies dataset and were addressed by the transform even though they were not specifically identified as a relevant seasonal disturbance. Despite information loss to saturation, areas in deep shadow were brightened (**left**, red and blue boxes are co-registered) and stable landscape features were enhanced. The transform was not exposed to either image during training.

as the presence of extreme topography at low altitudes or the lack of an attitude estimate, the additional information afforded by NCC-based matching is preferable to the geometric flexibility of SIFT when severe content changes are present. NCC was able to infer reproducible structure in aggressively changing environments where SIFT had difficulty finding features, and also proved to be robust to local topographic and viewpoint perturbations that would ordinarily call for the use of feature-based approaches. Nonetheless, the deep transform enhanced the reliability of SIFT-based navigation in areas that were impossible for grayscale imagery—unusable shots simply lacked features and were consistently identified in advance. Unlike grayscale, which proposed over-confident matches little better than random guessing, unusable shots were discarded rather than being naively allowed to disrupt navigation.

The NCC objective also supported registration by mutual information maximization without incurring its extreme computational expense or the complexity of backpropagation through a nondifferentiable objective [31]. NCC is well-known to be equivalent to MI for normally-distributed random variables, and transforms trained on NCC improved the separation of matching and non-matching mutual information scores (Fig. 2.5) while also being much easier and faster to train. Because such transforms are already directly optimized for NCC, MI-based registration architectures and their inherent difficulties can be replaced *entirely* by a real-time NCC-based architecture with a deep transform.

Finally, the results presented here isolate difficulties associated with content changes, and consider odometry and filtering only to the extent that they generate reference images including the aircraft location. Filtering can serve not only to improve running estimates of position, as with any navigation technique, but aid the VTRN matching process itself by restricting the size of a registration problem to a tight uncertainty. If position and attitude are sufficiently well known, a registration solution can be proposed in advance and VTRN used to modify and confirm it within tight bounds. Feature-based approaches stand much to gain from filtering in particular, as the relatively unconstrained nature of the geometric transformations they solve benefits highly from reliable initial guesses and bounds.

2.5 Materials and Methods

We structure our transform as an FCN, with a self-supervised training architecture that assumes co-registered image pairs but identifies and extracts seasonally-stable structures on its own. During training (Fig. 2.2), image pairs with cross-seasonal differences (leaf-on vs leaf-off or summer vs snow) are consecutively exposed to the FCN with shared weights, which outputs a pair of single-channel transformed images with the same dimensions as the inputs. The network is trained using a loss function calculated from the performance of a specified registration technique (for example, NCC or SIFT) on an auxiliary chip-matching task. Although the specific structure of the loss function depends on the choice of registration backend, the transform can be trained to optimize any algorithm with a differentiable and well-posed matching score. In doing so, we harness the statistical power of deep learning to add seasonal robustness to legacy techniques.

After training, the FCN constitutes a single-stream image transform that strips source and reference images of their seasonally-unstable content for stable registration. In the rest of this section, we highlight specific network details and consider the construction of seasonally-invariant transforms for area-based and feature-based registration.

Network Architecture

We chose U-Net [27] as an example model, although any FCN architecture that preserves image resolution will suffice. In a deep transform context, U-Net maps an input grayscale image of dimension $W \times H$, with intensity values between [0,1], to an output grayscale image of the same size and compressed to [0,1] using a sigmoid function. Inputs are normalized using a fixed mean and standard deviation derived from the training set. In order to avoid checkerboard artifacts associated with deconvolution, we replace the deconvolution layers of original U-Net with blocks consisting of an upsampling operation with bicubic interpolation followed by a convolution [23].

Loss function and regularization: During training, the loss function is calculated by passing pairs of transformed chips (T_i^1, T_i^2) to the matching function of a desired image registration algorithm. The matching function generates a normalized similarity score $\hat{y}(T_i^1, T_i^2) \in [0, 1]$. The normalized similarity score $\hat{y}$ is in turn compared to a binary label y_i that indicates whether the chips were originally co-registered ($y_i = 1$) or not ($y_i = 0$). The loss $\mathcal{L}$ is the sum of the contributions from each image pair and is used to update the network weights by backpropagation. Accordingly, the chip pair (T_i^1, T_i^2) contributes

$$\mathcal{L}^{(i)} = \|\hat{y}(T_i^1, T_i^2) - y_i\| \tag{2.1}$$

to $\mathcal{L}$, where $\| \cdot \|$ is a differentiable norm. The only requirement for $\hat{y}$ is that the matching score be differentiable with respect to the network parameters and well-posed for backpropagation.

To avoid trivial transforms, such as setting all pixels identically to zero, non-matching samples with $y_i = 0$ must be presented during training. Co-registered (positive) chips are selected with probability 0.5 and driven towards a perfect matching score, while non-matching (negative) samples are driven towards either a worst-case mismatch score or separation margin.

Implementation and training details: Our U-Net architecture is based off of the github repository, Pytorch-UNet [18], and adapted by replacing the deconvolutions to resolve checkerboard artifacts as mentioned before. All transforms were trained using the Adam optimizer with a learning rate of 1e-5 over 300 epochs. The learning rate was decayed at an exponential rate of 0.995 every two epochs. Batch size was adjusted depending on training chip size to fit on the GPU, but no larger than 16. We relied on the OpenCV and Kornia Python libraries to implement the loss function for SIFT-based image registration [6, 26].

Area Methods

Our training procedure for area-based registration methods directly optimizes registration performance on the training set. All area-based registration techniques use a similarity score based on intensity values to determine whether image chips match or not, which is simply driven towards perfect (matching) or worst-case (non-matching) values to train our network. As discussed in Section 2.4, the NCC objective is widely applicable and also useful for MI and related area-based methods.

Normalized Cross-Correlation: For a chip pair (T_i^1, T_i^2), the zero-mean NCC score is defined as

$$\hat{y}_i^{\text{NCC}} = \frac{1}{n\sigma_u\sigma_v}\sum_{u,v}(T_i^1(u,v) - \mu_1)(T_i^2(u,v) - \mu_2), \tag{2.2}$$

where u, v are pixel coordinates in the chips, n is the number of pixels in each chip, and μ_1, σ_1 and μ_2, σ_2 are the respective means and standard deviations of the transformed chips. To optimize the transform, we drive the similarity score $\hat{y}_i^{\text{NCC}}$ towards the label $y_i = 1$ for positive samples and $y_i = 0$ for negative samples using squared error over K pairs:

$$\mathcal{L}_{\text{NCC}} = \frac{1}{K}\sum_{i=0}^{K} \|\hat{y}_i^{\text{NCC}} - y_i\|_2^2. \tag{2.3}$$

Feature Methods

Unlike area methods, feature-based methods consist of two objectives that must be optimized: detection, in which reliable feature locations are identified, and extraction, in which descriptors that consistently represent unique structures are selected at detected locations (Fig. 2.2).

Scale-Invariant Feature Transform: To optimize for SIFT, we incorporate the two objectives mentioned above into our loss function. Detector response is driven to a common domain using NCC optimization over the difference-of-Gaussian (DoG) pyramids from a pair of transformed image chips (T_i^1, T_i^2) :

$$\mathcal{L}_{\text{det}} = \frac{1}{K}\sum_{i=0}^{K} \|\hat{y}^{\text{NCC}}(P_j^1, P_j^2) - y_i\|_2^2 \tag{2.4}$$

where (P_j^1, P_j^2) are pairs of 64×64 patches sampled from the DoG pyramids of the transformed input pair. During training, we randomly extract 100 pyramid sample pairs from each input image with a negative matching rate of 0.5.

To optimize descriptor performance, we calculate the pairwise distance between the M detected keypoints in T_i^1 and the N detected keypoints in T_i^2 (eq. 2.5). During training, M and N are each fixed at 500. Normalized descriptor pairs (D_m^1, D_n^2) are extracted from each image, where D_m^1 is the mth descriptor extracted from T_i^1 and D_n^2 is the nth descriptor extracted from T_i^2. The Euclidean distance between the descriptor pairs $\hat{y}_{m,n}^{\text{desc}}$ is driven towards a label $y_{m,n}$, which is 0 if the corresponding keypoints match in scale and location, and driven towards the maximum margin $a = 2$ if they do not:

$$\hat{y}_{m,n}^{\text{desc}} = ||D_m^1 - D_n^2||_2 \tag{2.5}$$

$$\mathcal{L}_{\text{desc}} = \frac{1}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} y_{m,n} \cdot \hat{y}_{m,n}^{\text{desc}} + (1 - y_{m,n}) \cdot \max(0, a - \hat{y}_{m,n}^{\text{desc}}). \tag{2.6}$$

The two loss functions are then jointly optimized as

$$\mathcal{L} = \beta \mathcal{L}_{\text{det}} + \mathcal{L}_{\text{desc}}, \tag{2.7}$$

where β is empirically chosen to be 10. In order to handle more keypoints at different scales, we randomly and identically scale the input pairs between 0.6 and 1.4, and crop to 256 × 256, 400 × 400, or 512 × 512 pixels before transformation.

Acknowledgments

The authors thank P. Tokumaru. **Funding:** This project was in part funded by the Boeing Company with R. K. Li as Boeing Project Manager. C. L. acknowledges the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1745301.

References

[1] A. Anoosheh, T. Sattler, R. Timofte, M. Pollefeys, and L. V. Gool. “Night-to-Day Image Translation for Retrieval-based Localization”. In: *2019 International Conference on Robotics and Automation (ICRA)*. 2019. DOI: 10.1109/icra.2019.8794387.

[2] A. Ansar and L. Matthies. “Multi-modal image registration for localization in Titan’s atmosphere”. In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2009. DOI: 10.1109/iros.2009.5354586.

[3] R. Arandjelovic and A. Zisserman. "All About VLAD". In: *2013 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2013. DOI: 10.1109/cvpr.2013.207.

[4] J.-P. Avouac and S. Leprince. "Geodetic Imaging Using Optical Systems". In: *Treatise on Geophysics*. Elsevier, 2015, pp. 387–424. DOI: 10.1016/b978-0-444-53802-4.00067-1.

[5] M. Blendowski and M. P. Heinrich. "Combining MRF-based deformable registration and deep binary 3D-CNN descriptors for large lung motion estimation in COPD patients". In: *International Journal of Computer Assisted Radiology and Surgery* 14.1 (2018), pp. 43–52. DOI: 10.1007/s11548-018-1888-2.

[6] G. Bradski. "The OpenCV Library". In: *Dr. Dobb's Journal of Software Tools* (2000).

[7] Capitol Region Council of Governments of Connecticut. *CRCOG Orthoimagery*. https://cteco.uconn.edu/data/flight2016/. 2016.

[8] J. R. Carr and J. S. Sobek. "Digital Scene Matching Area Correlator (DSMAC)". In: *Image Processing for Missile Guidance*. Ed. by T. F. Wiener. SPIE, 1980. DOI: 10.1117/12.959130.

[9] G. Conte and P. Doherty. "Vision-Based Unmanned Aerial Vehicle Navigation Using Geo-Referenced Information". In: *EURASIP Journal on Advances in Signal Processing* 2009.1 (2009). DOI: 10.1155/2009/387308.

[10] L. Downes, T. J. Steiner, and J. P. How. "Deep Learning Crater Detection for Lunar Terrain Relative Navigation". In: *American Institute of Aeronautics and Astronautics (AIAA) Scitech 2020 Forum*. 2020. DOI: 10.2514/6.2020-1838.

[11] J. P. Golden. "Terrain Contour Matching (TERCOM): A Cruise Missile Guidance Aid". In: *Image Processing for Missile Guidance*. Ed. by T. F. Wiener. SPIE, 1980. DOI: 10.1117/12.959127.

[12] A. Gupta, Y. Peng, S. Watson, and H. Yin. "Multitemporal Aerial Image Registration Using Semantic Features". In: *Intelligent Data Engineering and Automated Learning – IDEAL 2019*. 2019, pp. 78–86. DOI: 10.1007/978-3-030-33617-2_9.

[13] G. Haskins, U. Kruger, and P. Yan. "Deep learning in medical image registration: a survey". In: *Machine Vision and Applications* 31.1-2 (2020). DOI: 10.1007/s00138-020-01060-x.

[14] L. H. Hughes, M. Schmitt, L. Mou, Y. Wang, and X. X. Zhu. "Identifying Corresponding Patches in SAR and Optical Images With a Pseudo-Siamese CNN". In: *IEEE Geoscience and Remote Sensing Letters* 15.5 (2018), pp. 784–788. DOI: 10.1109/lgrs.2018.2799232.

[15] F. Kendoul. “Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems”. In: *Journal of Field Robotics* 29.2 (2012), pp. 315–378. DOI: 10.1002/rob.20414.

[16] N. Longbotham, F. Pacifici, S. Malitz, W. Baugh, and G. Camps-Valls. “Measuring the Spatial and Spectral Performance of WorldView-3”. In: *Fourier Transform Spectroscopy and Hyperspectral Imaging and Sounding of the Environment*. OSA, 2015. DOI: 10.1364/hise.2015.hw3b.2.

[17] D. G. Lowe. “Distinctive Image Features from Scale-Invariant Keypoints”. In: *International Journal of Computer Vision* 60.2 (2004), pp. 91–110. DOI: 10.1023/b:visi.0000029664.99615.94.

[18] A. Milesi. *Pytorch-UNet*. https://github.com/milesial/Pytorch-UNet. 2017.

[19] J. L. Moigne, N. S. Netanyahu, and R. D. Eastman, eds. *Image Registration for Remote Sensing*. Cambridge University Press, 2009. DOI: 10.1017/cbo9780511777684.

[20] A. Mourikis, N. Trawny, S. Roumeliotis, A. Johnson, A. Ansar, and L. Matthies. “Vision-Aided Inertial Navigation for Spacecraft Entry, Descent, and Landing”. In: *IEEE Transactions on Robotics* 25.2 (2009), pp. 264–280. DOI: 10.1109/tro.2009.2012342.

[21] Z. Murez, S. Kolouri, D. Kriegman, R. Ramamoorthi, and K. Kim. “Image to Image Translation for Domain Adaptation”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018. DOI: 10.1109/cvpr.2018.00473.

[22] A. Nassar, K. Amer, R. ElHakim, and M. ElHelw. “A Deep CNN-Based Framework For Enhanced Aerial Imagery Registration with Applications to UAV Geolocalization”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2018. DOI: 10.1109/cvprw.2018.00201.

[23] A. Odena, V. Dumoulin, and C. Olah. *Deconvolution and Checkerboard Artifacts*. http://distill.pub/2016/deconv-checkerboard/. 2016.

[24] W. K. Pratt. *Digital Image Processing: PIKS Scientific Inside*. Wiley-Interscience, 2007. ISBN: 0471767778.

[25] B. Reddy and B. Chatterji. “An FFT-based technique for translation, rotation, and scale-invariant image registration”. In: *IEEE Transactions on Image Processing* 5.8 (1996), pp. 1266–1271. DOI: 10.1109/83.506761.

[26] E. Riba, D. Mishkin, D. Ponsa, E. Rublee, and G. Bradski. “Kornia: an Open Source Differentiable Computer Vision Library for PyTorch”. In: *Winter Conference on Applications of Computer Vision*. 2020.

[27] O. Ronneberger, P. Fischer, and T. Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.

[28] J. Townshend, C. Justice, C. Gurney, and J. McManus. “The impact of misregistration on change detection”. In: *IEEE Transactions on Geoscience and Remote Sensing* 30.5 (1992), pp. 1054–1060. DOI: 10.1109/36.175340.

[29] D. Tuia, C. Persello, and L. Bruzzone. “Domain Adaptation for the Classification of Remote Sensing Data: An Overview of Recent Advances”. In: *IEEE Geoscience and Remote Sensing Magazine* 4.2 (2016), pp. 41–57. DOI: 10.1109/mgrs.2016.2548504.

[30] U.S. Department of Agriculture Farm Service Agency, Aerial Photography Field Office. *National Agricultural Imagery Program*. earthexplorer.usgs.gov. 2012.

[31] M. Unser and P. Thevenaz. “Optimization of mutual information for multiresolution image registration”. In: *IEEE Trans. Image Processing* 9.12 (2000), pp. 2083–2099. DOI: 10.1109/83.887976.

[32] P. Viola and W. M. Wells III. “Alignment by maximization of mutual information”. In: *International Journal of Computer Vision* 24.2 (1997), pp. 137–154.

[33] Z. Yang, T. Dan, and Y. Yang. “Multi-Temporal Remote Sensing Image Registration Using Deep Convolutional Features”. In: *IEEE Access* 6 (2018), pp. 38544–38555. DOI: 10.1109/access.2018.2853100.

[34] S. Zagoruyko and N. Komodakis. “Learning to compare image patches via convolutional neural networks”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015. DOI: 10.1109/cvpr.2015.7299064.

[35] M. D. Zeiler and R. Fergus. “Visualizing and Understanding Convolutional Networks”. In: *Computer Vision – ECCV 2014*. 2014, pp. 818–833. DOI: 10.1007/978-3-319-10590-1_53.

Chapter 3

SELF-SUPERVISED LANDMARK DISCOVERY FOR LARGE SCALE VISUAL TERRAIN-RELATIVE NAVIGATION

[1] C. Lee, E. Mesic, and S.-J. Chung. "Self-Supervised Landmark Discovery for Terrain-Relative Navigation". In: *ICRA 2023 Workshop on Unconventional spatial representations: Opportunities for robotics*. Available at https://usr2023.github.io/papers/Landmark.pdf. 2023.

3.1 Abstract

We present a landmark discovery algorithm to automatically detect and identify optimal landmarks for aerial localization in visual terrain-relative navigation (VTRN) pipelines for Global Navigation Satellite Systems (GNSS) denied navigation. Our method employs self-supervised contrastive learning to identify and encode visual landmarks despite illumination, viewpoint, and seasonal changes. Using publicly available aerial imagery, we demonstrate that our approach can detect and re-identify sparse landmarks across seasons and enable localization within 10 meters. Lastly, our method minimizes the storage requirement compared to current VTRN methods, expanding the navigable area size.

3.2 Introduction

In absence of Global Navigation Satellite Systems (GNSS), uninhabited aerial vehicles (UAV) can pinpoint their exact geolocation by matching images from their navigational camera (NAVCAM) to known, georeferenced images, in a process known as visual terrain-relative navigation (VTRN). In the last few decades, image registration-based VTRN approaches have dominated GNSS-denied robotic navigation systems, driving applications like planetary entry, landing, and descent (EDL) and cruise missile guidance [3, 10]. These approaches typically rely on registration backends, powered by area-based template matching and/or feature-based homography estimation, to provide precise geolocation [10]. However, they face two problems: First, they fail when faced with seasonal or illumination variation, relying on strategic mission planning [3, 10] or deep learning to compensate [7]. Second, they require georeferenced imagery or extracted local feature descriptors to be stored on memory-constrained UAVs, which limits navigation area size.

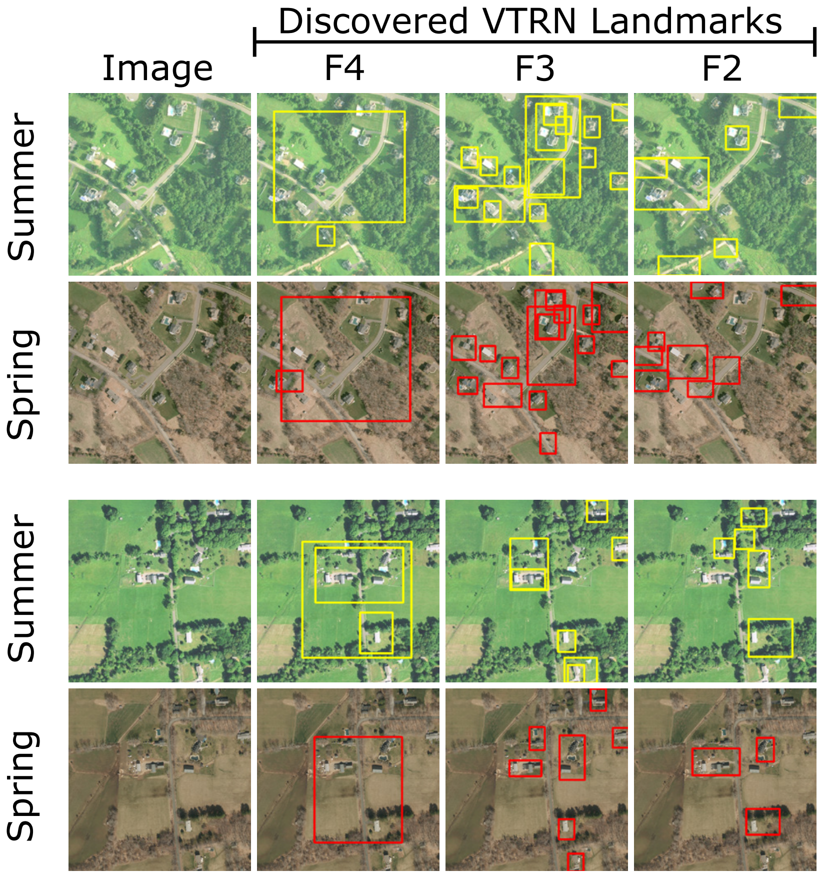


Figure 3.1: Examples images from the dataset and proposed landmarks via our discovery module. F2/3/4 are resolution streams of the network activations that the landmarks were extracted from (F4 is lowest). Landmarks from each stream are displayed prior to non-maximum suppression and may overlap with those in other streams. Note that not all discovered landmarks are required to have a matching pair across seasons for localization to work.

In contrast, landmark-based VTRN approaches are robust and lightweight, providing accurate but sparser geolocation updates by re-identifying a small set of known landmarks [6, 12, 14]. These landmarks can be encoded with invariances to common VTRN perturbations like seasonal variation and cached as low-dimensional vectors. For landmark-based approaches, the main challenge is choosing a set of landmarks that is large enough to provide a steady rate of geolocation updates, but with each landmark being easily re-identifiable.

Today, convolutional neural networks (CNN) have made it possible to easily detect and identify such landmarks despite appearance and illumination variations, but expert guidance is generally still used to select good landmarks for CNN training. Examples of this include crater detection for lunar EDL [6], and detection of various human-made structures (roads, houses, and buildings) for UAV navigation over urban/suburban areas [16, 13]. Although human expertise helps in these settings, there are three downsides: First, human cognitive and visual biases could result in potentially-useful landmarks being overlooked [11, 9]. Second, humans are not good at pattern recognition in unstructured and noisy terrain, whereas learning-based methods are good, when provided enough data. Third, having humans-in-the-loop means manual and tedious mission planning.

In this work, we propose a self-supervised landmark-based VTRN pipeline for UAV localization across seasons. Our primary contribution is a landmark discovery algorithm that learns to automatically identify navigationally-useful and seasonally-robust landmarks (Fig. 3.1) without requiring human expertise. We investigate the localization potential and robustness of individual components and demonstrate their efficiency over current VTRN techniques.

This work is outlined as follows: we briefly go over prior works in Sec. 3.3 and follow with our approach (Sec. 3.4), results (Sec. 3.5), and conclusion (Sec. 3.6).

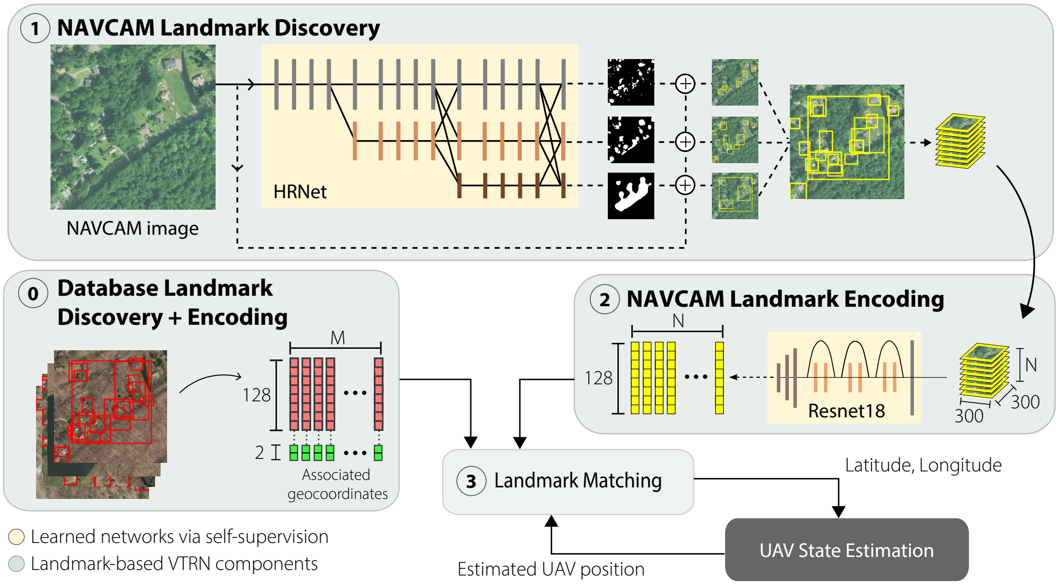

1 NAVCAM Landmark Discovery
HRNet
NAVCAM image
0 Database Landmark Discovery + Encoding
M
128
2
Associated geocoordinates
2 NAVCAM Landmark Encoding
N
128
Resnet18
N
300
300
3 Landmark Matching
Latitude, Longitude
UAV State Estimation
Estimated UAV position
○ Learned networks via self-supervision
○ Landmark-based VTRN components

Figure 3.2: Flowchart of our proposed landmark-based localization method in a UAV state estimation pipeline. The landmark discovery module (**Step 1**) extracts cropped landmark proposals based on the activations of a CNN. Landmark crops are encoded using a separate network (**Step 2**) and matched (**Step 3**) against a precomputed database of georeferenced landmark encodings (**Step 0**).

3.3 Related Work

Compared to local feature-based approaches, current landmark-based VTRN approaches tend to focus on landmarks with more semantic meaning, such as lunar craters, roads, and buildings [6, 16, 13]. They typically consist of three components: landmark detection, encoding, and matching. For example, [6] localizes lunar craters using a CNN and matches the geometric characteristics of found craters against a georeferenced crater database to get location. Recent works [16, 13] also leverage human-selected landmarks such as road networks and buildings for aerial navigation, but do not operate outside of urban environments. These current approaches reduce the storage overhead of local features but rely on humans to select landmark classes for CNN training. In our work, we seek to automate landmark discovery by directly learning from image data using a self-supervised learning scheme.

Recent VTRN approaches eschew landmarks and directly match globally encoded NAVCAM and database images. [2] trains a CNN autoencoder to densely encode database images along a 1.1 km flight path for fast location querying using a global NAVCAM descriptor during flight. Similarly, [4] discretizes database images of a small UAV flight area along a grid prior to encoding and perform pose refinement via learned local feature matching after reducing location uncertainty via global descriptor matching. [18] also uses global descriptor matching, but requires onboard storage and preprocessing of georeferenced database images before encoding to achieve illumination and viewpoint invariance. In our work, we use global descriptors with discovered landmarks to perform localization with a low storage overhead to enable navigation in larger areas.

3.4 Approach

An overview of our VTRN pipeline (Fig. 3.2) is as follows: Prior to flight, landmarks are discovered in georeferenced images using a CNN, encoded with another CNN and tagged with geocoordinates, and cached in an onboard landmark database. During flight, landmarks from NAVCAM images are detected, encoded, and queried against database encodings to find a match. UAV position is updated, and position uncertainty is used to restrict the database search space.

We give an overview of the self-supervised learning scheme we use for CNN training before detailing the landmark discovery, encoding, and matching components that comprise our proposed method.

Self-supervised Contrastive Learning

We use a self-supervised contrastive learning (SSCL) scheme similar to [5]. Our training procedure is as follows: for an image $\mathbf{x}$, we generate two views $\tilde{\mathbf{x}}_\mathbf{i}, \tilde{\mathbf{x}}_\mathbf{j}$ via random visual perturbations commonly encountered in flight. The views are encoded into 128-d vectors, $\mathbf{h_i}, \mathbf{h_j}$, via CNN encoder f. We maximize the cosine similarity between positive vector pairs (generated from the same $\mathbf{x}$) and minimize between negative pairs (sampled from within the batch). For a positive pair, the loss is formally defined

$$\mathcal{L}(\mathbf{h_i}, \mathbf{h_j}) = -\log \frac{\exp(S_c(\mathbf{h_i}, \mathbf{h_j})/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(S_c(\mathbf{h_i}, \mathbf{h_k})/\tau)}, \tag{3.1}$$

where S_c denotes cosine similarity and τ is a temperature parameter set to 0.1.

Landmark Discovery, Encoder, and Matching

Landmark discovery: Our algorithm uses the activations of an HRNet CNN feature extractor [15] to find landmarks. The HRNet is followed by a global average pool and a fully-connected layer. To focus the network on invariant landmarks, we train this network (Sec. 3.4) to predict if two image encodings describe the same location (positive) or not (negative). We create image pairs using random seasonal variations (leaf-on and leaf-off), rotation, perspective, color jitter, and motion blur augmentations.

To localize landmarks, we extract the final activations from the three lowest-resolution streams of the HRNet (Fig. 3.2), denoted F2, F3, F4 from highest to lowest resolution. Each activation is channel-wise averaged before upsampling to input size and binarized via thresholding. Thresholds are chosen based on percentile values computed over training set activations. Landmarks are localized via contour detection and fitted with a tight bounding box (Fig. 3.1). Overlapping landmarks with an intersection-over-union (IoU) ≥ 0.4 are non-maximum suppressed (NMS), with preference for landmarks extracted at higher thresholds.

Landmark encoder: We use a Resnet-18 [8] encoder to encode discovered landmarks for lightweight storage and matching. This network is trained on discovered landmark crops (300×300) using the same augmentations as before.

Landmark matching: Prior to flight, landmarks are detected over a target area, encoded, and cached with their associated geocoordinates. During flight, we match NAVCAM landmark encodings against database encodings within R meters of the current position estimate. A landmark pair is a match if its similarity is over a

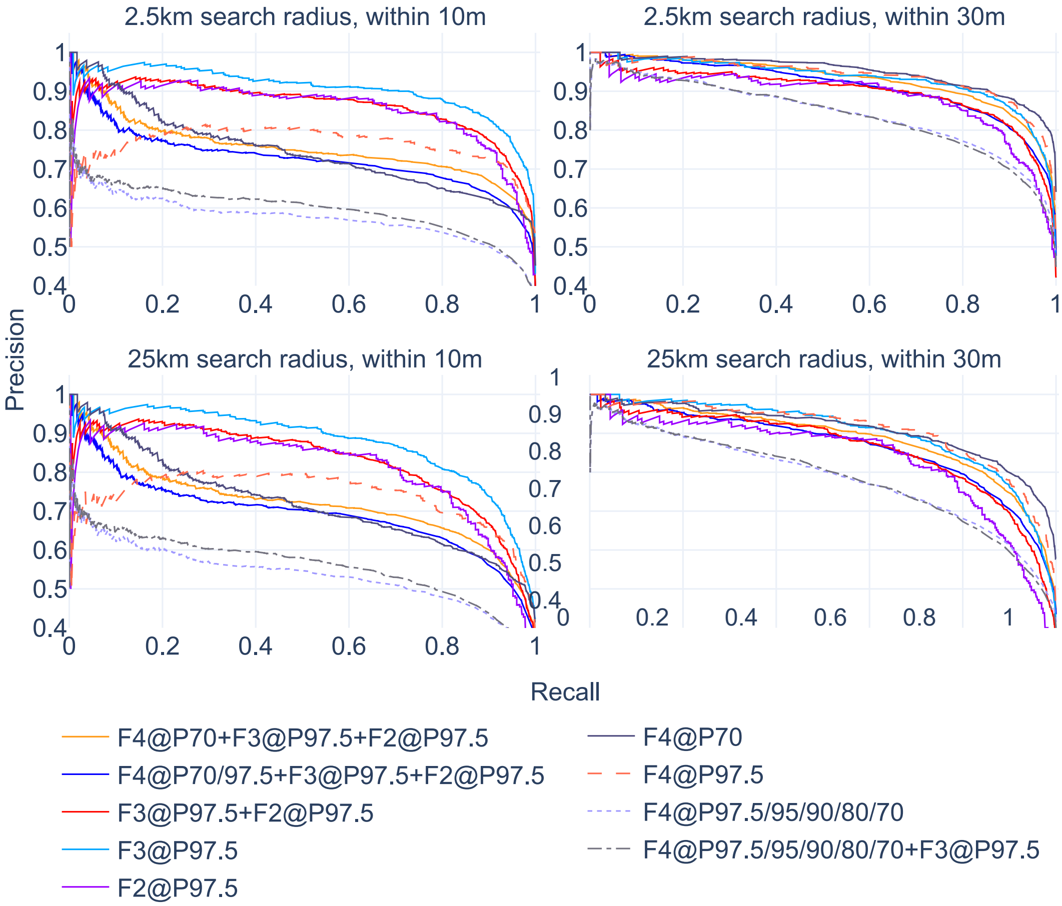


Figure 3.3: Joint landmark discovery and encoding matching results. Matching was done with 2.5 km and 25 km search radii. Ground truth matches were counted if a landmark pair were within 10 m or 30 m of one another.

threshold and has the maximum similarity over other possible pairs. R is hardcoded in this work, but we note that it could possibly be adapted based on current position uncertainties.

Implementation and Training Details

We implement networks in PyTorch using the `timm` library [17]. For landmark discovery, convolutions used reflection padding to avoid border effects in activation maps. We train for 1000 epochs, using a batch size of 128 and the Adam optimizer with a learning rate of $1e^{-4}$.

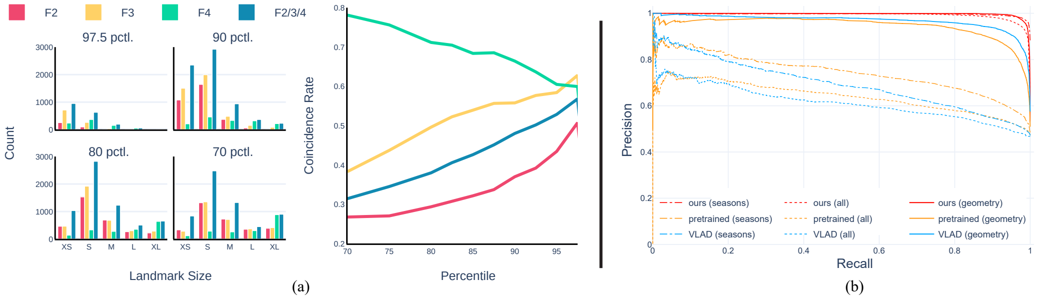


Figure 3.4: Ablation on landmark discovery and encoding using the Connecticut test set. (a) Metrics of coinciding landmarks discovered at different resolution streams and various percentile threshold values. (b) Effect of common VTRN perturbations (*seasons only*, *geometry only*, *all*) on the encoding similarities of known matching and non-matching landmark pairs.

3.5 Results

Datasets

We train and evaluate our method using the aerial image dataset from [7]. It consists of 3639 coregistered image pairs taken over the state of Connecticut (CT) in the United States during Spring and Summer 2016. Human-made structures, wooded forests, agricultural fields, and bodies of water are present, with “leaf-on vs. leaf-off” seasonal variation. We partition each 1270×1270 image into 600×600 crops with a 10 percent overlap and create train, val, and test splits at a 70:15:15 ratio. Each image has a resolution of 0.6m/pixel, resulting in 2112 km^2 of total landmass covered.

Seasonal effects like snow cover is not captured over this landmass and we leave explicit training and analysis of winter seasonal-invariance for future work. Also, we chose this setting with intuitive landmarks like buildings to easily validate our discovery algorithm, as the landmarks it proposes should overlap with those obvious to humans. In future work, we look to extend to less intuitive settings.

Localization Performance

Evaluation method: We evaluate matching performance of our VTRN pipeline via precision-recall analysis and use database search radii R of 2.5 and 25 km, simulating local (relatively lost) and global localization (completely lost) scenarios, respectively. Landmark pairs with maximum cosine similarities are considered as proposed matches in the evaluation. Distances between such landmark pairs are computed using the UTM coordinates at their bounding box centers and we consider ground truth matches to be within 10 and 30 m. Finally, similarity thresholds are applied to generate the precision-recall curves.

We test different configurations of resolution streams and percentile thresholds (Fig. 3.3) and find that small, highly salient landmarks (F2@P97.5, F3@P97.5) provide reliable position estimates within 10 m of ground truth, especially when paired with tighter search radii. Within 30 m of ground truth, using larger landmarks (F4@P70, F4@P97.5) yields best match rates. Aggregating landmarks from various resolution streams and thresholds does not achieve best performance but has the benefit of more landmarks for more location updates.

Ablation Studies

Landmark discovery: We quantify the number and size of the coinciding landmarks discovered using various resolution streams and percentile thresholds (Fig. 3.4a). In general, masking HRNet activations using high percentile thresholds (P97.5) increases the rate of landmark coincidence and favors small, sparse landmarks. Landmarks are generally easier to match when fewer but more salient landmarks are used, apart from very large landmarks (F4@P70).

Landmark encoding: Our landmark encoder outperforms other common image descriptors when faced with geometric and seasonal perturbations (Fig. 3.4b). We conduct precision-recall analysis by attempting to distinguish the encodings of an equal number of known matching and non-matching landmark pairs. We compare against ImageNet encodings (512-d) and VLAD [1] descriptors (2048-d). All encodings performed well with random geometrically-transformed landmarks from the same season, but only our encoding method was robust when seasons were varied in each pair, illustrating the benefit of explicitly training for such perturbation.

Computation Benchmarks

We benchmark landmark discovery and encoding using 600×600 NAVCAM images. Using a Nvidia Titan RTX and an Intel Core i9-7900X, images can be processed at 17 Hz. Benchmarks on an Nvidia Jetson AGX Orin, simulating small UAV use cases, sees slower rates of 8 Hz, due to slower processing during the CPU portions of the landmark localization step. We note that significant speed improvements can be made with smaller network architectures.

As our landmarks are sparse and low-dimensional, our method usually requires less onboard storage compared to techniques that densely encode a flight area [2, 4] or require onboard reference orthoimagery [3, 7, 13, 18]. For example, to localize over Salisbury, CT, which covers 155 km^2 of farmland, forests, and suburbs, 1.5 Gb is needed to store high-resolution, orthorectified reference images (assuming 0.6 m/pixel NAIP imagery) for methods that use database images during flight. Furthermore, an encoder-based method that lacks landmark detection like [2] would require roughly 19 Gb (extrapolated from their benchmarks). In contrast, our method requires between 90 to 200 Mb for the same landmass depending on configuration.

3.6 Conclusion

We presented the first landmark discovery algorithm for aerial VTRN. We showed that SSCL can find optimal landmarks for aerial navigation without human guidance and can consistently re-identify them across seasons. In conjunction with a seasonally-invariant CNN encoder, our discovery algorithm proposes landmarks that enable robust localization capabilities over large landmasses while demanding much less storage memory required by other methods. For future work, we aim to leverage our approach to better utilize sparse local features for more precise localization and pose estimation, integrate into a state estimation pipeline for UAV flight, and test in rugged mountainous and desert terrain where landmark selection is not as intuitive for humans.

3.7 Acknowledgement

This project was funded by the Boeing Company with R. K. Li as Boeing Project Manager. The authors thank R. K. Li and A. Tolstov of The Boeing Company for stimulating technical discussions.

References

[1] R. Arandjelovic and A. Zisserman. “All about VLAD”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 1578–1585.

[2] M. Bianchi and T. D. Barfoot. “Uav localization using autoencoded satellite images”. In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 1761–1768.

[3] J. R. Carr and J. S. Sobek. “Digital Scene Matching Area Correlator (DSMAC)”. In: *Image Processing for Missile Guidance*. Ed. by T. F. Wiener. SPIE, 1980. DOI: 10.1117/12.959130.

[4] S. Chen, X. Wu, M. W. Mueller, and K. Sreenath. “Real-time Geo-localization Using Satellite Imagery and Topography for Unmanned Aerial Vehicles”. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots*

and Systems. 2021, pp. 2275–2281. DOI: 10.1109/IROS51168.2021.9636705.

[5] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. “A simple framework for contrastive learning of visual representations”. In: *Int. Conf. Mach. Learning*. PMLR. 2020, pp. 1597–1607.

[6] L. Downes, T. J. Steiner, and J. P. How. “Deep learning crater detection for lunar terrain relative navigation”. In: *AIAA SciTech 2020 Forum*. 2020, p. 1838.

[7] A. T. Fragoso, C. T. Lee, A. S. McCoy, and S.-J. Chung. “A seasonally invariant deep transform for visual terrain-relative navigation”. In: *Science Robotics* 6.55 (2021), eabf3320. DOI: 10.1126/scirobotics.abf3320. eprint: https://www.science.org/doi/pdf/10.1126/scirobotics.abf3320.

[8] K. He, X. Zhang, S. Ren, and J. Sun. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 770–778.

[9] A. M. Hussain Ismail, J. A. Solomon, M. Hansard, and I. Mareschal. “A perceptual bias for man-made objects in humans”. In: *Proceedings of the Royal Society B* 286.1914 (2019), p. 20191492.

[10] A. E. Johnson, S. B. Aaron, H. Ansari, C. Bergh, H. Bourdu, J. Butler, J. Chang, R. Cheng, Y. Cheng, K. Clark, et al. “Mars 2020 Lander Vision System Flight Performance”. In: *AIAA SciTech 2022 Forum*. 2022, p. 1214.

[11] M. Kwon, E. Biyik, A. Talati, K. Bhasin, D. P. Losey, and D. Sadigh. “When Humans Aren’t Optimal: Robots that Collaborate with Risk-Aware Humans”. In: *2020 15th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. 2020, pp. 43–52.

[12] L. Matthies, S. Daftry, S. Tepsuporn, Y. Cheng, D. Atha, R. M. Swan, S. Ravichandar, and M. Ono. “Lunar rover localization using craters as landmarks”. In: *2022 IEEE Aerospace Conference (AERO)*. IEEE. 2022, pp. 1–17.

[13] A. Nassar, K. Amer, R. ElHakim, and M. ElHelw. “A deep cnn-based framework for enhanced aerial imagery registration with applications to uav geolocalization”. In: *Proc. IEEE Conf. Comput. Vis. Pattern Recog. workshops*. 2018, pp. 1513–1523.

[14] J. Vander Hook, R. Schwartz, K. Ebadi, K. Coble, and C. Padgett. “Topographical landmarks for ground-level terrain relative navigation on mars”. In: *2022 IEEE Aerospace Conference (AERO)*. IEEE. 2022, pp. 1–6.

[15] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, et al. “Deep high-resolution representation learning for visual recognition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.10 (2020), pp. 3349–3364.

[16] T. Wang, Y. Zhao, J. Wang, A. K. Somani, and C. Sun. “Attention-Based Road Registration for GPS-Denied UAS Navigation”. In: *IEEE Transactions on Neural Networks and Learning Systems* 32.4 (2021), pp. 1788–1800. DOI: 10.1109/TNNLS.2020.3015660.

[17] R. Wightman. *PyTorch Image Models*. https://github.com/rwightman/pytorch-image-models. 2019. DOI: 10.5281/zenodo.4414861.

[18] P. Yin, I. Cisneros, S. Zhao, J. Zhang, H. Choset, and S. Scherer. “iSimLoc: Visual Global Localization for Previously Unseen Environments With Simulated Images”. In: *IEEE Transactions on Robotics* (2023), pp. 1–17. DOI: 10.1109/TRO.2023.3238201.

Chapter 4

ONLINE SELF-SUPERVISED THERMAL WATER SEGMENTATION FOR AERIAL VEHICLES

[1] C. Lee, J. G. Frennert, L. Gan, M. Anderson, and S.-J. Chung. "Online Self-Supervised Thermal Water Segmentation for Aerial Vehicles". In: *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2023, pp. 7734–7741. DOI: 10.1109/IROS55552.2023.10342016.

4.1 Abstract

We present a new method to adapt an RGB-trained water segmentation network to target-domain aerial thermal imagery using online self-supervision by leveraging texture and motion cues as supervisory signals. This new thermal capability enables current autonomous aerial robots operating in near-shore environments to perform tasks such as visual navigation, bathymetry, and flow tracking at night. Our method overcomes the problem of scarce and difficult-to-obtain near-shore thermal data that prevents the application of conventional supervised and unsupervised methods. In this work, we curate the first aerial thermal near-shore dataset, show that our approach outperforms fully-supervised segmentation models trained on limited target-domain thermal data, and demonstrate real-time capabilities onboard an Nvidia Jetson embedded computing platform.

4.2 Introduction

Water segmentation is advantageous for uninhabited aerial vehicles (UAV) operating in near-shore environments. It can enable GPS-denied visual navigation [40], and assist tasks such as bathymetry [8]. However, current water segmentation algorithms operate on color (RGB) images and do not work well at night. Thermal cameras, on the other hand, can highlight details in conditions in which color cameras fail. In this paper, we look to develop a thermal water segmentation algorithm to bring autonomous nighttime capabilities to aerial robotics operating in near-shore settings like rivers, lakes, and coastlines.

Compared to RGB water segmentation, which has been well studied in context of uninhabited surface vehicles (USV) [6, 41, 7], thermal water segmentation has received little attention. As result, it lacks data, especially from aerial platforms, which prevents modern, state-of-the-art convolutional neural networks (CNN) from being easily applied. Moreover, three problems make it difficult to collect an aerial thermal near-shore data diverse enough for CNN training: Water bodies often coincide with no-fly zones; municipal-specific permits are required for non-recreational UAV usage; and distinct bodies of water are geographically dispersed, slowing diverse dataset collection for training and validation.

Aside from dataset limitations, thermal imagery is out-of-distribution relative to RGB imagery. As such, harnessing RGB data for thermal model training requires domain adaptation. Due to ongoing interest in self-driving cars, RGB-thermal domain adaptation has been well explored in urban settings [22, 3, 24, 10, 34, 44]. However, such works still require training over target thermal data. Because we lack aerial, near-shore thermal data, we cannot effectively apply existing domain adaptation methods.

In this work, we propose a thermal water segmentation algorithm for UAVs that adapts to incoming thermal images during flight. Our contributions are as follows:

1. We present an online self-supervised approach that uses thermal water cues to adapt a RGB-pretrained water segmentation network to the near-shore thermal domain during flight.

2. We demonstrate superior performance on aerial near-shore datasets compared to baselines.

3. We present ablation studies of our self-supervision cues and test different RGB pretraining methods to assist online thermal adaptation.

4. We release our algorithm as a Robot Operating System [35] (ROS) package and demonstrate real-time online training and inference on a Nvidia Jetson AGX Orin.

5. We release an annotated thermal water segmentation dataset, capturing aerial and ground near-shore settings, to bootstrap future work in this area.

4.3 Related Work

RGB water segmentation: RGB water segmentation methods typically leverage various combinations of water appearance, reflections, and location priors to segment water. To detect water for uninhabited ground vehicle (UGV) navigation, [37]

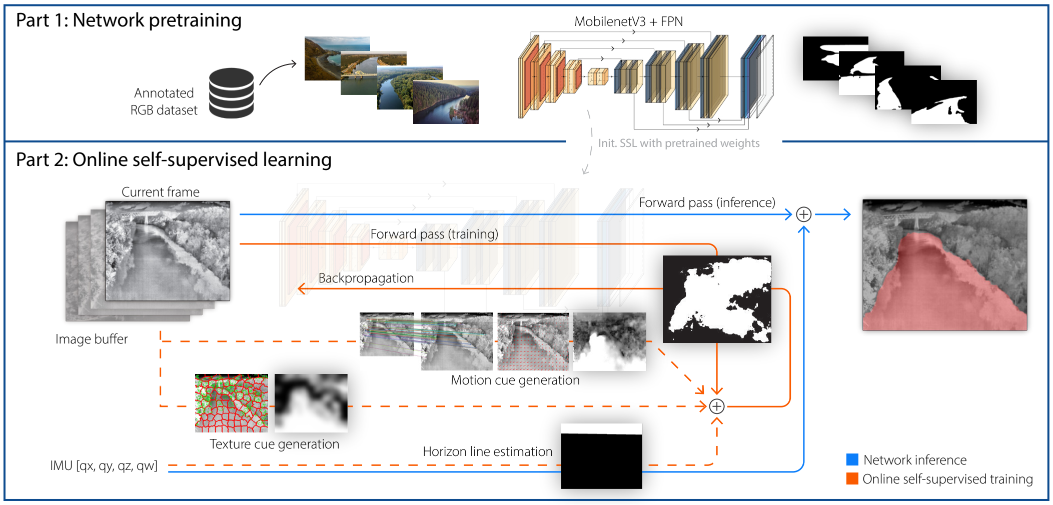


Figure 4.1: Schematic of our proposed online self-supervised training for thermal water segmentation. A few online training loops are performed prior to network inference on the current scene.

fuses color, texture, stereo range, and horizon line cues to systematically identify true water pixels. [36] takes a similar approach but utilizes correlation between color and water reflections at predetermined distances to segment water. In river settings, [28] exploits the shallow viewing angle of USVs to estimate and segment the river plane via water reflection symmetry. In contrast to these works, our aerial application precludes the use of water reflections as they are less prominent in the thermal domain and less accentuated at higher altitudes. We use texture cues and horizon line estimation in this work, but not color, as hue and saturation do not exist in thermal data.

Other water segmentation algorithms leverage geometric priors based on their target setting and vehicle. For USV navigation, [2] trains an online self-supervised classifier to segment rivers, by assuming that shore regions above the horizon line are similar to shore regions below. Their assumption does not extend to our UAV setting, however, as the horizon line is not necessarily always in the frame due to aircraft pitch. [23] proposes an obstacle segmentation algorithm for USVs in maritime environments, using the horizontal stacking of water, land/horizon, and sky as location priors for components in a Gaussian Mixture Model. However, these location priors are exclusive to USVs in maritime environments where shoreline generally isn't visible on either sides and the camera is always near the water surface. [7] extends this by incorporating inertial measurement unit-based (IMU) horizon line estimates into the location priors and enforcing stereo constraints for obstacle detection.

Recent deep learning-based approaches leverage small annotated RGB water datasets to train supervised CNN segmentation models [6, 41, 5]. These methods are more robust and require less parameter tuning compared to earlier approaches such as [37], but require large, i.i.d datasets to properly train. As segmentation datasets are expensive and time-consuming to assemble, these works focus on maximizing generalization performance on existing, but limited, water segmentation datasets.

Thermal water segmentation: Little work has been done in this domain. RGB images have been used directly as input to train CNNs for thermal maritime obstacle segmentation [30] but were found to perform poorly compared to training on maritime thermal imagery [29]. To the best of our knowledge, this is the only public thermal water segmentation dataset that exist today and was released around the time of our work. However, this dataset targets USV maritime environments and does not yield good performance on our aerial near-shore data (Table 4.3). As aerial thermal near-shore datasets do not yet exist, we opt for an online self-supervised approach in this work instead of these offline supervised methods. However, we do leverage such methods for network pretraining and further improve thermal segmentation performance via online self-supervised learning.

RGB-thermal domain adaptation: RGB-thermal (RGB-T) domain adaptation (DA) has been well studied in urban environments due to their applications to self-driving cars and have been used to harness large RGB datasets in conjunction with thermal data to train thermal networks. Generative methods like image translation have been used to automatically synthesize fake thermal imagery with labels from annotated RGB datasets for thermal model training [24, 44]. However, they are known to hallucinate and introduce spatial structures that don't appear in the target domain [21]. Unsupervised domain adaptation (UDA) methods like [22] and [18] seek to align RGB and thermal CNN features by training with shared weights and adversarial losses using annotated RGB data and unlabeled thermal data. Because public datasets of aerial thermal near-shore settings did not exist at the time of this work, we use collected aerial thermal near-shore data for validation and operate under the assumption of having no target domain data available.

Self-supervised learning (SSL): In SSL, labels are automatically generated from data rather than from annotation. SSL is a broad topic and has been used in offline settings with applications including monocular road detection [46, 15, 13], terrain traversability [39], and general representation learning [11, 12]. It has also been used to adapt semantic segmentation networks to out-of-distribution data by enforcing augmentation consistency via a momentum network [4] which we leverage in our work.

SSL can be applied online to adapt to new environments: For RGB river segmentation, [2], as previously discussed, uses visual cues with horizon line river priors to create training patches for online classifier training. [42] creates training labels for a river segmentation network by assigning labels to unsupervised segmentation output based on the response of an onboard LiDAR sensor. [14] performs online SSL on lightweight CNNs using stereo information for ground plane segmentation and is most related to our work. In this work, we use online SSL to adapt an RGB-pretrained water segmentation network to the aerial thermal near-shore domain by generating labels from water texture and motion cues, and horizon line estimates.

4.4 Method

We develop a thermal water segmentation method for UAVs that does not see thermal data prior to test time. Our method adapts an RGB-pretrained CNN segmentation model with online SSL to compensate for RGB-T covariate shift (Fig. 4.1). Self-supervised labels are generated by exploiting texture and motion differences between land and water. To increase robustness, we utilize IMU-based horizon line estimation to remove false positive water pixels in the sky and use an alternative CNN-based sky segmentation when IMU data is corrupted or unavailable. We now outline our preprocessing procedure for 16-bit thermal images, RGB-based network pretraining method, self-supervised label generation process, and the online learning algorithm.

Thermal Image Preprocessing

Raw 16-bit thermal images are contrast stretched using the 1st and 99th percentile pixel values and followed by Contrast Limited Adaptive Histogram Equalization [32]. In Sec. 4.4, image pairs are stretched with the maximum of the 2nd percentile and the minimum of the 98th.

Segmentation Network Pretraining

We pretrain a segmentation network to speed up online training convergence. As UAVs are resource-constrained, we choose a compute-efficient Feature Pyramid Network (FPN) built on a MobileNetV3-small backbone with 2.3 million parameters [25, 19]. The network takes in $1 \times H \times W$ images and outputs $2 \times H \times W$ class probability maps.

Table 4.1: Network pretraining dataset breakdown.

Dataset	# Train	# Val.	Water-Related Indices
ADE20k	1743	163	22, 27, 61, 114, 129
COCO-Stuff	10,977	458	148, 155, 178, 179
River Dataset [26]	300	0	—
Flickr	1,220	0	—

We train the network using water-related RGB images from ADE20K [45], COCO-stuff [9], and a river segmentation dataset [26]. We supplement with Flickr images, found by querying keywords like *aerial river* and *drone ocean*, and annotated using an ADE20K-pretrained segmentation network from [45], resulting in 14,240 training images. Annotations are converted into *water* or *non-water* classes. Dataset-specific label indices and training set composition are shown in Table 4.1.

As thermal images are single-channel, we transform 3-channel RGB images into 1-channel grayscale between [0, 1] prior to training using one of these methods:

1. **Grayscale**: OpenCV's default RGB to grayscale conversion method.

2. **Random mix**: Weighted channel-wise mean with randomly selected weights. Random inversion is applied to simulate thermal temperature inversion.

3. **Random mix (PCA)**: RGB channels are decorrelated via principle components analysis (PCA). The first 2 channels are randomly mixed, normalized, and randomly inverted.

4. **RGB2Thermal**: RGB images are translated to thermal using contrastive unpaired translation [31] after training on the MassMIND thermal, MaSTr1325 [6], and our RGB dataset.

Self-Supervision from Texture Cues

Given image I, we create a soft water/non-water label for online SSL (Fig. 4.2b) by observing that water tends to have less texture compared to surrounding land. We first perform unsupervised segmentation on I via Simple Linear Iterative Clustering, creating superpixels similar in shape and size [1]. Each image pixel is assigned a class label based on the texture of the encompassing superpixel. We quantify texture using the Difference-of-Gaussian (DoG) keypoint detector [27] and compute

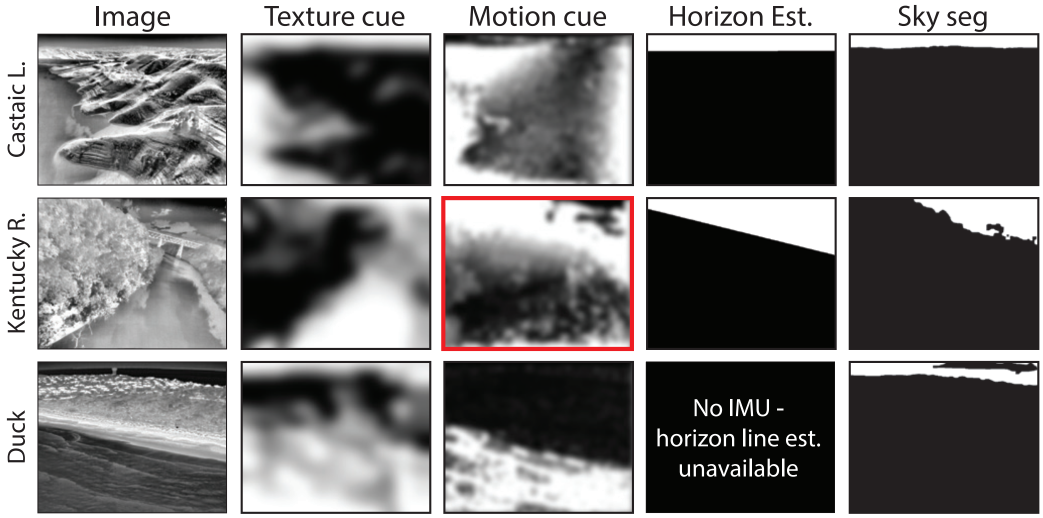


Figure 4.2: Texture, motion, horizon, and sky segmentation cues used for online SSL. Motion cue failure case outlined in red.

a probability map of non-water pixels

$$P^T_{\neg W}(i,j) = \left(G \circledast \frac{S_{kp}}{\alpha_T}\right)[i,j] \tag{4.1}$$

by normalizing the keypoint count of each superpixel S_{kp} with a parameter α_T and smoothing with a Gaussian kernel G. The probability of water pixels P^T_W is the inverse $1 - P^T_{\neg W}$. Although the DoG detector filters out edge responses, we further mitigate jagged edge responses, such as along river banks, by pruning keypoints within 2 pixels of superpixel boundaries.

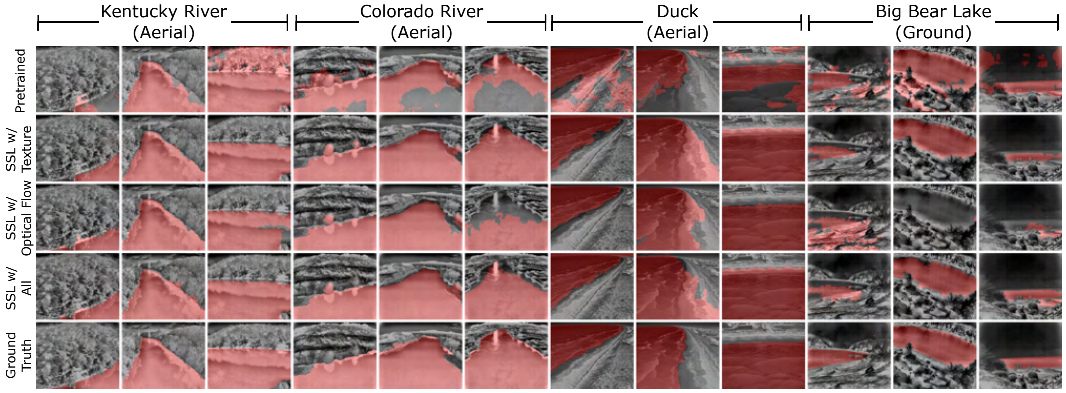


Figure 4.3: (a) Water segmentation results *(red)* from a pretrained RGB network; our method with texture, and motion, and both cues; and ground truth.

Self-Supervision from Motion Cues

In near-shore settings with fast flowing water like coastlines, water can appear choppy which breaks the assumption of the texture cue. However, this non-uniformity allows us to use water motion as another indicator of water pixels. We estimate water motion magnitude between successive image frames $I_{t-\Delta t}$ and I_t using a two-step process.

First, we discount UAV-attributed motion by aligning successive frames using feature-based image registration. We match ORB features [38] detected in $I_{t-\Delta t}$ and I_t and compute a homography matrix H using Random Sample Consensus [17]. Because camera pose does not change significantly between successive frames, matches should consist mainly of shore features. H is used to align the image coordinate frames via image warp $\mathcal{T}$ before cropping to the greatest common area. We then assume

$$I_t^{\text{crop}} \approx \mathcal{T}(I_{t-\Delta t}; H)^{\text{crop}} \tag{4.2}$$

effectively removing any UAV motion-induced transformations. We note that static shore regions must be in view in order to mitigate the risk of aligning based on water motion.

Second, to create a probability map of water pixels $P_W^F(x, y)$, we quantify water motion using Farneback's algorithm [16], $\mathcal{F}$, to compute the dense optical flow field

$$V=[V_x, V_y]^\top=\mathcal{F}\big(I_t^{\text{crop}},\mathcal{T}(I_{t-\Delta t}; H)^{\text{crop}}\big) \tag{4.3}$$

between the aligned image frame crops. We normalize the flow field magnitude by α_F to create a water probability map

$$P_W^F(x, y) = \begin{cases} \frac{\|V(x,y)\|_2}{\alpha_F}, & \text{if } x \in [x_1, x_2],\ y \in [y_1, y_2] \\ 0, & \text{otherwise} \end{cases} \tag{4.4}$$

and the non-water probabilities $P_{\neg W}^F$ arise as the inverse. We set α_F to be the 75th percentile of the flow magnitudes.

Discerning Sky and Water Pixels

Sky and water may both have little texture, causing issues for texture cues. To fix this, we use horizon line estimation or sky segmentation to correct any resulting false positive water labels.

Horizon line estimation with IMU data: Horizon (vanishing) line estimation allows us to unequivocally label all pixels above the horizon as non-water [7, 2].

We estimate it by projecting distant points $\mathbf{x}_\mathrm{u}$ in the UAV coordinate frame that lay within the camera's field-of-view, to image coordinates $\mathbf{x}_\mathrm{c}$ via the relation

$$\mathbf{x}_\mathrm{c} = \mathbf{P}_\mathrm{c}\mathbf{R}_\mathrm{i}^\mathrm{c}\,\mathbf{R}_\mathrm{u}^\mathrm{i}\mathbf{x}_\mathrm{u}, \tag{4.5}$$

where $\mathbf{R}_\mathrm{u}^\mathrm{i}$, $\mathbf{R}_\mathrm{i}^\mathrm{c}$ represent rotation matrices from UAV-to-IMU and IMU-to-camera, and $\mathbf{P}_\mathrm{c}$ is the camera projection matrix. We find the horizon by fitting a line to $\mathbf{x}_\mathrm{c}$ and take all pixels above to be *non-water*.

Sky segmentation without IMU data: In situations where IMU data is not accessible, we use a lightweight Fast-SCNN [33] segmentation network to quickly segment the sky. We remove the second bottleneck layer in the feature extractor to reduce computational burden. We train on MassMIND [29], KAIST Pedestrian [20] with segmentation labels from [22], SODA [24], and FLIR aligned [43] data after reducing annotations to *sky* and *not-sky*.

The FLIR aligned dataset does not have segmentation annotations. To create them, we segment FLIR RGB images using the same pretrained RGB network used to label Flickr images (Sec. 4.4), creating sky masks. As the masks may be rough, and sky is *usually* the coldest part in a thermal image, we refine each mask by binary searching the corresponding 14-bit thermal pixel values for a threshold that generates a new mask whose area falls within 10 % of the RGB mask's area. We visually inspect the results and retain 4,201 annotations out of 5,142 for sky segmentation training.

Online Training

To perform online training (Algorithm 1), we initialize our pretrained segmentation network f from Sec. 4.4. We freeze the encoder and the first two decoder blocks to reduce trainable parameters. Like [4], we initialize a separate momentum network g which is a copy of f. Network g generates a soft self-label P^g that is improved by ensembling with other cues and is updated with f's weights at a rate of $\lambda = 0.3$ after every training loop.

We adapt to the current scene by performing N training iterations using images from a buffer that holds the past L images seen, including the current image frame I_t. For each image in the buffer, we find the horizon line or sky segmentation depending on IMU availability, and generate our self-supervised labels. We merge the labels with P^g using a per-class weighted average

$$y_\text{water} = w_1 P^g_W + w_2 P^F_W + w_3 P^T_W \tag{4.6}$$

$$y_\text{non-water} = \xi_1 P^g_{\neg W} + \xi_2 P^F_{\neg W} + \xi_3 P^T_{\neg W} \tag{4.7}$$

Algorithm 1 Online Training and Inference

```
Input: Network weights $\theta$, Image buffer length $L$
Output: Segmentation masks $M_{0,1,\ldots t}$
Initialize: Networks $f_\theta$, $g_\theta$, Image buffer $Q$

while camera on do
    Grab current image frame $I_t$
    Add $I_t$ to $Q$ and remove $I_{t-\Delta t L}$ if exists

    if train at time $t$ then
        $\mathcal{D} \leftarrow$ CreateBatches($Q$)
        for $n = 1 : N$ do
            Sample batch $(I_n, P^F_{W,n}, P^T_{W,n})$ from $\mathcal{D}$
            $P^g_n \leftarrow g(I_n)$
            $y_n \leftarrow$ MergeLabels($P^g_n, P^F_{W,n}, P^T_{W,n}$)                ▷ Eq. 4.6-4.7
            $\theta_f \leftarrow \theta_f + \gamma \nabla_{\theta_f} \mathcal{L}_{\text{bce}}\,(f(I_n), y_n)$
        end for
        $\theta_g \leftarrow \lambda \cdot \theta_f + (1 - \lambda) \cdot \theta_g$                ▷ Momentum update
    end if
    $P^f_t \leftarrow f(I_t)$                                  ▷ Inference on current frame
    Apply channel-wise argmax on $P^f_t$ to create $M_t$
    yield $M_t$
end while
```

and mark locations of sky pixels, or those above the horizon line, as definitively non-water. Network f is trained on these soft labels using the binary cross entropy loss $\mathcal{L}_{\text{bce}}$.

$$\mathcal{L}_{\text{bce}}(y, \hat{y}) = \sum_{i,j,k} y \cdot \log \hat{y} + (1 - y) \cdot \log(1 - \hat{y}) \tag{4.8}$$

During online training, images of size 512 × 640 are randomly cropped to 320 × 320 and subject to random horizontal flips. After N training iterations, we perform inference on I_t using f to get P^f and apply a channel-wise argmax to create segmentation mask M_t. We clean up the mask using morphological operations and keep the largest segmented contour as water. When IMU data is available, we also remove any water pixels still present above the horizon line. Lastly, we note that it is not necessary to perform online training prior to every inference call as image frames within a narrow time window are very similar.

4.5 Results

Dataset

Our dataset consists of aerial and ground thermal sequences covering river, coastal, and lake scenery (Table 4.2) captured in 16-bit[1] using a FLIR ADK long-wave

[1] The data from Duck, NC was captured in 8-bit format using a separate sensor stack and does not have IMU information available.

Table 4.2: Thermal river, lake, and coastal datasets

Dataset	Near-shore Category	Capture Method	# Images	# Annot.	# Seq.
Kentucky River, KY	River	UAV Flight	7826	94	1
Colorado River, CA	River	UAV Flight	84,993	659	2
Duck, NC[†]	Coast	UAV Hover	4143	68	7
Castaic Lake, CA	Lake	UAV Flight	101,999	128	2
Big Bear Lake, CA	Lake	Ground	48,676	282	8
Arroyo Seco, CA	Stream	Ground	7	7	—

[†] Captured and stored in processed 8-bit data.

thermal camera (Fig. 4.3, 4.4). These datasets are provided as ROS bag files, and also as individual frames with synchronized IMU and geolocation data for convenience. Frames were sampled for annotation at 2 second intervals, but at 12 second intervals for lengthy sequences from Castaic Lake. Some frames were skipped, at annotators' discretion, if indistinguishable to preceding frames. A single frame was used per Arroyo Seco sequence due to minimal change in each recording.

Aerial sequences were used for experimental validation while ground sequences were used for training and ablations in non-target settings (see Table 4.4 for list of sequences). Overall, the locations are very distinct and the datasets consist of a rich variety of sun positions, shore topography, water body size and shape, and surrounding flora. Aerial data from the Kentucky River (near Shakers Ferry Rd), KY; Colorado River (near Parker Dam), CA; Castaic Lake, CA; and the coastline at Duck, NC, were nominally recorded between 40 and 50 m above the water surface. Lower altitude imagery was also captured to enlarge the dataset for use in

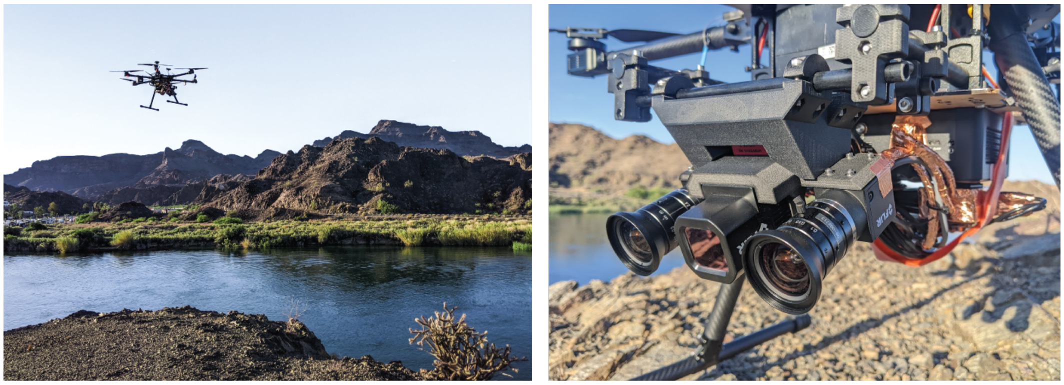


Figure 4.4: Our UAV operating over the Colorado River, CA, and the sensor stack mounted to the UAV showing the time-synchronized FLIR ADK thermal camera and VN100 IMU. Visible light cameras were not used as part of this work.

future work. Ground-level datasets from Big Bear Lake, CA and the Arroyo Seco (Pasadena), CA feature much shallower viewing angles of water and scenes with reduced visibility due to fog.

Network Training Details

The RGB-pretrained network from Sec. 4.4 and the sky segmentation network from Sec. 4.4 were trained as follows: Training images were resized to a longest dimension of 512, rescaled between 0.5 and 2.0, and randomly cropped to 320 × 320. Random horizontal flips, rotations (within 10°), and color jitter followed prior to single channel conversion if needed. We used stochastic gradient descent (SGD) with a momentum of 0.9, a learning rate of 1×10^{-2}, L_2 weight decay of 1×10^{-4}, and a batch size of 32.

Thermal-trained networks used as baselines in Sec. 4.5 were trained using the same hyperparameters and data augmentations as above. However, 16-bit thermal images were first normalized using the thermal preprocessing technique described in Sec. 4.4, but contrast stretched with random low and high values bounded by the 5th and 95th percentiles.

Online Training Setup

For online training, we used the Adam optimizer with learning rate 1×10^{-3} and L_2 weight decay 1×10^{-4}. Batch normalization was turned off. To increase speed, we scaled down images by 0.5 prior to label generation. By default, each round of online training ran for $N = 8$ iterations with an 8 image buffer and a batch size of 4. Online training was performed every 120 frames and before each annotated frame, with every 4th image added into the buffer.

We set texture cue generation $\alpha_T = 10$ and set cue merging parameters $w = [1, 1, 1]$ and $\xi = [1, 0, 1]$. When a cue is not used, its corresponding weights are set to 0, e.g. $w = [1, 1, 0]$ and $\xi = [1, 0, 0]$ when only the motion cue is in use. We test networks initialized with weights via grayscale, random mixing, and PCA random mixing pretraining.

Performance Evaluation

We demonstrate our method's robustness and superiority in a no-data regime over standard supervised segmentation with limited data. We compare our online SSL method against five thermal segmentation networks by testing in the primary aerial near-shore settings of this work: river, lake, and coast. These baselines were trained

Table 4.3: Performance evaluation of our online method in target aerial settings compared to fully-supervised networks trained with limited thermal data.

Method	Training Set	Aerial Test Setting mIoU		
		River	Lake	Coast
MobilenetV3 + FPN	Arroyo Seco	0.619	0.560	0.583
MobilenetV3 + FPN	Big Bear Lake	0.687	0.526	0.638
MobilenetV3 + FPN	Big Bear Lake + Arroyo	0.794	0.630	0.719
MobilenetV3 + FPN	Colorado River	—	0.745	0.436
MobilenetV3 + FPN	MassMIND [29]	0.454	0.310	0.445
Online SSL (Grayscale) + TC	—	**0.902**	0.909	0.623
Online SSL (Grayscale) + MC	—	0.451	0.275	0.713
Online SSL (Grayscale) + All	—	0.885	**0.911**	0.668
Online SSL (Rand. Mix) + TC	—	0.900	0.891	0.617
Online SSL (Rand. Mix) + MC	—	0.482	0.275	0.726
Online SSL (Rand. Mix) + All	—	0.884	0.904	0.659
Online SSL (PCA) + TC	—	0.895	0.889	0.611
Online SSL (PCA) + MC	—	0.474	0.746	**0.805**
Online SSL (PCA) + All	—	0.878	0.909	0.654

TC – Texture Cue MC – Motion Cue

Table 4.4: Near-shore water segmentation ablation in different thermal sequences.

Setting	Dataset Sequence	PT	PT + Self-Train	TC Only	MC Only	w/o Sky Seg. nor Horizon Est.			w/ Sky Segmentation			w/ Horizon Est.		
						PT + TC	PT + MC	PT + All	PT + TC	PT + MC	PT + All	PT + TC	PT + MC	PT + All
Aerial River	Kentucky River 2-1	0.700	0.528	0.787	0.506	0.859	0.809	0.834	0.881	0.797	0.860	**0.884**	0.810	0.857
	Colorado River 1	0.500	0.453	0.796	0.476	0.894	0.295	0.881	0.897	0.295	0.884	**0.898**	0.295	0.886
	Colorado River 3	0.513	0.690	0.798	0.440	0.886	0.315	0.881	0.898	0.315	0.888	**0.902**	0.317	0.892
	Avg. Seq. mIoU	0.571	0.557	0.794	0.474	0.880	0.473	0.865	0.892	0.469	0.877	**0.895**	0.474	0.878
Aerial Lake	Castaic Lake 2	0.324	0.241	0.830	0.521	0.901	0.701	0.911	0.804	0.227	0.826	0.886	0.703	**0.918**
	Castaic Lake 4	0.552	0.495	0.775	0.417	0.876	0.790	0.876	0.890	0.322	0.889	0.893	0.789	**0.900**
	Avg. Seq. mIoU	0.438	0.368	0.802	0.469	0.889	0.746	0.893	0.847	0.275	0.857	0.889	0.746	**0.909**
Aerial Coast	Duck 4	0.799	0.915	0.366	0.541	0.448	**0.933**	0.469	0.499	0.693	0.499	—	—	—
	Duck 5	0.347	0.854	0.674	0.870	0.792	**0.931**	0.859	0.375	0.500	0.456	—	—	—
	Duck 6	0.519	**0.842**	0.500	0.683	0.506	0.832	0.562	0.460	0.601	0.517	—		—
	Duck 10	0.743	**0.782**	0.489	0.551	0.457	0.740	0.461	0.493	0.552	0.498		**No IMU**	
	Duck 13	0.300	0.260	0.546	0.164	**0.579**	0.429	0.573	0.578	0.429	0.572	—		—
	Duck 14	0.532	0.454	0.770	0.684	0.758	**0.963**	0.817	0.758	**0.963**	0.817	—	—	—
	Duck 15	0.838	0.819	0.673	0.779	0.738	0.809	0.838	0.738	0.803	**0.839**	—	—	—
	Avg. Seq. mIoU	0.583	0.704	0.574	0.610	0.611	**0.805**	0.654	0.557	0.649	0.600	—	—	—
Ground Lake	Big Bear Lake 23	0.430	0.423	0.251	0.352	0.278	0.406	0.295	0.429	0.433	0.436	**0.785**	0.759	0.783
	Big Bear Lake 27	0.638	0.687	0.355	0.518	0.408	0.399	0.431	0.653	0.485	0.703	0.713	0.378	**0.749**
	Big Bear Lake 30	0.471	0.424	0.317	0.472	0.345	0.374	0.353	0.506	0.382	0.518	0.611	0.413	**0.613**
	Big Bear Lake 34	0.643	0.702	0.436	0.391	0.504	0.696	0.511	**0.857**	0.705	0.834	**0.857**	0.524	0.842
	Big Bear Lake 37	0.479	0.495	0.313	0.420	0.307	0.475	0.308	0.597	0.457	0.584	**0.743**	0.465	0.710
	Big Bear Lake 40	0.741	0.820	0.532	0.368	0.652	0.430	0.662	**0.854**	0.456	0.851	0.818	0.457	0.819
	Big Bear Lake 44	0.756	**0.846**	0.376	0.400	0.289	0.772	0.375	0.830	0.828	0.802	0.824	0.843	0.832
	Big Bear Lake 50	0.642	0.630	0.280	0.356	0.318	**0.719**	0.340	0.555	0.683	0.569	0.609	0.697	0.661
	Avg. Seq. mIoU	0.600	0.628	0.357	0.410	0.388	0.534	0.409	0.660	0.554	0.662	0.745	0.567	**0.751**

PT – Base Pretrained Network TC – Texture Cue MC – Motion Cue on the recently released MassMIND thermal USV segmentation dataset, the thermal ground-based data from Table 4.2, and the aerial Colorado River dataset (Table 4.3).

The baseline performances confirm our suspicions of poor generalization capabilities and overfitting due to limited dataset size, diversity, and covariate shift from surface/ground to aerial (Table 4.3). Notably, neither networks trained on ground-level lake data (Big Bear Lake + Arroyo) nor networks trained on aerial river data (Colorado River) perform well when moving to aerial lake, and the lackluster performance of the MassMIND-trained network in these settings further motivates the collection and curation of aerial thermal datasets for nighttime UAV applications.

In contrast, we report strong evidence favoring our texture- and motion-based online SSL over the fully-supervised networks: All three online variants using texture-based adaptation attain roughly 0.9 mIoU, outperforming the best thermal-trained networks in the aerial river (0.794 mIoU) and lake (0.745 mIoU) domains (Table 4.3). Motion-based online adaptation performs best in the aerial coastal setting where wave motion and currents are highly visible. Here, the PCA-initialized variant outperforms the best thermal supervised network by a 0.08 margin, while the other two variants match performance. None of the motion-based variants perform well in rivers and lakes likely due to calmer waters. Likewise, texture-based cues do not perform well in coastal scenes due to confusion with highly-textured, fast-moving waves. Lastly, we see no significant advantages in leveraging both cues at the same time: river and lake settings see minor improvements while coastal settings see a performance drop. We finally note that these observations could be used to select suitable weights for cue merging (Eq. 4.6-4.7) during mission planning for operations in known near-shore environments.

Ablation Study

Influence of online SSL: Using the overall best online model (PCA) from Sec. 4.5, we analyze the role of the texture and motion cues in the aerial near-shore settings (Table 4.4). Overall, we find that cues do not provide adequate segmentation when used alone (Table 4.4, TC/MC Only) and should be used to adapt a pretrained network as intended. They are necessary for online training robustness as self-training alone (PT+Self-Train) performs inconsistently. Our SSL cues, when used in appropriate online settings, i.e. texture-based with river/lake and motion-based with coastal, generally see mIoUs increase. Segmentation results with different SSL cues are displayed in Fig. 4.3a.

To find the limits of our method, we perform further ablations on the ground-level Big Bear Lake sequences and find lower overall performance compared to aerial scenes (Table 4.4). We attribute this to three things: First, a ground-level viewing angle from land causes water bodies to appear smaller, making the effect of noisy labels more pronounced. Second, water reflections tend to be more intense at shallower angles which texturizes water even when still. Lastly, dense fog and thermal sensor noise in some of sequences obscure the scene, making sky, background land, and water appear very uniform. Despite this, texture-based adaptation (PT+TC) still outperforms motion-based (PT+MC) by 0.11 mIoU with sky segmentation and 0.18 mIoU with IMU-based horizon estimation, reaffirming its use in calm water settings.

Horizon estimation and sky segmentation: When IMU is available, horizon estimation can be used to boost segmentation performance (Table 4.4). Moreover, as it does not rely on vision, it can mitigate the impact of fog and cloud obfuscations, as evident in the Big Bear Lake sequences where using the horizon yields over 0.35 gain in texture-based (PT+TC) mIoU versus having no knowledge of the sky or horizon. Sky segmentation via Fast-SCNN is less robust but still works well in the river settings and Castaic Lake 4. However, it is prone to mistaking far-field water as sky in Castaic Lake 2 and coastal scenes at Duck, leading to mIoU drop. It shows marked improvement over no sky segmentation in the Big Bear Lake scenes, demonstrating some robustness to fog, but still leaves room for improvement.

Table 4.5: Pretraining method ablation using all thermal sequences.

Pretraining Method	River	Lake	Coast	Ground	Avg. mIoU
Grayscale	0.365	0.443	0.482	0.422	0.428
Rand. Mixing	0.419	0.390	0.447	0.503	0.440
Rand. Mixing (PCA)	**0.563**	**0.441**	**0.642**	**0.574**	**0.555**
RGB2Thermal	0.291	0.276	0.224	0.370	0.290
MassMIND [29]	0.454	0.310	0.445	0.488	0.424

Network pretraining ablation: We evaluate the RGB-pretrained networks (Sec. 4.4) on our thermal data in absence of online SSL (Table 4.5). PCA channel mixing outperforms others, possibly because it can modulate the amount of image detail shown However, we leave a thorough investigation and explanation of this observation for future work. RGB-T image translation does not perform well likely because it had limited access to target domain data and introduced numerous structural artifacts that affected segmentation training.

UAV Embedded System Benchmarks

To demonstrate deploying our algorithm in real-time on UAV hardware, we implement our algorithm in ROS and test with bagfiles on an Nvidia Jetson AGX Orin.

ROS architecture: A node (Fig. 4.5) processes incoming thermal imagery (Sec. 4.4) and estimates horizon line based on IMU readings or segments the sky using Fast-SCNN if IMU is unavailable. Texture- and/or motion-based labeling nodes generate segmentation labels in parallel. Thermal images and labels are cached in a buffer and training begins once the buffer is full. A third inference network segments incoming images continuously and receives weights from the training network after each online SSL cycle.

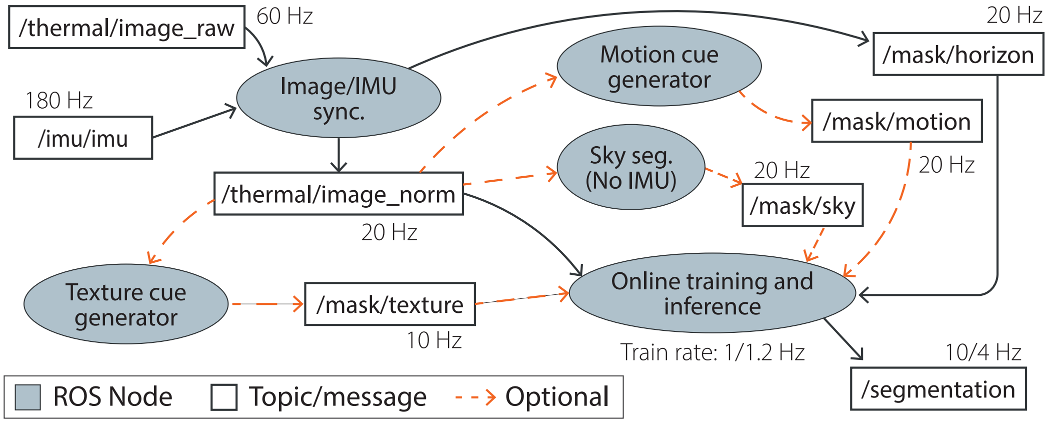


Figure 4.5: ROS architecture for real-time online learning and water segmentation on a Nvidia Jetson AGX Orin.

Computation benchmarks: We benchmarked our system using training parameters from Sec. 4.5 and list component frequencies in Fig. 4.5. Texture- and motion-based adaptation perform online updates at 1 and 1.2 Hz respectively. Actual training takes 0.5 s for 8 iterations with the rest of the time spent filling the training buffer. The inference network produces segmentations at 10 and 4 Hz when using texture and motion cues respectively. Code optimization, better parallelization strategies, and lower online SSL update rates should allow us to attain closer to 15-20 Hz. Overall, we find these metrics to be suitable to enable our future work in nighttime navigation and planning in near-shore areas, as well as other work in bathymetry.

4.6 Conclusion

We presented a CNN-based thermal water segmentation algorithm that provides UAVs operating in near-shore environments with nighttime capabilities. We demonstrated that our online SSL approach with simple water cues can achieve strong and consistent results in the aerial setting despite the lack of aerial thermal data. Furthermore, we showed that our method is superior and more robust compared to fully-supervised networks trained on existing thermal data. This work can enable thermal vision-based UAV science missions in near-shore settings for tasks such as bathymetry and coastline mapping. In the future, we look to use this to assist UAV navigation and planning in near-shore environments, and to help curate a larger, aerial thermal near-shore dataset to enable fully-supervised training.

Acknowledgements

We thank Kara Koetje, Katherine Brodie, Tyler Hesser, Thomas Almeida, Thomas Cook, and Reginald Beach for their helpful discussions and thank Nicholas Spore with the U.S. Army ERDC-CHL Field Research Facility for collecting the data sequences at Duck, NC.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. "SLIC superpixels compared to state-of-the-art superpixel methods". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.11 (2012), pp. 2274–2282.

[2] S. Achar, B. Sankaran, S. Nuske, S. Scherer, and S. Singh. "Self-supervised segmentation of river scenes". In: *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE. 2011, pp. 6227–6232.

[3] I. B. Akkaya, F. Altinel, and U. Halici. "Self-training guided adversarial domain adaptation for thermal imagery". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2021, pp. 4322–4331.

[4] N. Araslanov and S. Roth. "Self-supervised augmentation consistency for adapting semantic segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2021, pp. 15384–15394.

[5] B. Bovcon and M. Kristan. "A water-obstacle separation and refinement network for unmanned surface vehicles". In: *Proceedings of the IEEE International Conference on Robotics and Automation*. 2020, pp. 9470–9476. DOI: 10.1109/ICRA40945.2020.9197194.

[6] B. Bovcon, J. Muhovič, J. Perš, and M. Kristan. "The MaSTr1325 dataset for training deep USV obstacle detection models". In: *Proceedings of the*

IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE. 2019.

[7] B. Bovcon, J. Perš, M. Kristan, et al. “Stereo obstacle detection for unmanned surface vehicles by IMU-assisted semantic segmentation”. In: *Robotics and Autonomous Systems* 104 (2018), pp. 1–13.

[8] K. L. Brodie, B. L. Bruder, R. K. Slocum, and N. J. Spore. “Simultaneous Mapping of Coastal Topography and Bathymetry From a Lightweight Multicamera UAS”. In: *IEEE Transactions on Geoscience and Remote Sensing* 57.9 (2019), pp. 6844–6864. DOI: 10.1109/TGRS.2019.2909026.

[9] H. Caesar, J. Uijlings, and V. Ferrari. “Coco-stuff: Thing and stuff classes in context”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1209–1218.

[10] J. Chen, Z. Liu, D. Jin, Y. Wang, F. Yang, and X. Bai. “Light Transport Induced Domain Adaptation for Semantic Segmentation in Thermal Infrared Urban Scenes”. In: *IEEE Transactions on Intelligent Transportation Systems* 23.12 (2022), pp. 23194–23211.

[11] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. “A simple framework for contrastive learning of visual representations”. In: *International conference on machine learning*. PMLR. 2020, pp. 1597–1607.

[12] X. Chen and K. He. “Exploring simple siamese representation learning”. In: *C-CVPR*. 2021, pp. 15750–15758.

[13] J. Cho, Y. Kim, H. Jung, C. Oh, J. Youn, and K. Sohn. “Multi-Task Self-Supervised Visual Representation Learning for Monocular Road Segmentation”. In: *2018 IEEE International Conference on Multimedia and Expo (ICME)*. 2018, pp. 1–6. DOI: 10.1109/ICME.2018.8486472.

[14] S. Daftry, Y. Agrawal, and L. Matthies. “Online Self-Supervised Long-Range Scene Segmentation for MAVs”. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2018, pp. 5194–5199.

[15] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. R. Bradski. “Self-supervised monocular road detection in desert terrain.” In: *Proceedings of the Robotics: Science and Systems Conference*. Vol. 38. Philadelphia. 2006.

[16] G. Farnebäck. “Two-frame motion estimation based on polynomial expansion”. In: *Image Analysis: 13th Scandinavian Conf., SCIA 2003 Halmstad, Sweden, June 29–July 2, 2003 Proceedings 13*. Springer. 2003, pp. 363–370.

[17] M. A. Fischler and R. C. Bolles. “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”. In: *Commun. ACM* 24.6 (1981), pp. 381–395. ISSN: 0001-0782. DOI: 10.1145/358669.358692.

[18] L. Gan, C. Lee, and S.-J. Chung. “Unsupervised rgb-to-thermal domain adaptation via multi-domain attention network”. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE. 2023, pp. 6014–6020.

[19] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, et al. “Searching for mobilenetv3”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 1314–1324.

[20] S. Hwang, J. Park, N. Kim, Y. Choi, and I. S. Kweon. “Multispectral Pedestrian Detection: Benchmark Dataset and Baselines”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015.

[21] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. “Image-to-Image Translation with Conditional Adversarial Networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017.

[22] Y.-H. Kim, U. Shin, J. Park, and I. S. Kweon. “MS-UDA: Multi-Spectral Unsupervised Domain Adaptation for Thermal Image Semantic Segmentation”. In: *IEEE Robotics and Automation Letters* 6.4 (2021), pp. 6497–6504. DOI: 10.1109/LRA.2021.3093652.

[23] M. Kristan, V. S. Kenk, S. Kovačič, and J. Perš. “Fast image-based obstacle detection from unmanned surface vehicles”. In: *IEEE Trans. on Cybernetics* 46.3 (2015), pp. 641–654.

[24] C. Li, W. Xia, Y. Yan, B. Luo, and J. Tang. “Segmenting objects in day and night: Edge-conditioned cnn for thermal image semantic segmentation”. In: *IEEE Transactions on Neural Networks and Learning Systems* 32.7 (2020), pp. 3069–3082.

[25] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. “Feature pyramid networks for object detection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 2117–2125.

[26] L. Lopez-Fuentes, C. Rossi, and H. Skinnemoen. “River segmentation for flood monitoring”. In: *2017 IEEE Int. Conf. on Big Data*. 2017, pp. 3746–3749. DOI: 10.1109/BigData.2017.8258373.

[27] D. G. Lowe. “Object recognition from local scale-invariant features”. In: *Proceedings of the IEEE International Conference on Computer Vision*. Vol. 2. 1999, pp. 1150–1157.

[28] K. Meier, S.-J. Chung, and S. Hutchinson. “River segmentation for autonomous surface vehicle localization and river boundary mapping”. In: *Journal of Field Robotics* 38.2 (2021), pp. 192–211.

[29] S. Nirgudkar, M. DeFilippo, M. Sacarny, M. Benjamin, and P. Robinette. “MassMIND: Massachusetts Maritime INfrared Dataset”. In: *International Journal of Robotics Research* 42.1-2 (2023), pp. 21–32.

[30] S. Nirgudkar and P. Robinette. "Beyond visible light: Usage of long wave infrared for object detection in maritime environment". In: *Int. Conf. on Advanced Robotics (ICAR)*. IEEE. 2021, pp. 1093–1100.

[31] T. Park, A. A. Efros, R. Zhang, and J.-Y. Zhu. "Contrastive Learning for Unpaired Image-to-Image Translation". In: *Proceedings of the European Conference on Computer Vision*. 2020.

[32] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, et al. "Adaptive histogram equalization and its variations". In: *Computer vision, graphics, and image processing* 39.3 (1987), pp. 355–368.

[33] R. P. Poudel, S. Liwicki, and R. Cipolla. "Fast-scnn: Fast semantic segmentation network". In: *arXiv preprint arXiv:1902.04502* (2019).

[34] X. Qian, M. Zhang, and F. Zhang. "Sparse gans for thermal infrared image generation from optical image". In: *IEEE Access* 8 (2020), pp. 180124–180132.

[35] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, et al. "ROS: an open-source Robot Operating System". In: *ICRA Workshop on Open Source Software*. Kobe, Japan, 2009.

[36] A. Rankin and L. Matthies. "Daytime water detection based on color variation". In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2010, pp. 215–221. DOI: 10.1109/IROS.2010.5650402.

[37] A. L. Rankin, L. H. Matthies, and A. Huertas. "Daytime water detection by fusing multiple cues for autonomous off-road navigation". In: *Transformational Science And Technology For The Current And Future Force*. World Scientific, 2006, pp. 177–184.

[38] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. "ORB: An efficient alternative to SIFT or SURF". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2011, pp. 2564–2571. DOI: 10.1109/ICCV.2011.6126544.

[39] R. Schmid, D. Atha, F. Schöller, S. Dey, S. Fakoorian, K. Otsu, B. Ridge, M. Bjelonic, L. Wellhausen, M. Hutter, et al. "Self-supervised traversability prediction by learning to reconstruct safe terrain". In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2022, pp. 12419–12425.

[40] J. Yang, A. Dani, S.-J. Chung, and S. Hutchinson. "Vision-based localization and robot-centric mapping in riverine environments". In: *J. of Field Robotics* 34.3 (2017), pp. 429–450.

[41] L. Yao, D. Kanoulas, Z. Ji, and Y. Liu. “ShorelineNet: an efficient deep learning approach for shoreline semantic segmentation for unmanned surface vehicles”. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2021, pp. 5403–5409.

[42] W. Zhan, C. Xiao, Y. Wen, C. Zhou, H. Yuan, S. Xiu, Y. Zhang, X. Zou, X. Liu, and Q. Li. “Autonomous visual perception for unmanned surface vehicle navigation in an unknown environment”. In: *Sensors* 19.10 (2019), p. 2216.

[43] H. Zhang, E. Fromont, S. Lefèvre, and B. Avignon. “Multispectral Fusion for Object Detection with Cyclic Fuse-and-Refine Blocks”. In: *Proceedings of the International Conference on Image Processing*. Abu Dhabi, United Arab Emirates, 2020, pp. 1–5.

[44] L. Zhang, A. Gonzalez-Garcia, J. Van De Weijer, M. Danelljan, and F. S. Khan. “Synthetic data generation for end-to-end thermal infrared tracking”. In: *IEEE Transactions on Image Processing* 28.4 (2018), pp. 1837–1850.

[45] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba. “Semantic understanding of scenes through the ade20k dataset”. In: *International Journal of Computer Vision* (2018).

[46] S. Zhou and K. Iagnemma. “Self-supervised learning method for unstructured road detection using Fuzzy Support Vector Machines”. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2010, pp. 1183–1189. DOI: 10.1109/IROS.2010.5650300.

Chapter 5

UNSUPERVISED RGB-TO-THERMAL DOMAIN ADAPTATION VIA MULTI-DOMAIN ATTENTION NETWORK

[1] L. Gan, C. Lee, and S.-J. Chung. “Unsupervised RGB-to-Thermal Domain Adaptation via Multi-Domain Attention Network”. In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. 2023, pp. 6014–6020. DOI: 10.1109/ICRA48891.2023.10160872.

5.1 Abstract

This work presents a new method for unsupervised thermal image classification and semantic segmentation by transferring knowledge from the RGB domain using a multi-domain attention network. Our method does not require any thermal annotations or co-registered RGB-thermal pairs, enabling robots to perform visual tasks at night and in adverse weather conditions without incurring additional costs of data labeling and registration. Current unsupervised domain adaptation methods look to align global images or features across domains. However, when the domain shift is significantly larger for cross-modal data, not all features can be transferred. We solve this problem by using a shared backbone network that promotes generalization, and domain-specific attention that reduces negative transfer by attending to domain-invariant and easily-transferable features. Our approach outperforms the state-of-the-art RGB-to-thermal adaptation method in classification benchmarks, and is successfully applied to thermal river scene segmentation using only synthetic RGB images. Our code is made publicly available at https://github.com/ganlumomo/thermal-uda-attention.

5.2 Introduction

Cameras are critical for robot perception as they provide dense measurements and rich environmental information. However, most existing vision models are developed for cameras operating in the visible spectrum due to their ubiquity and the accessibility of large-scale RGB datasets [5, 18]. Although these models allow robotic systems such as autonomous vehicles (AV) to work well in ideal conditions with sufficient illumination, their performance is largely degraded at night and in adverse conditions. Thermal cameras, on the other hand, detect electromagnetic

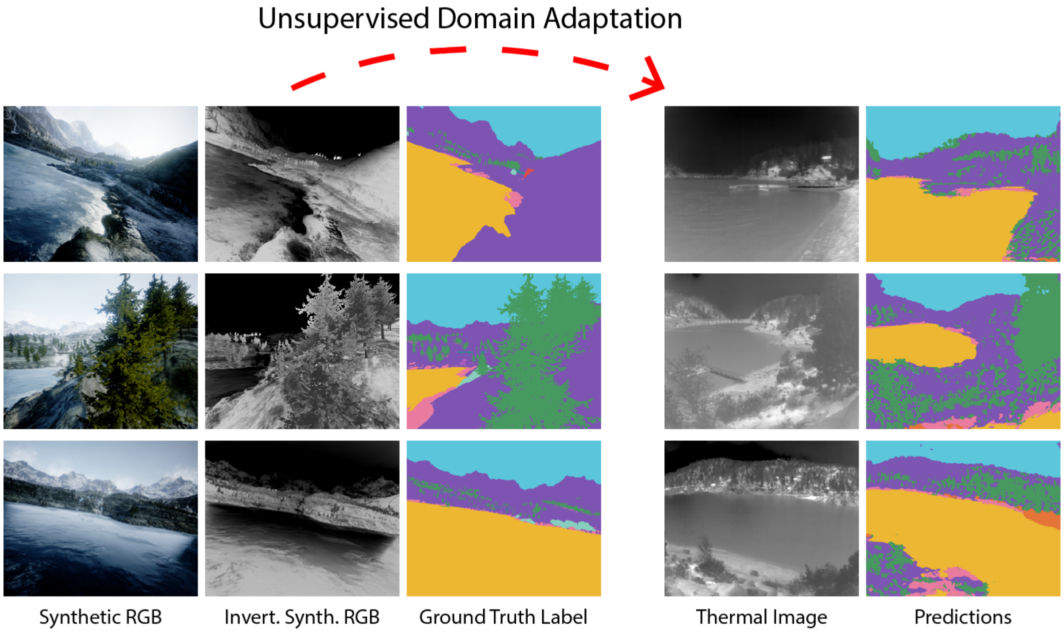


Figure 5.1: Our RGB-to-thermal unsupervised domain adaptation (UDA) leverages knowledge learned from a synthetic annotated RGB dataset to perform semantic segmentation on thermal river scenes without requiring thermal annotations.

waves beyond the visible spectrum that penetrate through dust, smoke, and light fog, enabling around-the-clock robotic operations.

One popular approach towards robust vision is to leverage thermal images in conjunction with RGB via multi-spectral sensor fusion. These methods have largely benefited from recent interests in AV technology, resulting in curated datasets [6, 19] being made publicly available. Notable examples are GAFF [43] and CFT [25], two multi-spectral object detection networks trained on paired RGB-thermal image datasets for feature extraction and fusion. In particular, the fusion network in [25] sees a 25% performance improvement over a single RGB branch on the FLIR-aligned dataset [6]. Urban semantic segmentation has also been improved for nighttime and adverse weather after integrating thermal capabilities [10, 45, 14, 36]. However, these models are fully-supervised, using annotated images or co-registered RGB-thermal pairs which are expensive to acquire and small in scale [15]. In non-AV applications, the lack of thermal data and cost of labeling hinder the development of thermal vision models, especially when current vision models, like Transfomers, have been trending larger [11].

To overcome this issue, we look to leverage existing large-scale RGB datasets to learn thermal models via unsupervised domain adaptation (UDA) techniques. UDA aims to transfer the knowledge learned in a labeled source domain to an unlabeled target domain [38]. Although most UDA methods focus on domains from different environments but within the same modality (mainly RGB images), such as GTAV-to-Cityscapes [32, 12], the underlying assumption that a domain-invariant feature representation exists also applies to cross-modal data, especially for semantic-related tasks.

In this work, we aim to transfer knowledge learned from labeled RGB images to unlabeled thermal images. This is challenging for two reasons: First, cross-modal domains have larger domain shifts and more dissimilar features compared to domains within same modalities. UDA methods that match global images or feature distributions of both domains can hurt generalization and lead to *negative transfer* in which untransferable features are forcefully aligned [44, 37], [42]. Second, UDA methods based on generative adversarial networks (GANs) need a large amount of unlabeled target data to be well-trained [38] which can also be unavailable in the thermal domain.

We surmount these challenges by designing a multi-domain attention network with a shared backbone and domain-specific attention for RGB-to-thermal adaptation. This shared backbone promotes generalization across domains, prevents feature over-alignment, and relaxes the thermal dataset size requirement. For feature alignment, we train the target-specific attention using adversarial learning to attend to and transfer more domain-invariant and transferable features among all shared features to alleviate negative transfer. The main contributions of our work are as follows:

1. We establish an unsupervised RGB-to-thermal domain adaptation method using a multi-domain attention network and adversarial attention learning.
2. We evaluate our method on thermal image classification tasks and outperform the state-of-the-art RGB-to-thermal adaptation approach on two benchmarks.
3. We demonstrate the versatility of our approach, leveraging it to perform thermal river scene segmentation, and to the best of our knowledge, are the first to utilize synthetic RGB data for thermal semantic segmentation.

5.3 Related Work

Unsupervised Domain Adaptation: UDA has been successfully applied to a variety of vision tasks including image classification [7, 34, 29, 20, 1], semantic segmentation [12, 32, 9], and 2D/3D object detection [40, 22]. Domain alignment is the fundamental principle of UDA, and can be achieved by two main methodologies: domain mapping and domain-invariant feature learning [38]. Domain mapping can be viewed as pixel-level alignment which maps images from one domain to another via image translation. For instance, PixelDA [2] and CyCADA [12] map source training data into the target domain using conditional GANs and train the downstream model on the fake target data. Pixel-level alignment can remove the domain differences in the input space to some extent but such differences are primarily low-level [38]. Other works achieve domain adaptation by domain-invariant feature learning or feature-level alignment. By mapping source and target input data to the same feature distribution, a downstream predictor trained on such domain-invariant features from source can also work well on the target domain. This is typically done by minimizing a distance defined on distributions [29], or by adversarial training via a domain discriminator that attempts to distinguish between source and target features [7, 34, 20, 32, 1]. Our method is similar to these works and can be viewed as an instance of the general pipeline in [34] by leveraging multi-domain network and attention mechanisms.

RGB-to-Thermal UDA: Despite the success of UDA on visible images, adapting models from visible to thermal remains challenging due to their larger domain gap. Existing RGB-to-thermal adaptation works like MS-UDA [14] and HeatNet [36] distill knowledge from a semantic segmentation network pretrained on RGB datasets to their two-stream network by pseudo-labeling RGB-thermal image pairs. However, as the pseudo-labels are generated for the RGB image in a pair, the main domain gap here is intra-modal, between the pretraining dataset and RGB images in the paired dataset, rather than inter-modal.

Our work is mostly related to SGADA [1] and Marnissi *et al.* [22] which aim to transfer knowledge from RGB to thermal without requiring thermal annotations or RGB-thermal pairs. For pedestrian detection, Marnissi *et al.* [22] incorporates alignment at difficult levels into Faster R-CNN [28] using adversarial training. SGADA [1] is built upon ADDA [34] with an additional self-training procedure. For pseudo-labeling, not only the model prediction and confidence are considered, but also the prediction and confidence from the domain discriminator. It achieves the best results on MS-COCO [18] to FLIR ADAS [6] adaptation benchmark, however, its performance largely depends on the quality of pseudo labels generated by ADDA.

Attention Networks: Attention mechanisms allow models to dynamically attend to certain parts of the input that are more effective for a task, and become important

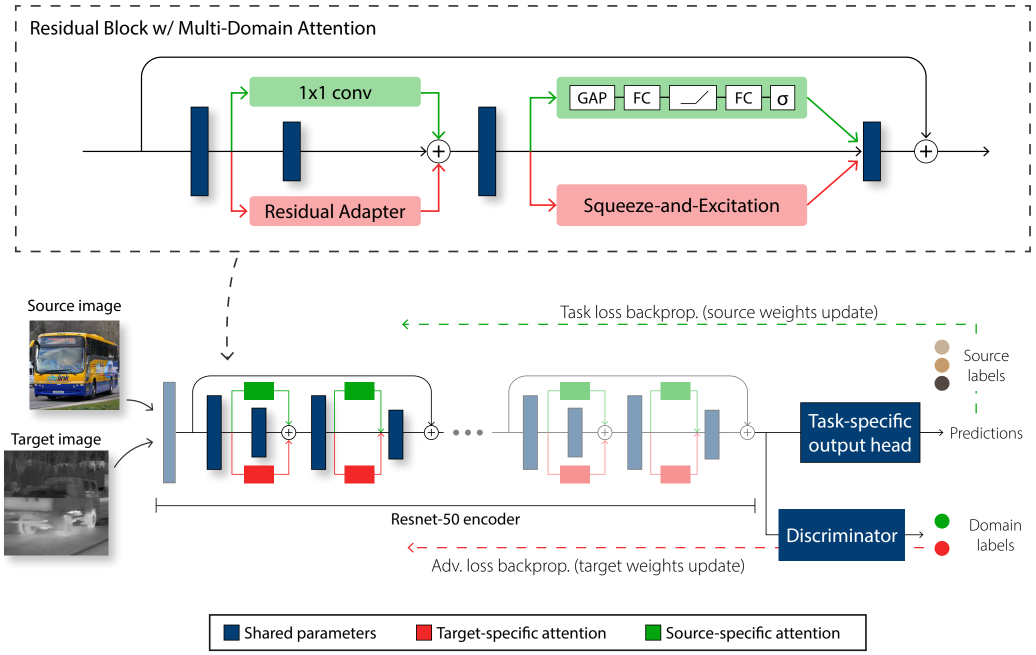

Residual Block w/ Multi-Domain Attention
1x1 conv
Residual Adapter
GAP
FC
FC
σ
Squeeze-and-Excitation
Source image
Target image
Task loss backprop. (source weights update)
Source labels
Task-specific output head
Predictions
Resnet-50 encoder
Discriminator
Domain labels
Adv. loss backprop. (target weights update)
Shared parameters
Target-specific attention
Source-specific attention

Figure 5.2: The network architecture and training procedure of our proposed unsupervised RGB-to-thermal domain adaptation method. The specific architecture is shown for image classification task.

concepts in neural networks. Attention can be grouped into different types, including sequence attention, channel attention [13], and spatial attention [39], etc. For domain adaptation, Wang *et al.* [37] and Zhang *et al.* [42] propose transferable attention networks using self-attention mechanisms to highlight transferable features. The spatial attention they employed attend to different regions in a feature map. Instead, we use channel-wise attention [13] to attend to different feature maps and use residual adapters [27] to align them, with the intuition that certain types of features are more transferable than others. The transferability difference in feature types (i.e., channels) should be focused on more than in feature regions (i.e., spatial locations) for cross-modal domains.

5.4 Method

Multi-Domain Attention Network

Our multi-domain attention network design draws ideas from multi-domain learning [27] and task attention mechanisms in multi-task learning [21]. Both works use a shared backbone network and domain/task-specific parameters to separate a shared representation learned from all domain/tasks and domain/task-specific modeling capabilities. It has been shown that sharing weights across domains/tasks promotes the generalization ability. In contrast with encouraging disentanglement in a supervised setup [27, 21], we use domain-specific attention with adversarial learning to facilitate domain-invariant feature extraction and alignment for domain adaptation.

Our multi-domain attention network consists of an encoder-decoder backbone, shared by both source and target domains, with domain-specific attention modules attached at various stages of the encoder. For UDA classification (Fig. 5.2), the architecture consists of the shared backbone and classifier (blue), source-specific (green), and target-specific (red) attention modules. Hypothesizing that different sensor modality favors different types of features, we use channel-wise attention, i.e., Squeeze-and-Excitation (SE) [13], to highlight more domain-invariant and easily-transferable feature maps among all shared features, and use residual adapters [27] to align them across domains.

Let $F_c \in \mathbb{R}^{h \times w \times C'}$ denote a convolutional layer of C kernels of size $h \times w$ operating on C' input channels, we have $F_c : x \longrightarrow f$, $x \in \mathbb{R}^{H' \times W' \times C'}$, $f \in \mathbb{R}^{H \times W \times C}$, where $f = [f_1, f_2, ..., f_C]$ represent C output feature maps. A SE module [13] first "squeezes" f into a low-dimensional channel descriptor $d \in \mathbb{R}^{\frac{C}{r}}$ with reduction ratio r. This is done using global average pooling followed by a fully connected (FC) layer with ReLU activations. The channel descriptor is then transformed into channel-wise weight coefficients $s = [s_1, s_2, ...s_C]$, $s_c \in (0, 1)$ through another FC layer and a sigmoid function. Finally, s is used to "excite" different feature maps in f by feature channel reweighting: $\widetilde{f_c} = s_c \cdot f_c$. In our network, we use domain-specific SE blocks operating on the shared feature maps right before the residual addition in residual blocks, as shown in Fig. 5.2.

By attaching domain-specific SE modules to the shared backbone network, they have the ability to accentuate more domain-invariant and transferable features in the shared features while attenuate less-transferable ones. To further align the reweighted features across domains, we leverage residual adapters [27] to directly and dynamically adapting the shared feature extractors F_c to domain-specific feature extractors G_d. Specifically, $d = c$ in our network.

Given a shared convolutional layer of C kernels $F_c \in \mathbb{R}^{h \times w \times C'}$, a domain-specific convolutional layer with D filters $G_d \in \mathbb{R}^{h \times w \times C'}$ can be simply constructed as an affine transformation of F_c using only a small amount of additional parameters

$\alpha = \{\alpha_{dc}\}$:

$$G_d = \sum_{c=1}^{C} \alpha_{dc} F_c. \tag{5.1}$$

Here, $\alpha \in \mathbb{R}^{D \times C}$ are the trainable residual adapter parameters [27]. This linear parameterization reduces constructing G_d for each domain to the shared F_c with a small amount of domain-specific parameters α. The works [27, 26] further show that α can be reparameterized and implemented as a convolutional layer of 1×1 filters connected in parallel with the shared convolutional layer. In our network, we add residual adapters to the middle 3×3 convolutional layer in residual blocks for feature alignment, as shown in Fig. 5.2.

We emphasize the differences of our attention modules from those in [27, 21]. The multi-domain learning in [27] and multi-task learning in [21] are essentially supervised. Their objective is to learn a domain/task-invariant feature representation f_{inv} and domain/task-specific attention θ_a, θ_b, so that $\theta_a(f_{inv}) = f_a$, $\theta_b(f_{inv}) = f_b$, where f_a and f_b are features tailored for domain/task A and B respectively. In contrast, for our UDA problem, we learn discriminative features f_s for a given task using supervised training in the source domain and target-specific attention θ_t using adversarial training for feature alignment, i.e. $\theta_s(f_{sh}) = f_s$, $\theta_t(f_{sh}) = f_{t \to s}$, where θ_s and θ_t are source and target attention, f_{sh} and $f_{t \to s}$ are the shared features and the target features aligned with f_s respectively.

Adversarial Attention Learning

To perform unsupervised domain adaptation, we train subsets of network parameters in an alternating fashion. We denote the shared parameters, including the backbone network and the decoder, as θ_{sh}, and the source- and target-specific attention modules as θ_s and θ_t respectively. We train θ_{sh} and θ_s using labeled data from the source domain and train the task-specific attention modules θ_t adversarially in an alternating fashion.

Let M denote the proposed multi-domain attention network, and let $\mathcal{D}_s = \{(x_s^i, y_s^i)\}_{i=1}^{n_s}$ and $\mathcal{D}_t = \{(x_t^j)_{j=1}^{n_t}\}$ denote the annotated training data in the source domain and the unlabeled target training data respectively. The shared and source-specific network parameters can be trained with supervision by minimizing the standard cross-entropy loss. For a classification problem, the loss can be written as

$$\mathcal{L}_{task}(x_s, y_s) = -\sum_{c=1}^{C} \mathbb{1}_{[c=y_s]} \log M(x_s; \theta_{sh+s}), \tag{5.2}$$

Algorithm 2 Multi-Domain Attention Network for Unsupervised Domain Adaptation

```
Input: Training data: $\mathcal{D}_s, \mathcal{D}_t$,
       Network: $M = \{\theta_{sh}, \theta_s, \theta_t\}$, Discriminator: $D$
       Learning rate: $\alpha, \beta, \gamma$
Initialize $M^0, D^0$
for $n = 1$ to $N$ do
    Sample batch data $(x_s, y_s)$ from $\mathcal{D}_s$, and $x_t$ from $\mathcal{D}_t$
    $l_{task} \leftarrow \mathcal{L}_{task}(x_s, y_s)$                ▷ Evaluate (5.2)
    $\theta^n_{sh+s} = \theta^{n-1}_{sh+s} - \alpha \nabla_{\theta^{n-1}_{sh+s}} l_{task}$
    $l_{adv} \leftarrow \mathcal{L}_{adv}(x_t, D^{n-1})$               ▷ Evaluate (5.5)
    $\theta^n_t = \theta^{n-1}_t - \beta \nabla_{\theta^{n-1}_t} l_{adv}$
    $l_{dis} \leftarrow \mathcal{L}_{dis}(x_s, x_t, M^n)$              ▷ Evaluate (5.4)
    $D^n = D^{n-1} - \gamma \nabla_{D^{n-1}} l_{dis}$
end for
Output: $M^N, D^N$
```

where (x_s, y_s) are source data-label pairs drawn from $\mathcal{D}_s$, $\mathbb{1}_{[x]}$ is an indicator function so that $\mathbb{1}_{[x]} = 1$ if $x = 1$, and 0 otherwise, and C is the number of categories.

We train the target-specific attention in our network adversarially by forcing the target attention to attend to domain-invariant features from the shared features and further align them with the source feature distributions. Adversarial attention learning can be achieved by approaching the following minimax game [8, 34] between the target-specific attention θ_t and a domain discriminator D:

$$\begin{aligned} &\min_{\theta_t} \max_{D} \mathcal{L}(D, \theta_t) = \\ &\qquad \mathbb{E}_{x_s \sim \mathcal{D}_s} \log D(f_s) + \mathbb{E}_{x_t \sim \mathcal{D}_t} \log(1 - D(f_t)), \end{aligned} \tag{5.3}$$

where f_s and f_t are source and target features from the entire encoder with weights θ_{sh+s} and θ_{sh+t}, respectively.

Specifically, the minimax loss in (5.3) is split into two objectives, where the domain discriminator plays the adversarial role and attempts to distinguish between source features f_s and target features f_t by minimizing the following loss:

$$\mathcal{L}_{dis}(x_s, x_t, M) = -\log D(f_s) - \log\left(1 - D(f_t)\right), \tag{5.4}$$

and the target-specific attention is trained to fool the domain discriminator and increase domain confusion by minimizing an adversarial loss:

$$\mathcal{L}_{adv}(x_t, D) = -\log D(f_t). \tag{5.5}$$

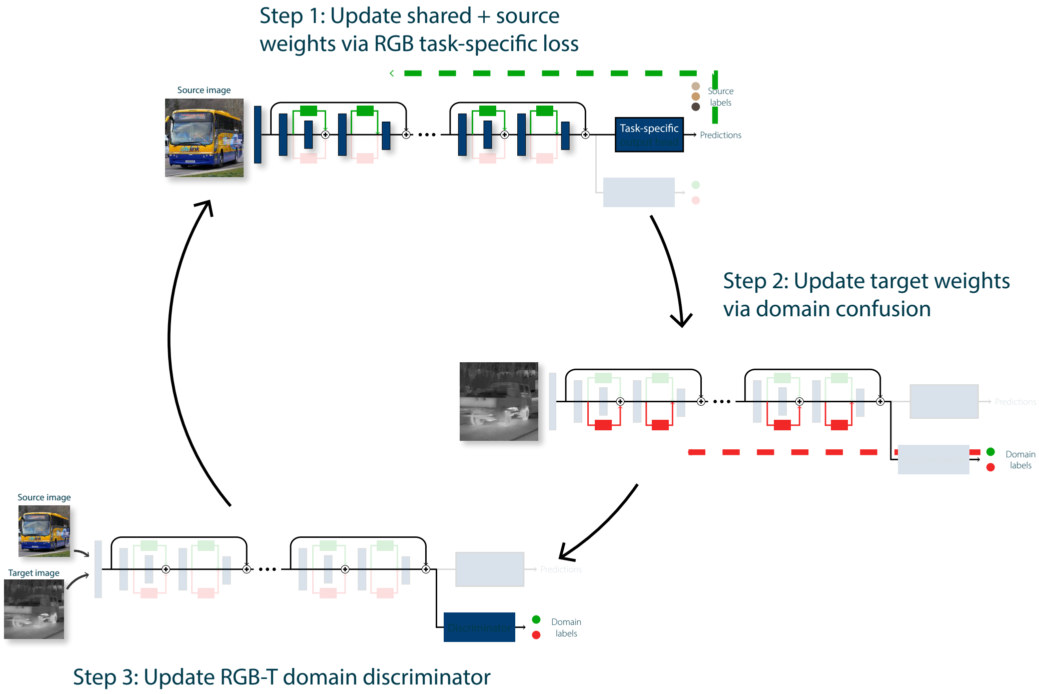

Step 1: Update shared + source weights via RGB task-specific loss
Source image
Source labels
Task-specific
Predictions
Step 2: Update target weights via domain confusion
Predictions
Domain labels
Source image
Target image
Predictions
Domain labels
Step 3: Update RGB-T domain discriminator

Figure 5.3: Visualized training loop to perform unsupervised domain adaptation through acheiving domain confusion.

The three-step training procedure of our multi-domain attention network is given in Algorithm 2 and depicted in Fig. 5.3.

Advantages of this alternating training are twofold. First, when training θ_{sh+s} using $\mathcal{L}_{task}(x_s, y_s)$, it reduces to training a supervised source model and the feature extractor learns to extract the most discriminative features for the given task. When training θ_t with fixed θ_{sh}, θ_t learns to select and adapt the most domain-invariant ones among the discriminative features, leading to better adaptation performance. Second, it eliminates a weighting hyperparameter for two loss functions and makes the training procedure more stable.

Self-Training

We further fine-tune the model with a single self-training step using the pseudo labels generated for the target training data. Following [1], we save the prediction and corresponding confidence (the maximum of softmax probabilities) of the model M trained in Sec. 5.4 for all unlabeled target training samples. In the meantime, the prediction and confidence of the domain discriminator D are also recorded. For target samples that successfully fool the discriminator (D predicts them as source samples with a high confidence), we assign them pseudo-labels according to the

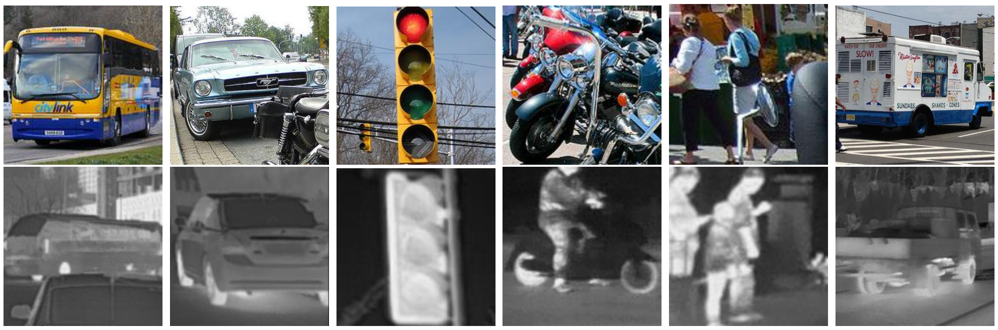


Figure 5.4: Examples of the prepared data from MS-COCO [18] and M^{3}FD Detection datasets [19].

model prediction. For those target samples that the discriminator recognize but with low domain confidence, we also include them in pseudo-labeling. The pseudo-labels are further filtered by the model prediction confidence.

In this stage, we only train the target-specific attention parameters using a cross-entropy loss in a supervised setup:

$$\mathcal{L}_{st}(x_t, \hat{y}_t) = -\sum_{c=1}^{C} \mathbb{1}_{[c=\hat{y}_t]} \log M(x_t; \theta_t), \tag{5.6}$$

where $\hat{y}_t$ is the generated pseudo-label for target training data x_t. This way, we can further improve the performance in target while keeping the performance in source, so that we have a single unified model that performs well on both source and target data. This learning-without-forgetting [17] property is another benefit of our multi-domain attention network.

5.5 Results

Implementation

For a fair comparison, we employ the same backbone architecture used in other methods, i.e. a ResNet-50 pretrained on ImageNet [5]. We use a FC classifier and a FC discriminator for image classification, and use an Atrous Spatial Pyramid Pooling [4] decoder and a fully-convolutional discriminator for semantic segmentation. Parameters are all updated using the ADAM optimizer with $\beta_1 = 0.5, \beta_2 = 0.999$ and weight decay of 2.5×10^{-5}. The learning rate α, β, γ in Algorithm 2 is set to $1 \times 10^{-4}, 1 \times 10^{-5}$ and 1×10^{-3}, respectively. All experiments are conducted on a NVIDIA Quadro RTX 8000 GPU with 48GB memory.

Table 5.1: Data statistics of MS-COCO versus M^3FD dataset.

		Bus	Car	Light	Motor.	People	Truck	Total
MS-COCO	Train	3,887	36,830	12,139	6,330	200,831	7,232	267,249
	Val	189	1,623	603	229	8,331	315	11,290
M^3FD	Train	441	12,969	1,902	382	8,770	696	25,160
	Val	55	1,621	238	48	1,096	87	3,141
	Test	55	1,620	237	47	1,096	86	3,141

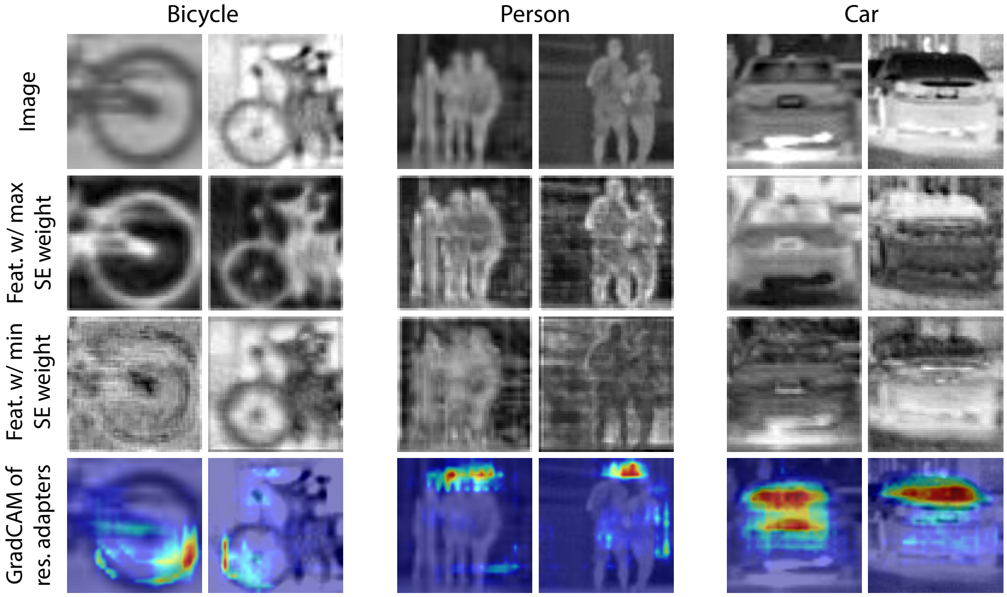


Figure 5.5: Task-specific attention visualization for classes in the FLIR dataset and Grad-CAM [30] visualizations of the residual adapters for that class.

Ablation Study

To investigate the effects of different attention modules and training strategies on adaptation performance, we conduct a thorough ablation study on 9 training combinations resulting from two types of attention modules, i.e. with (✓) and without the residual adapter/SE module, and three different strategies to train them:

1. θ_{sh+s+t}: We jointly train all network parameters using the sum of $\mathcal{L}_{task}$ in (5.2) and $\mathcal{L}_{adv}$ in (5.5), reducing the three training steps in Algorithm 2 to only alternatively training the model M and domain discriminator D.

2. $\theta_{sh}, \theta_{s+t}$: We alternatively train the shared parameters θ_{sh} and all domain-specific parameters θ_{s+t}. In this setting, only θ_{sh} is updated using $\mathcal{L}_{task}$, while θ_{s+t} are adversarially trained using a cross-entropy domain loss in [33] instead of $\mathcal{L}_{adv}$ in Algorithm 2.

Table 5.2: Ablation study of different attention modules and training strategies on MS-COCO to FLIR ADAS dataset.

Training Strategy	Residual Adapter	Squeeze & Excitation	Bicycle	Car	Person	Average
θ_{sh+s+t}	✓		89.43	**97.14**	88.89	**91.83**
		✓	**91.72**	93.79	83.96	89.83
	✓	✓	87.82	94.34	**91.52**	91.23
θ_{sh}, θ_{s+t}	✓		81.84	96.36	96.66	**91.62**
		✓	**82.03**	93.18	95.84	90.35
	✓	✓	79.54	**96.69**	**96.99**	91.07
θ_{sh+s}, θ_t	✓		**90.57**	97.22	89.83	92.54
		✓	85.75	97.48	**95.85**	93.03
	✓	✓	89.20	96.87	95.59	**93.88**

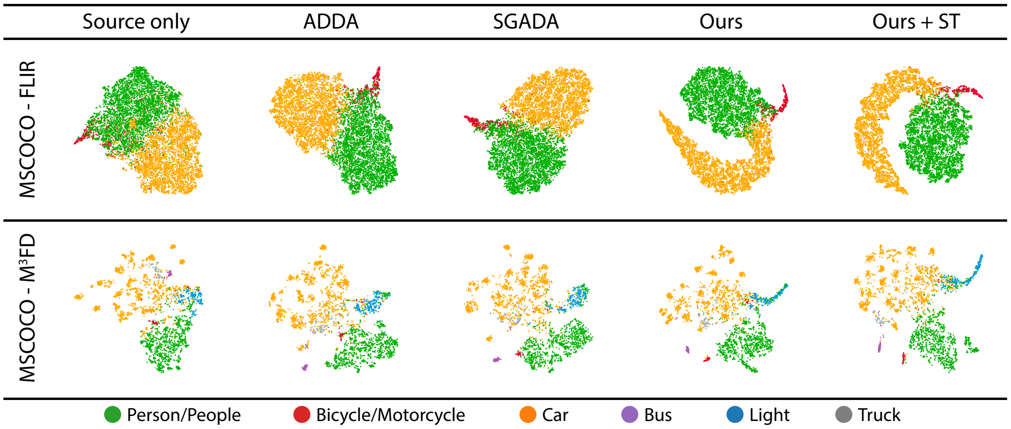


Figure 5.6: The t-SNE visualization of the encoded features of all target test samples by different methods (ST: Self-training).

3. θ_{sh+s}, θ_t: The training procedure given in Algorithm 2.

Table 5.2 lists the ablation study results using top-1 accuracy. From the table, alternatively training θ_{sh+s} and θ_t has better performance compared with the other two training strategies, and with both residual adapter and SE, it achieves the best result among all configurations. This observation aligns with the discussion in Sec. 5.4. In the following experiments, we use setting 3 with both attention modules for our method.

Unsupervised Thermal Image Classification

MS-COCO to FLIR ADAS: We first compare our method with SGADA [1] which achieves the best performance on MS-COCO to FLIR ADAS classification benchmark [24] and several other the-state-of-the-art general UDA methods.

MS-COCO [18] is a large-scale RGB dataset and FLIR ADAS [6] is a popular thermal image dataset for urban environments. We use the dataset prepared by [1] including three categories, i.e., bicycle, car and person, in this experiment. Same as [1], we train our network for 15 epochs with a batch size of 32. Per-class accuracy of all methods are given in Table 5.3, where the proposed method outperforms other methods by a significant margin even without self-training. In this particular data setting, the results indicate that our method (ours + ST) performs on par with fully-supervised training on target domain data (target only) while other UDA methods do not. Here, the fully-supervised "target only" method marks the gold-standard classification performance since the amount of target samples in this dataset is sufficiently large to directly apply fully-supervised learning.

MS-COCO to M^{3}FD: MS-COCO to FLIR ADAS dataset has 633440 unannotated target samples [1]. To further evaluate the adaptation performance when target training samples are scarce, we prepare a new RGB-to-thermal adaptation benchmark using MS-COCO and M^{3}FD [19] including 6 categories for evaluation, following the data preparation process in [1]. Examples and statistics of the prepared dataset are given in Fig. 5.4 and Table 5.1. Due to less training data, we train all networks for 30 epochs using a batch size of 32. We have similar observations from Table 5.4 as from previous experiment, except that all methods outperform the "target only" model which shows the effectiveness of UDA when sufficient annotated data is unavailable. The "target only" model performs poorly in this setting due to the dearth of target domain training data compared to the prior setting.

Experiment Analysis: We visualize the feature representations for all test samples on target domain using t-SNE [35] in Fig. 5.6, where the better feature separation in (d) and (e) shows our method can learn discriminative features for the given task. To examine the effectiveness of attention modules in our method, we further visualize the trained target-specific attentions in Fig. 5.5 by plotting the features they attend to. For SE modules, we plot the feature map with the highest and the lowest attention weight in the first residual block. As for residual adapters, we plot the Grad-CAM [30] of the last task-specific adapter layer.

We have several interesting observations. First, for bicycle and person categories, the feature maps that the SE highlights the most tend to have high activation on the object contour, which suggests that contour-sensitive features are more domain-invariant and transferable between RGB and thermal domains, aligning with the conclusion in [3]. Second, we notice that cars in thermal images are usually brighter

Table 5.3: Top-1 accuracy for MS-COCO to FLIR ADAS.

Method	Bicycle	Car	Person	Average
Source only	69.89	83.89	86.52	80.10
Pixel-DA [2]	62.53	89.99	76.73	76.42
DTA [16]	75.45	**97.65**	92.45	88.52
MCD-DA [29]	81.71	94.90	91.83	89.48
DANN [7]	78.16	95.07	96.24	89.82
CDAN [20]	78.16	97.10	94.82	90.03
ADDA [34]	86.67	96.95	89.10	90.90
SGADA [1]	87.13	94.44	92.03	91.20
Ours	89.20	96.87	95.59	93.88
Ours + ST	**89.63**	97.06	**96.03**	**94.24**
Target only	87.59	98.78	96.35	94.24

Table 5.4: Top-1 accuracy for MS-COCO to M^{3}FD.

Method	Bus	Car	Light	Motor.	People	Truck	Average
Source only	63.64	76.98	91.14	4.26	94.07	56.98	64.51
MCD-DA [29]	89.09	76.98	**95.36**	**76.59**	93.89	30.23	77.00
DANN [7]	89.09	82.72	51.90	68.09	92.15	74.42	76.4
CDAN [20]	89.09	**88.58**	72.15	46.81	93.98	45.35	72.7
ADDA [34]	**96.36**	85.86	60.34	51.06	76.73	**87.21**	76.26
SGADA [1]	94.55	87.22	70.04	51.06	77.01	81.40	76.88
Ours	90.91	85.37	72.57	74.47	93.80	51.16	78.05
Ours + ST	90.91	84.26	85.65	70.21	**95.44**	56.98	**80.57**
Target only	94.55	92.53	83.12	21.28	90.24	20.93	67.11

at the bottom due to the high temperature in those regions, as opposed to cars in RGB images which appear darker at the bottom due to shadows. The feature maps that the SE module attends to eliminate this phenomenon and appear visually similar to that of cars from an RGB image. Those observations show the effectiveness of our attention network in extracting domain-invariant and transferable features.

Unsupervised Thermal River Scene Segmentation

We present an effective and inexpensive approach for thermal semantic segmentation by adapting from synthetic RGB images using the proposed method, and test on thermal river scene segmentation. We collect 8 sequences of thermal images at

Table 5.5: Thermal segmentation performance before and after our adaptation using Intersection over Union (IoU).

Method	Non-water	Water	Average
Source only	78.33	32.90	55.62
Our adapted model	85.67	54.77	70.22

60 Hz using a hand-held FLIR ADK Longwave Infrared (LWIR) thermal camera with an NUC Ruby Mini PC at Big Bear Lake, CA. We sample images every 100 frames from the collected 48676 sequential frames and form an unlabeled training set of 486 thermal images. As our ultimate goal is to enable the nighttime coastline exploration ability of our aerial robots [41, 23] by thermal river segmentation, we manually annotate 282 diverse test images with pixel-level ground truth water labels for evaluation. Examples of collected thermal images are shown in Fig. 5.1 (4th column).

Due to the lack of annotated RGB dataset for natural scenes similar to our riverine environment, we generate synthetic RGB images with automatically obtained semantic labels using the AirSim simulator [31]. To that end, we use a publicly available simulation environment, i.e. the Landscape Mountains, and simulate a drone platform with a mounted RGB camera to follow a simple survey trajectory around rivers using the built-in simple flight controller. Following our thermal camera, we set the simulated RGB camera to have 75-degree FoV and capture 640×480 images. We acquire RGB images and the corresponding semantic labels at 2Hz, and obtain a synthetic labeled river scene RGB dataset of 1357 samples. We further convert the RGB images to grayscale and invert the intensity values (except for the foliage class), resulting in training samples visually close to our thermal images. Examples of the synthetic RGB images, semantic labels, and inverted grayscale images are shown in the first three columns of Fig 5.1.

We train the network for 50 epochs using a batch size of 8 without performing self-training. From Table 5.5, our adapted model obtains a performance gain of 14.6% mIoU and 21.87% water-class IoU over the source only model. The effectiveness of our method can be also seen in Fig. 5.1 and Fig. 5.7, where the adapted model corrects a large portion of false positive foliage prediction. This experiment demonstrates that thermal vision models can be effectively learned from synthetic RGB data using the proposed method without any manual annotations, even in the source domain.

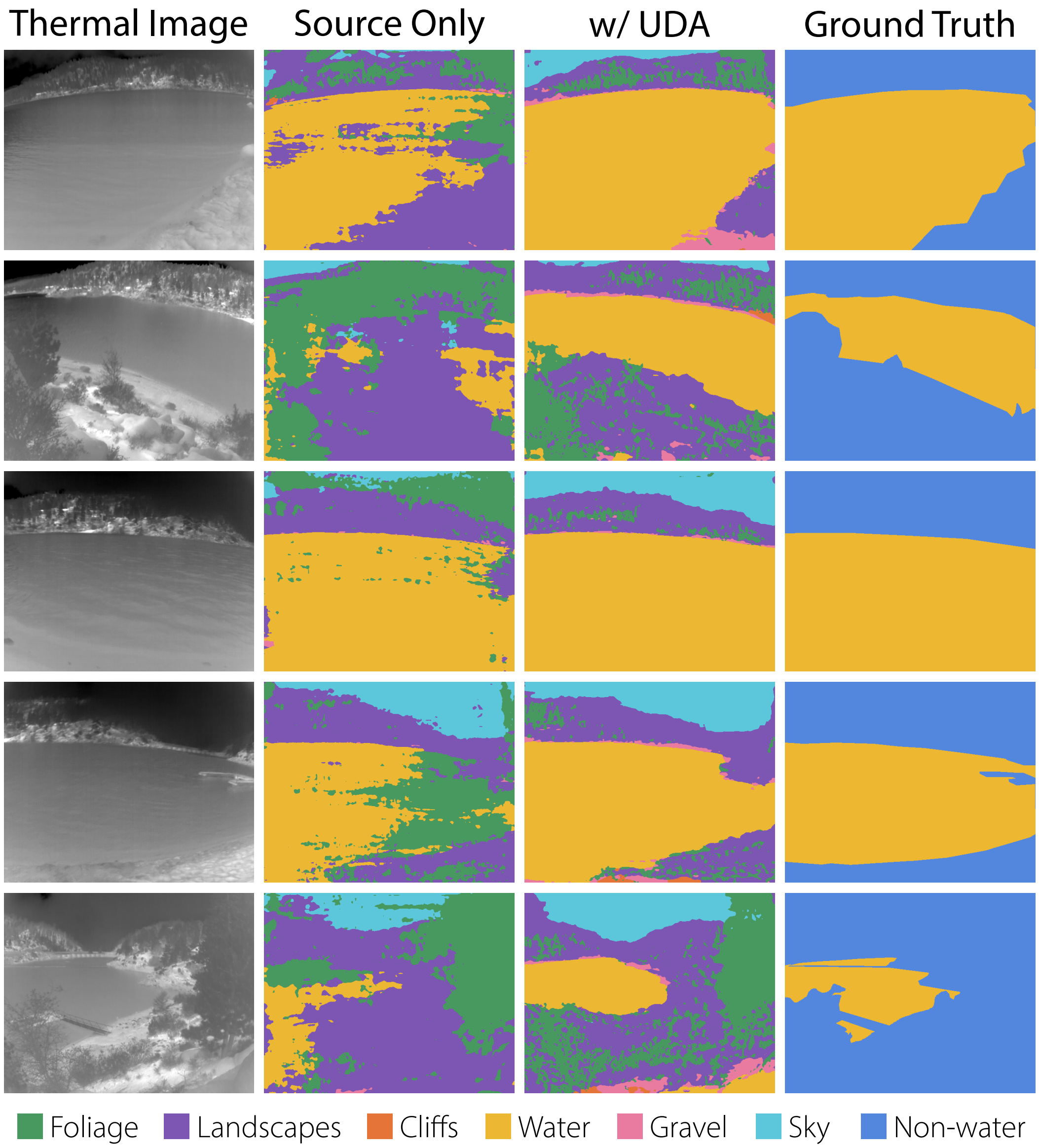


Figure 5.7: Qualitative results of our unsupervised thermal river segmentation model adapted from synthetic RGB data.

5.6 Conclusion

This work presented an unsupervised RGB-to-thermal domain adaptation method using multi-domain attention network and adversarial attention learning, and demonstrated its effectiveness on both image classification and semantic segmentation tasks. Vision models adapted by our method achieved a large performance gain over source-only models, and performed on-par with supervised models trained on target. The proposed method can enable robots thermal vision ability without incurring the exorbitant costs of data labeling. In addition, our adaptation method is designed to keep the source performance, i.e. learn without forgetting, providing a unified vision model for both RGB and thermal images.

Acknowledgements

This work was funded by the Ford Motor Company and the Office of Naval Research.

References

[1] I. B. Akkaya, F. Altinel, and U. Halici. “Self-Training Guided Adversarial Domain Adaptation for Thermal Imagery”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 2021, pp. 4322–4331.

[2] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. “Unsupervised pixel-level domain adaptation with generative adversarial networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 3722–3731.

[3] J. Chen, Z. Liu, D. Jin, Y. Wang, F. Yang, and X. Bai. “Light Transport Induced Domain Adaptation for Semantic Segmentation in Thermal Infrared Urban Scenes”. In: *IEEE Transactions on Intelligent Transportation Systems* 23.12 (2022), pp. 23194–23211.

[4] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.4 (2017), pp. 834–848.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. “Imagenet: A large-scale hierarchical image database”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Ieee. 2009, pp. 248–255.

[6] *Teledyne FLIR ADAS Dataset*. https://www.flir.com/oem/adas/adas-dataset-form/, Last accessed on 2023-10-27.

[7] Y. Ganin and V. Lempitsky. “Unsupervised domain adaptation by backpropagation”. In: *International conference on machine learning*. PMLR. 2015, pp. 1180–1189.

[8] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. “Domain-adversarial training of neural networks”. In: *Journal of Machine Learning Research* 17.1 (2016), pp. 2096–2030.

[9] H. Gao, J. Guo, G. Wang, and Q. Zhang. “Cross-Domain Correlation Distillation for Unsupervised Domain Adaptation in Nighttime Semantic Segmentation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2022, pp. 9913–9923.

[10] Q. Ha, K. Watanabe, T. Karasawa, Y. Ushiku, and T. Harada. “MFNet: Towards real-time semantic segmentation for autonomous vehicles with multi-spectral scenes”. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2017, pp. 5108–5115.

[11] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, et al. “A survey on vision transformer”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022).

[12] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell. “CyCADA: Cycle-consistent adversarial domain adaptation”. In: *International conference on machine learning*. Pmlr. 2018, pp. 1989–1998.

[13] J. Hu, L. Shen, and G. Sun. “Squeeze-and-excitation networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7132–7141.

[14] Y.-H. Kim, U. Shin, J. Park, and I. S. Kweon. “MS-UDA: Multi-spectral unsupervised domain adaptation for thermal image semantic segmentation”. In: *IEEE Robotics and Automation Letters* 6.4 (2021), pp. 6497–6504.

[15] Z. Kütük and G. Algan. “Semantic Segmentation for Thermal Images: A Comparative Survey”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2022, pp. 286–295.

[16] S. Lee, D. Kim, N. Kim, and S.-G. Jeong. “Drop to adapt: Learning discriminative features for unsupervised domain adaptation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 91–100.

[17] Z. Li and D. Hoiem. “Learning without forgetting”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.12 (2017), pp. 2935–2947.

[18] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. “Microsoft COCO: Common objects in context”. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2014, pp. 740–755.

[19] J. Liu, X. Fan, Z. Huang, G. Wu, R. Liu, W. Zhong, and Z. Luo. “Target-aware Dual Adversarial Learning and a Multi-scenario Multi-Modality Benchmark to Fuse Infrared and Visible for Object Detection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2022, pp. 5802–5811.

[20] M. Long, Z. Cao, J. Wang, and M. I. Jordan. “Conditional adversarial domain adaptation”. In: *Proceedings of the Advances in Neural Information Processing Systems Conference* 31 (2018).

[21] K.-K. Maninis, I. Radosavovic, and I. Kokkinos. “Attentive single-tasking of multiple tasks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 1851–1860.

[22] M. A. Marnissi, H. Fradi, A. Sahbani, and N. E. B. Amara. “Unsupervised thermal-to-visible domain adaptation method for pedestrian detection”. In: *Pattern Recognition Letters* 153 (2022), pp. 222–231.

[23] K. Meier, S.-J. Chung, and S. Hutchinson. “River segmentation for autonomous surface vehicle localization and river boundary mapping”. In: *Journal of Field Robotics* 38.2 (2021), pp. 192–211.

[24] *MSCOCO to FLIR Adas benchmark (unsupervised domain adaptation)*.

[25] F. Qingyun, H. Dapeng, and W. Zhaokui. “Cross-modality fusion transformer for multispectral object detection”. In: *arXiv preprint arXiv:2111.00273* (2021).

[26] S.-A. Rebuffi, H. Bilen, and A. Vedaldi. “Efficient parametrization of multi-domain deep neural networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8119–8127.

[27] S.-A. Rebuffi, H. Bilen, and A. Vedaldi. “Learning multiple visual domains with residual adapters”. In: *Proceedings of the Advances in Neural Information Processing Systems Conference* 30 (2017).

[28] S. Ren, K. He, R. Girshick, and J. Sun. “Faster r-cnn: Towards real-time object detection with region proposal networks”. In: *Advances in neural information processing systems* 28 (2015).

[29] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada. “Maximum classifier discrepancy for unsupervised domain adaptation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 3723–3732.

[30] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. “Grad-CAM: Visual explanations from deep networks via gradient-based localization”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 618–626.

[31] S. Shah, D. Dey, C. Lovett, and A. Kapoor. “AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles”. In: *Field and Service Robotics*. 2017. eprint: arXiv:1705.05065.

[32] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker. “Learning to adapt structured output space for semantic segmentation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7472–7481.

[33] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. “Simultaneous deep transfer across domains and tasks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 4068–4076.

[34] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. “Adversarial discriminative domain adaptation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 7167–7176.

[35] L. Van der Maaten and G. Hinton. “Visualizing data using t-SNE.” In: *Journal of machine learning research* 9.11 (2008).

[36] J. Vertens, J. Zürn, and W. Burgard. “HeatNet: Bridging the day-night domain gap in semantic segmentation with thermal images”. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2020, pp. 8461–8468.

[37] X. Wang, L. Li, W. Ye, M. Long, and J. Wang. “Transferable attention for domain adaptation”. In: *Proceedings of the AAAI National Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 5345–5352.

[38] G. Wilson and D. J. Cook. “A survey of unsupervised deep domain adaptation”. In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 11.5 (2020), pp. 1–46.

[39] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon. “CBAM: Convolutional block attention module”. In: *Proceedings of the European Conference on Computer Vision*. 2018, pp. 3–19.

[40] Q. Xu, Y. Zhou, W. Wang, C. R. Qi, and D. Anguelov. “SPG: Unsupervised domain adaptation for 3d object detection via semantic point generation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2021, pp. 15446–15456.

[41] J. Yang, A. Dani, S.-J. Chung, and S. Hutchinson. “Vision-based localization and robot-centric mapping in riverine environments”. In: *J. of Field Robotics* 34.3 (2017), pp. 429–450.

[42] C. Zhang, Q. Zhao, and Y. Wang. “Transferable attention networks for adversarial domain adaptation”. In: *Information Sciences* 539 (2020), pp. 422–433.

[43] H. Zhang, E. Fromont, S. Lefèvre, and B. Avignon. “Guided attentive feature fusion for multispectral pedestrian detection”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2021, pp. 72–80.

[44] H. Zhao, R. T. Des Combes, K. Zhang, and G. Gordon. “On learning invariant representations for domain adaptation”. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 7523–7532.

[45] W. Zhou, S. Dong, C. Xu, and Y. Qian. “Edge-Aware Guidance Fusion Network for RGB–Thermal Scene Parsing”. In: *Proceedings of the AAAI National Conference on Artificial Intelligence*. Vol. 36. 3. 2022, pp. 3571–3579.

Chapter 6

AERIAL RGB-THERMAL DATASET IN THE WILD

[1] C. Lee*, M. Anderson*, N. Raganathan, X. Zuo, K. Do, G. Gkioxari, and S.-J. Chung. "CART: Caltech Aerial RGB-Thermal Dataset in the Wild". In: *arXiv preprint arXiv:2403.08997* (2024). Available at https://arxiv.org/abs/2403.08997.

6.1 Abstract

We present the first publicly available RGB-thermal dataset designed for aerial robotics operating in natural environments. Our data-set captures a variety of terrains across the continental United States, including rivers, lakes, coastlines, deserts, and forests, and consists of synchronized RGB, long-wave thermal, global positioning, and inertial data. Furthermore, we provide semantic segmentation annotations for 10 classes commonly encountered in natural settings in order to facilitate the development of perception algorithms robust to adverse weather and nighttime conditions. Using this dataset, we propose new and challenging benchmarks for thermal and RGB-thermal semantic segmentation, RGB-to-thermal image translation, and visual-inertial odometry. We present extensive results using state-of-the-art methods and highlight the challenges posed by temporal and geographical domain shifts in our data.

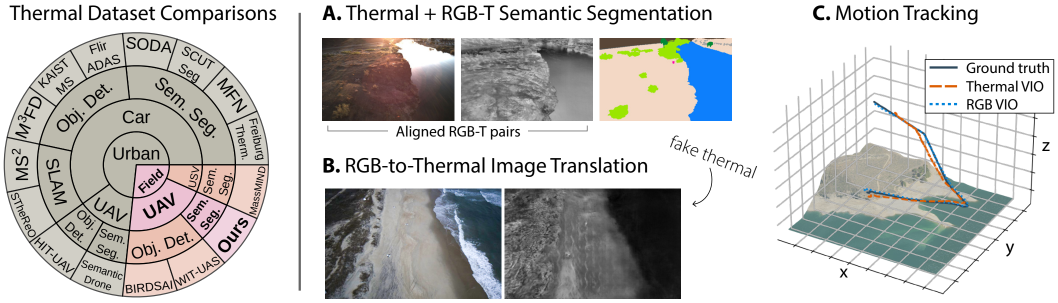


Figure 6.1: *Left:* Our dataset is the first dataset designed to improve thermal semantic perception for field robotics. *Right:* Our dataset provides new thermal and RGB-thermal benchmarks for field robotics and computer vision algorithms, encompassing (a) semantic segmentation, (b) image translation, and (c) motion tracking.

6.2 Introduction

Current field robots rely predominantly on sensors such as visual cameras, lidar, and radar to perceive their surroundings [16, 18]. While these sensors enhance robustness of downstream vision algorithms, their performance degrades in low-light and adverse weather (snow, fog, rain) conditions [68]. In contrast, thermal cameras exploit long-wave infrared wavelengths to capture emitted heat, offering dense, visual information even in conditions in which other methods struggle [19]. Recently, thermal cameras have been used in multimodal perception algorithms for autonomous vehicle applications and are increasingly being explored for field robotic applications to enable nighttime autonomy [36, 74, 69, 11, 76, 48, 12]. However, successful integration of thermal imagery requires extensive datasets across diverse settings, preventing widespread adoption in field robotics.

Existing thermal datasets primarily focus on urban environments for autonomous driving applications (Fig. 6.1). They typically comprise of thermal-only imagery [74, 36] or RGB-Thermal (RGB-T) pairs [17, 22, 69, 40], with some including global positioning (GPS/GNSS) and inertial measurements (IMU) for visual-inertial odometry (VIO) and simultaneous localization and mapping (SLAM) [76, 11, 60]. While comprehensive, these datasets rarely extend beyond urban areas. They lack data depicting natural settings like rivers and forests, which are typical operating areas for field robotics that perform coastline mapping and bathymetry [4] or monitor activity during forest fires [30].

Due to the lack of thermal benchmarks for algorithms like semantic segmentation and SLAM, field robots cannot easily operate at night or in adverse conditions, requiring online learning [70, 34] or unsupervised domain adaptation to compensate [20, 32, 69] which still require labeled data for evaluation. As such, enabling thermal perception for field robotics requires collecting and annotating field-specific data from scratch. However, curating such a dataset is more complex than doing so for RGB: First, thermal data is hard to crowdsource and web-scrape due to the cost and nicheness of thermal cameras. Hence, acquiring a dataset for algorithm development requires physically going to distinct field environments, increasing time and cost. Second, operating most field robotics, especially uninhabited aerial vehicles (UAV), usually require special permits unique to each location-of-capture. This further complicates the creation of a comprehensive thermal dataset for field robotics research.

In this work, we present the first dataset targeting thermal scene perception for field robotics and for wider use by the computer vision community. The dataset consists of oblique-facing imagery captured from a UAV, supplemented with image sets captured at ground level, and focuses on littoral settings within various desert, forest, and coastal environments across the United States. We contribute new benchmarks for thermal and RGB-T semantic segmentation, RGB-T image translation, and VIO/SLAM algorithms, with unique challenges characterized by temporal and geographical domain shift for learning-based methods and periodic feature sparsity for motion tracking algorithms.

This chapter is organized as follows: Section 6.3 reviews relevant datasets, benchmarks, and algorithms. Section 6.4 describes our dataset details and curation process, and Section 6.5 presents benchmark results. Section 6.6 offers concluding remarks.

6.3 Related Work

Datasets and Benchmarks

Thermal/RGB-T benchmarks for urban robotics: Current urban thermal and RGB-T datasets primarily focus on autonomous vehicle (AV) technology and surveillance applications. AV-related datasets cover tasks such as object detection [17, 40, 11], semantic segmentation [36, 22, 69, 74], and localization [76, 11, 60]. Some include data from other spectra, benchmarking not only thermal algorithms but also RGB-T [40, 11, 22, 69] and other multispectral models. In contrast, datasets for surveillance applications use fixed cameras [29] and UAVs [65] to detect personnel, vehicles, and other objects. Despite many urban thermal benchmarks, they cannot be directly used to develop algorithms for robots operating in natural environments due to the urban/non-urban domain gap.

Thermal/RGB-T benchmarks for field robotics: Datasets for robots operating in non-urban environments remain limited. MassMIND [48] and PST900 [61] benchmark thermal semantic segmentation for maritime and subterranean robotic environments, respectively. WIT-UAS [30] provides UAV-borne thermal images for object detection in wildfire-prone environments. [34] proposes an aerial RGB-T dataset for thermal water segmentation in littoral areas but only provides 1272 labels consisting of a single *water* class. Benchmarks like [3, 7] depict natural environments but are targeted towards wildlife tracking. To address this gap, we present an aerial RGB-T dataset which can be used to benchmark multiple tasks, including semantic segmentation, RGB-T image translation, and localization, across diverse natural environment robotic applications. Specifically, we build upon the dataset from [34] by extending the number of semantic classes from 1 to 10 and adding additional RGB-T data captured from air and ground, resulting in a total of 4195 semantic segmentation annotations.

UAV-based semantic segmentation datasets: Aside from [34], existing UAV-captured semantic segmentation datasets [47, 10, 44, 6, 46] cover urban environments. Wth the exception of [46] which provides 400 thermal image samples, all of these datasets contain only RGB data. Although they provide images with similar view angles as in our dataset, the RGB-T modality gap and their focus on urban semantic content makes them impractical for field robotic applications.

Algorithms using Thermal Datasets

Thermal semantic segmentation: Since thermal images are spatially identical to their RGB counterparts, popular RGB semantic segmentation algorithms [52, 8, 5, 72] can be applied directly. However, some works [36, 74, 50] develop models specifically for thermal imagery, and leverage edge priors to overcome low resolution and blurriness due to thermal crossover [55].

Although specifically designed for thermal, these models only marginally outperform standard RGB segmentation models [50] and their performance in field settings, where edges are less structured, remains untested. Aside from architectural advancements, improvements in thermal semantic segmentation come from domain adaptation (DA) methods [20, 36, 69, 67] that use labeled RGB data with unlabeled thermal data. To provide fair comparisons, we do not provide DA baselines in our benchmarks as performance varies based on the choice of RGB data.

RGB-T semantic segmentation: RGB-T semantic segmentation methods compensate for the weaknesses of each modality via deep feature fusion, data augmentation, and adversarial training. Most approaches [22, 64, 63, 59, 77] utilize encoder-decoder architectures with modality-specific encoders and shared decoders, and show significant improvements over channel-stacked RGB-T inputs. Works like [38, 79] make improvements by integrating RGB and thermal features at multiple stages within the encoders, while [61] eschews RGB-T training pairs by processing RGB-T inputs sequentially. Other methods include random input masking to reduce reliance on a single modality [59] and adversarial training to adapt to different times of day [69].

RGB-T image translation: RGB-T image translation is vital as it may enable thermal training data to be generated from large-scale RGB datasets. Works based on generative adversarial networks (GAN) such as [41, 27, 80, 51, 35] can be used without RGB-T image pairs. However, GANs like [28, 71] that require paired imagery provide better translations for both visual appeal and domain adaptation [36]. Recent diffusion methods [57, 56] have demonstrated remarkable results on RGB imagery, but have not been tested on RGB-T image translation.

Thermal VIO and SLAM: Low signal-to-noise ratio, reduced contrast, and blurred boundaries in thermal images present challenges for VIO and SLAM. Several studies [2, 13, 24, 26, 14] use indirect methods by tracking keypoints extracted from thermal images for visual constraints and rely on image normalization techniques to improve keypoint extraction. In contrast, [31] uses a direct method to extract and track high-gradient points by minimizing radiometric error. Other works utilize learned local and global features to improve feature tracking [78] and loop closure detection [58], respectively.

6.4 The Aerial RGB-Thermal Dataset

Data Acquisition Hardware

To capture this dataset, we developed a custom sensor stack with synchronized RGB-thermal imagery, IMU, and global positioning data (Fig. 6.2). The sensor stack features a non-radiometric FLIR ADK thermal camera (640×512 px and 75° horizontal FoV, 60 Hz) flanked by a pair of FLIR Blackfly electro-optical (EO) monochrome and color cameras[1] (960×600 px and 75° horizontal FoV, 30 Hz). Pose information is provided by a VectorNav VN100 IMU (200 Hz) and a u-blox M8N GPS (5 Hz). All three cameras and the VN100 are hardware synchronized using a dedicated signal generator and are rigidly attached to each other using a stiff, 3D printed mount. A Simply NUC Ruby Mini PC (AMD Ryzen 7 4800U, 32 GB RAM) is used for the compute, and the entire system is interfaced using ROS Noetic. Our cameras and IMU were calibrated following the procedures found in Supplementary Material Sec S2.2 and were not disassembled from each other between any of the dataset collections.

The sensor stack is mounted rigidly onboard an Aurelia X6 hexacopter (Fig. 6.2) without gimbal stabilization at angles between 0° (level with the horizon) to 45° downwards to provide different viewpoints. When mounted to a UAV, we also log the position and attitude estimates from the UAV at 20 Hz providing improved po-

[1] BFLY-U3-23S6M-C mono/color with Tamron M112FM08 lenses

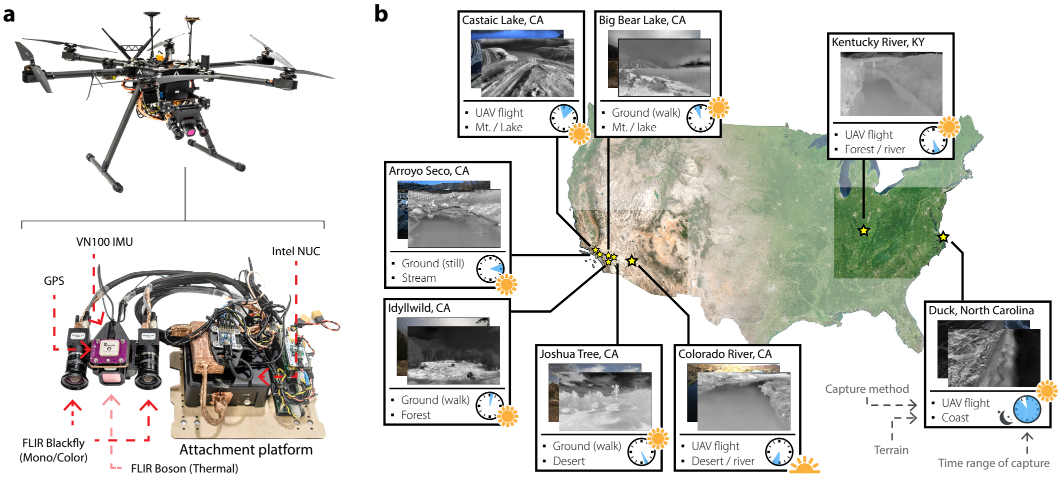


Figure 6.2: **(a)** The Aurelia X6 hexacopter and sensor stack used to capture our aerial dataset. **(b)** Geographic distribution of our data collection sites and collection times.

sitioning accuracy from the onboard RTK GPS. Additionally, the sensor stack's modularity allows us to capture datasets on foot by mounting it to a tripod where flight restrictions are in-place.

Data Collection and Processing

Data capture: We captured 37 aerial and ground trajectories covering lakes, rivers, coastlines, deserts, and mountains from around Southern California, Kentucky, and North Carolina (Fig. 6.2). Aerial trajectories (18) involve high motion with intermittent hovering, and mostly depict scenes from 40 m altitude. Ground-level trajectories (19) were captured on foot, with 7 captured without any movement for VIO debugging purposes. For more details, see Supplementary Materials Tab. S2 and Sec. S2.1.

Thermal image normalization: For the baselines presented in this work, we typically normalize 16-bit thermal data by rescaling between the 1st and 99th percentile pixel values and follow with contrast limited adaptive histogram equalization (CLAHE). We keep normalized data as floats to minimize information loss, and only convert to 8-bit format when necessary or for visualization. **Semantic segmentation annotation:** To aid development of scene perception algorithms for field robots, we annotated a subset of thermal images for semantic segmentation with classes in Fig. 6.3. To avoid redundancy, images were sampled every 3 s when moving and every 20 s when still, resulting in 4195 samples. We considered a frame in motion if the distance moved within the past 2 seconds exceeded 0.5 meters, with

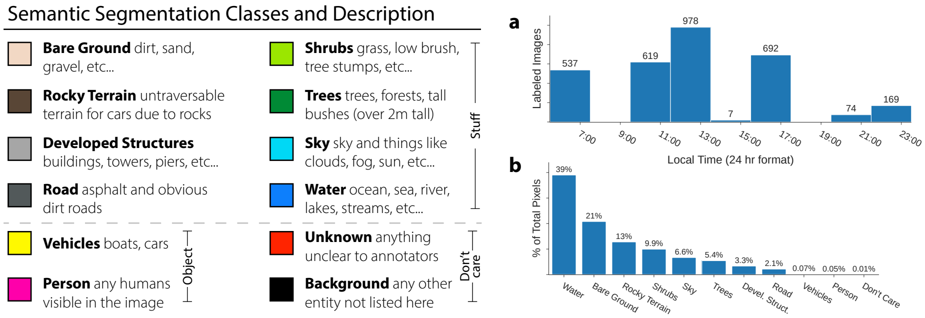


Figure 6.3: Semantic segmentation classes in our dataset. The color mapping is used throughout this paper. **(a)** Hourly distribution of annotated thermal images. **(b)** Histogram of semantic classes.

distances computed using GPS coordinates. Labeling was outsourced to an external contractor and underwent 3 rounds of review. More details of the annotation process can be found in the supplement (Sec. S2.3).

RGB-thermal image alignment: To align thermal and RGB image pairs, we stereo rectified the RGB-T image pairs using the camera matrices obtained from calibration (Sec. S2.2) and projected the thermal image into the larger RGB image frame to preserve RGB resolution. As our baseline is small compared to the depth of the scenes we capture, image pairs are near-coregistered after rectification. Due to differences in optics, the field-of-view of the aligned image pairs is narrower compared to that of the thermal camera.

Dataset Splits

General (benchmark) split: This is our primary split for semantic segmentation and image translation benchmarks. We randomly partition the annotated thermal dataset (Section 6.4) into train/val/test sets at a 75:12.5:12.5 ratio.

Temporal split: To promote studies into the effect of different time-of-day capture of the thermal images, we split the dataset using three time periods: twilight/sunrise (5 AM - 7 AM), daytime (10 AM - 5 PM), and nighttime (7 PM - 4 AM). We randomly partition the daytime images into train/val/test sets using the ratio from above and leave the twilight and nighttime sets for testing.

Geographical split: The geographical splits are intended to test algorithm adaptability to unseen settings. We provide two splits:

1. *Terrain-based:* This split partitions data into the following terrain categories: aerial river, aerial lake, aerial coast, and ground-captured images.

Train/val/test splits are created per category with a 50:15:35 ratio.

2. *Region-based:* This split partitions data based on the general area of the United States that the data was captured. The regions include California (CA), Kentucky (KY), and North Carolina (NC), with train/val/test splits created in the same manner as the terrain split.

6.5 Experiments

We provide baselines for the following benchmarks: thermal semantic segmentation, RGB-T semantic segmentation, RGB-T image translation, and VIO/SLAM. Furthermore, we explore the zero-shot thermal segmentation capabilities of large vision-language models on our dataset and perform further ablation studies.

Table 6.1: Supervised model performance (mIoU), framerate (FPS at batch size 1), and floating point operations (FLOP) for various semantic segmentation network baselines.

Model	Bare ground	Rocky terrain	Develop. struct.	Road	Shrubs	Trees	Sky	Water	Vehicles	Person	Stuff	Object	All	Orin (GPU)	NUC (CPU)	FLOPS (G)
FastSCNN [52]	0.807	0.907	0.728	0.631	0.694	0.776	0.956	0.965	0.384	0.058	0.808	0.221	0.690	**124.2**	**85.6**	**1.0**
MobileNetV3-S 0.75 [25]	0.802	0.914	0.676	0.633	0.704	0.814	0.955	0.976	0.383	0.111	0.809	0.247	0.697	75.2	24.0	3.3
MobileNetV3-L 0.75	0.801	0.914	0.701	0.615	0.713	0.793	0.959	**0.976**	0.432	0.148	0.809	0.290	0.705	59.0	15.5	4.8
MobileViTV2 0.50 [45]	0.822	0.913	0.759	0.650	0.694	0.781	0.954	0.970	0.222	0.030	0.818	0.126	0.680	60.1	10.7	5.5
EfficientViT-B0 [5]	0.825	0.917	**0.777**	0.618	0.710	0.793	0.961	0.965	0.495	**0.191**	0.821	**0.343**	**0.725**	51.2	5.5	3.9
EfficientNet-Lite0 [66]	0.819	0.908	0.769	0.601	0.709	0.805	0.962	0.968	0.380	0.079	0.818	0.229	0.700	47.7	12.8	7.2
EfficientNet-Lite2	**0.825**	0.920	0.765	0.614	0.698	0.805	0.958	0.971	0.386	0.118	0.820	0.252	0.706	38.9	11.2	9.4
Segformer-B0 [72]	0.804	0.910	0.690	0.624	0.696	0.788	0.960	0.970	0.195	0.000	0.805	0.097	0.664	38.3	5.9	10.2
Segformer-B1	0.814	0.909	0.773	0.603	0.713	0.799	0.960	0.965	0.366	0.000	0.817	0.183	0.690	30.3	3.5	19.9
ResNet18 [23]	0.810	0.905	0.766	0.618	0.711	0.807	0.959	0.966	0.431	0.158	0.818	0.294	0.713	69.3	12.4	22.4
ResNet50	0.819	0.916	0.747	**0.658**	0.711	0.799	0.961	0.974	0.361	0.182	0.823	0.272	0.713	29.9	6.1	45.4
ResNeXt50 [73]	0.825	0.919	0.773	0.653	0.709	0.794	0.964	0.976	0.436	0.137	**0.827**	0.287	0.719	23.0	5.6	45.5
ConvNext-T [43]	0.799	0.909	0.719	0.608	0.706	0.808	0.961	0.970	0.363	0.003	0.810	0.183	0.685	25.8	3.3	47.4
ConvNext-S	0.810	0.913	0.771	0.603	**0.718**	0.810	0.965	0.964	0.492	0.048	0.819	0.270	0.709	19.1	2.3	75.0
ConvNext-B	0.810	**0.921**	0.697	0.611	0.706	0.803	0.963	0.972	0.348	0.112	0.811	0.230	0.694	14.6	1.4	130.5
ConvNext-B (CLIP) [54]	0.813	0.918	0.683	0.632	0.718	0.813	**0.965**	0.972	**0.517**	0.137	0.814	0.327	0.717	14.6	1.4	130.4
ConvNext-B (CLIP) ❄	0.773	0.887	0.713	0.487	0.656	0.756	0.946	0.951	0.158	0.019	0.771	0.089	0.635	14.6	1.4	130.4
DINOv2 [49] (linear head)	0.800	0.907	0.681	0.606	0.693	0.796	0.956	0.967	0.375	0.079	0.801	0.227	0.686	15.2	2.1	—
DINOv2 (linear head) ❄	0.705	0.844	0.619	0.419	0.558	0.725	0.922	0.920	0.356	0.158	0.714	0.257	0.623	15.2	2.1	—
DINOv2 (nonlin. head)	0.810	0.916	0.708	0.635	0.700	**0.819**	0.959	0.973	0.399	0.119	0.815	0.259	0.704	15.2	2.1	—
DINOv2 (nonlin. head) ❄	0.796	0.903	0.732	0.606	0.691	0.794	0.954	0.963	0.471	0.149	0.805	0.310	0.706	15.2	2.1	—
FTNet† [50]	0.755	0.867	0.635	0.576	0.643	0.653	0.787	0.947	0.234	0.024	0.733	0.129	0.613	11.1	1.4	100.5

†Thermal-specific model ❄ Frozen encoder

Thermal Semantic Segmentation

We run multiple baselines on our general split before choosing a specific model to test on the geographic and temporal splits. We go over our baselines before analyzing their performance on our benchmarks.

Baselines: Motivated by robotic applications, we are interested in both inference speed and segmentation performance. Our baselines place an emphasis on lightweight, real-time networks, but also on foundation models like DINOv2 [49] and ConvNext-B (CLIP) [43, 54] to explore any latent multimodal capabilities. We also test a thermal-specific model, FTNet [50], which uses an edge-based loss function to compensate for blurred boundaries in thermal images.

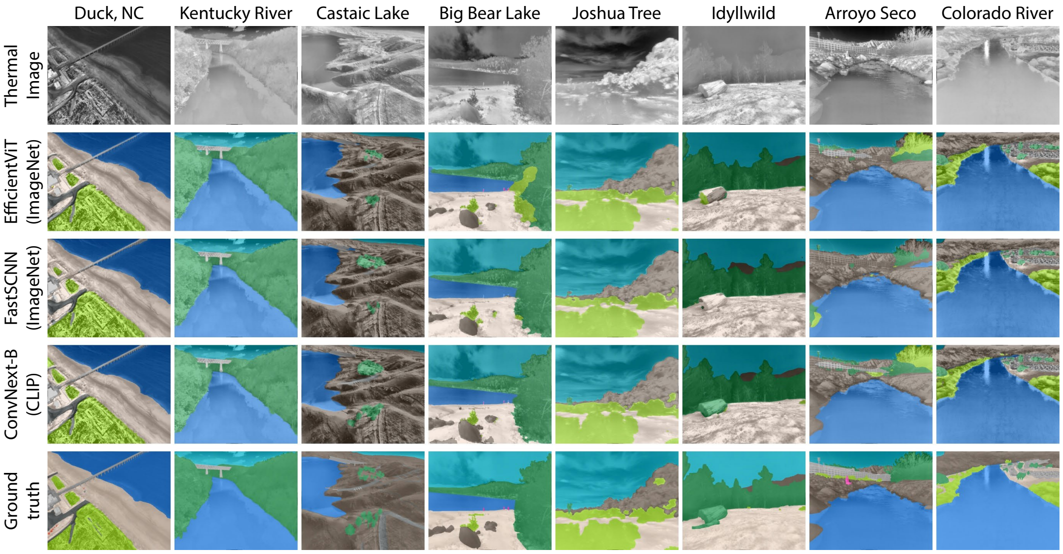


Figure 6.4: Thermal images and semantic segmentation labels from each capture area with inference results from EfficientViT, FastSCNN, and ConvNext-B (CLIP).

Besides FastSCNN [52], EfficientViT [5], Segformer [72], DINOv2, and FTNet, all network encoders in Table 6.1 feed into a DeepLabV3+ segmentation head. In particular, DINOv2 was employed with linear and non-linear, multi-scale heads in order to study the generalization capacity of its pretrained features to the thermal modality. All baselines except DINOv2 and ConvNext-B (CLIP) were trained from pretrained ImageNet weights. All networks, besides FTNet, were trained using the cross entropy loss. Further details on baseline implementation and training can be found in Supplementary Material Sec. S3.1.

General benchmark: We use the general (random) split (Section 6.4) to compare thermal semantic segmentation performance between baselines (Table 6.1). With field robotic applications in mind, we analyze segmentation results in context of GPU and CPU frame rates, which were measured onboard the Nvidia Jetson AGX Orin (GPU) and the Ruby NUC (CPU) embedded platforms, respectively[2].

In general, we found that larger models provide marginal benefit over smaller, compute-efficient models. In particular, a 0.027 mIoU difference between the largest (ConvNext-B) and fastest (FastSCNN) baselines comes at the cost of 130× more FLOPs. Overall, EfficientViT-B0 performed the best, attaining the highest mIoU (0.725) while performing 50 Hz inference on the Jetson AGX Orin.

While large RGB foundation models show little gain over ImageNet pretraining,

[2] Models of input size 512×640 were converted to ONNX and inference speed was averaged over 50 forward passes after a warm-up period.

their features do extend to the thermal modality. Specifically, a frozen DINOv2 (linear head) outperforms the thermal-specific FT-Net. With a nonlinear head, it matches the performance of other end-to-end trained networks. While this shows the benefit of large-scale RGB pretraining, the choice of RGB pretraining data or learning strategy seems to matter. Notably, the frozen ConvNext-B (CLIP) model gives marginal gain over the frozen DINOv2 (linear head), despite a larger, nonlinear segmentation head. Overall, foundation models do not confer significant performance advantages for semantic segmentation on our dataset in order to justify their computational cost.

Overall, all baselines can segment *stuff* classes well but none excel with the rare *object* classes: *vehicle* and *person*. Future models would likely need to consider weighted loss functions, smarter data augmentation, or domain adaptation techniques in order to improve on object classes. Lastly, we note that the thermal-specific model, FTNet, performs poorly, likely as its edge priors are less effective in unstructured, natural environments compared to urban environments.

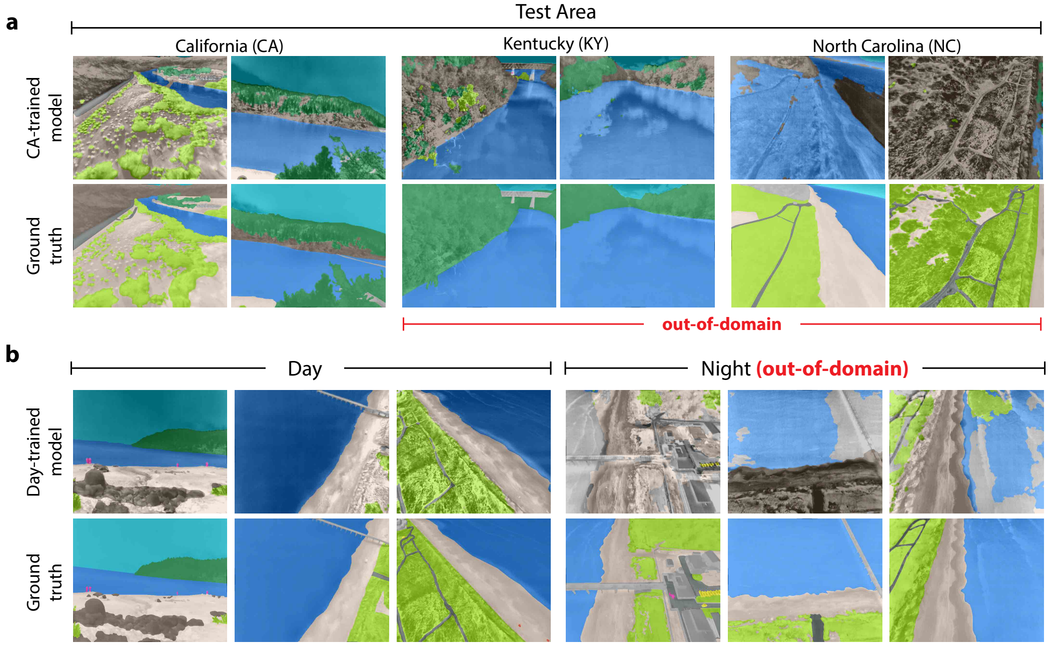


Figure 6.5: (a) Thermal semantic segmentation failures due to intra-class semantic variations when testing on geographically-partitioned, *out-of-domain* data. (b) Failures due to photometric variations between day *(in-domain)* and night *(out-of-domain)*.

Geographically-partitioned benchmarks: We retrained our best baseline (Tab. 6.1) on geographically-partitioned sets to evaluate intra-thermal generalization. Given the cost and effort of collecting diverse aerial thermal data of field settings, we aim to determine the least amount of annotated data required to effectively deploy. We performed two experiments using EfficientViT-B0: one on generalization across geographic regions and another across differing terrain types. In the first experiment, we trained on thermal data from CA and tested on images from KY and NC (Table 6.2a). Despite common classes in each region, results show poor generalization to class variations across geographic areas. We see this again when repeating with the NC split. In the second experiment, we divided our data into three terrain types, *aerial river*, *aerial lake*, and *aerial coast*. Remaining ground-captured images were lumped into a fourth *ground* category. We trained and tested all pairwise combinations and found poor generalization across terrain types as well (Table 6.2b). Notably, we found low, in-domain performance for the lake setting but saw gains after training on aerial data from other terrain and further improvements after training on ground level images. Overall, our results reiterate the benefit of having training data representative of the testing environment, but also show that gains can be made by including out-of-domain (OOD) data from different terrain and different view points.

Overall, the geographic benchmark presents a difficult domain adaptation problem, requiring both inter- and intra-modality domain adaptation techniques to overcome. Although such techniques could perform well in this challenge, the results also highlight a need for an even larger and more diverse thermal dataset capturing common field environments than what we presented.

Temporally-partitioned benchmark:

Here, we assess EfficientViT-B0's performance against intra-thermal, temporal distribution shift due to thermal inversion. We train the model on images from the daytime and test on images captured at sunrise and at night. The model achieved mIoUs of 0.242 at sunrise, 0.193 at night, and 0.777 on a separate daytime test

Table 6.2: Thermal semantic segmentation results on geographically-split data

(a) Region-based split

Train Area	Test Area			Avg.
	CA	NC	KY	mIoU
CA	0.666	0.084	0.193	0.314
NC	0.084	0.508	0.031	0.208
All	0.648	0.525	0.587	0.586

(b) Terrain-based split

Training Terrain	Testing Terrain			Avg.
	River	Lake	Coast	mIoU
River	0.668	0.196	0.117	0.327
Lake	0.196	0.364	0.061	0.207
Coast	0.127	0.119	0.522	0.256
All	0.646	0.415	0.493	0.518
All + ground	0.658	0.416	0.520	0.532

Table 6.3: RGB-Thermal semantic segmentation network baseline results (mIoU).

Model	Bare ground	Rocky terrain	Devel. struct.	Road	Shrubs	Trees	Sky	Water	Vehicles	Person	All	FLOPS (G)
RGB Only†	0.790	0.873	0.808	0.559	0.664	0.752	0.891	0.967	0.399	0.000	0.670	45
Thermal Only†	0.766	0.835	0.789	0.536	0.654	0.749	0.907	0.971	0.399	0.000	0.661	45
EAEFNet [38]	0.805	0.863	0.813	0.545	0.702	0.802	**0.937**	0.976	0.481	0.000	0.692	358
CRM [59]	0.816	**0.900**	0.851	**0.616**	0.695	0.779	0.923	0.974	0.377	**0.206**	0.714	310
CMNeXt [77]	**0.830**	**0.900**	**0.861**	0.607	**0.718**	**0.808**	0.933	**0.980**	**0.560**	0.190	**0.740**	148

† DeepLabV3+ (w/ ResNet50 encoder)

split. Despite training on daytime images from the same location, the model fails on nighttime scenes, struggling to classify *water* due to thermal inversion (Fig. 6.5b). As such, to perform well on this split, future algorithms need to augment datasets to simulate such inversions or intentional collect data capturing such phenomena.

RGB-T Semantic Segmentation

We use the paired RGB-T dataset (Section 6.4) and partition it using the general split (Section 6.5). Recall that RGB and thermal input sizes are 960×600 pixels and the thermal FoV is narrower than in the previous benchmarks (Section 6.5).

Baselines: We test three RGB-T semantic segmentation algorithms achieving state-of-the-art results in urban RGB-T scene segmentation benchmarks: CRM [59], EAEFNet [38], and CMNeXt [77]. To compare against single-modality models, we train two additional ResNet50/DeepLabV3+ segmentation models on RGB and thermal data accordingly (Section 6.5).

Performance analysis: All RGB-T models outperform the single-modality baselines, with CMNeXt outperforming the next best RGB-T model by 0.026 mIoU (Table 6.3). Overall, incorporating RGB imagery into the segmentation pipeline greatly improved the distinction between land-based *stuff* classes which can be hard to distinguish in thermal due to thermal crossover. However, these improvements come at a high computational cost and would not currently be suitable for field robotic applications. Excelling in this benchmark would involve improving segmentation performance of the *object* classes, possibly by leveraging RGB data in unsupervised domain adaptation training schemes, as well as increasing computational efficiency through prudent architectural design choices.

RGB-Thermal Image Translation

We use the RGB-T paired dataset from Section 6.5 to benchmark image translation algorithms operating in the RGB→T direction.

Table 6.4: RGB-Thermal image translation results

Method	Method Type	RGB → Thermal		
		PSNR ↑	SSIM ↑	mIoU ↑
UNIT [41]	Unpaired GAN	12.85	0.421	0.189
MUNIT [27]	Unpaired GAN	16.08	0.459	0.173
Edge-guided RGB-T [35]	Unpaired GAN	12.20	0.436	0.225
Pix2Pix [28]	Paired GAN	**20.06**	0.547	0.354
Pix2PixHD [71]	Paired GAN	20.05	**0.579**	0.379
VQGAN [15]	Paired GAN	18.09	0.540	**0.388**
Palette [57]	Paired Diffusion	11.51	0.399	0.194

Baselines: We benchmark GAN methods that need RGB-T pairs (Pix2Pix [28], Pix2PixHD [71], VQGAN [15]) and unpaired methods (UNIT [41], MUNIT [27], [35]) that do not. We also evaluate Palette [57], a paired diffusion approach.

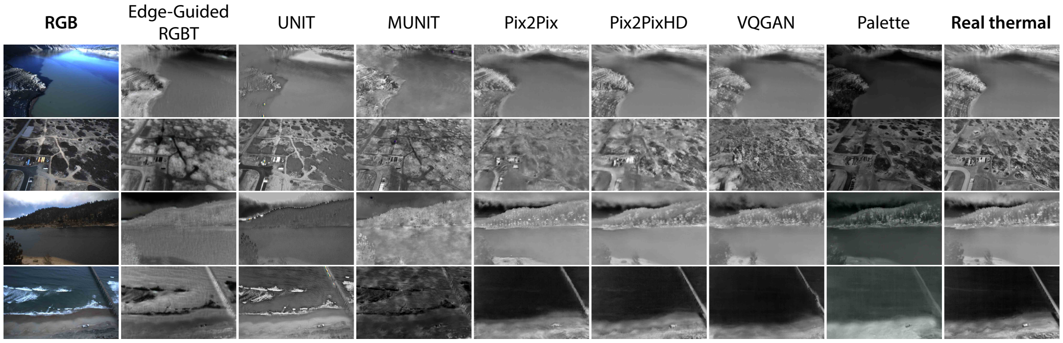


Figure 6.6: RGB-to-thermal image translation results. Zoom-in to see fine-details.

Image translation metrics: We quantify RGB-T image translation using the Peak-Signal-to-Noise ratio (PSNR) and the Structural Similarity Index (SSIM). In addition, we propose a third metric: the thermal mIoU. This is equivalent to the FCN-score used in RGB image translation works [28]. Instead of an RGB network, our metric uses our EfficientViT thermal segmentation network (Table 6.1) to segment translated thermal images before evaluating with ground truth.

Performance analysis: Paired GANs outperform unpaired GANs and diffusion methods, achieving higher values across all three metrics (Table 6.4). Qualitative assessment (Fig. 6.6) shows poor translations from unpaired techniques, with most retaining geometric artifacts unique to RGB (ocean waves and shadows) and ignoring relative temperature characteristics. Paired GANs produce results that appear similar to real thermal images but with inconsistent details upon closer inspection. Likewise, the diffusion method produces accurate translations but is not consistent. Overall, RGB-T translation still requires improvements before being able to provide a reliable source of thermal training data from existing RGB datasets. For domain adaptation purposes, we emphasize that maximizing the thermal mIoU score is more important than photometric consistency.

Motion Tracking

To quantify VIO/SLAM robustness, we select 12 clipped sequences from our dataset ranging in motion tracking difficulty: from urban sports fields (easy) to our natural environments (hard). We normalize thermal images using the 5th and 95th percentile pixel values (Section 6.4) to enable feature detection.

Table 6.5: VIO/SLAM performance (Absolute Trajectory Error [m]) on aerial sequences

Method	Type	Modality	North Field						Castaic Lake, CA				Duck, NC	
			N1	N2	N3	N4	N5	N6	C1	C2	C3	C4	D1	D2
VINS-Fusion	SLAM	RGB	2.883	7.530	4.036	2.009	4.045	0.834	1.555	—	3.277	1.566	1.031	0.700
VINS-Fusion		Thermal	9.454	12.26	9.854	2.926	11.00	5.114	7.296	1.513	5.321	2.025	0.879	—
VINS-Fusion	VIO	RGB	5.458	9.509	6.518	2.121	6.026	1.277	1.555	—	3.277	1.377	1.031	0.725
VINS-Fusion		Thermal	13.82	12.86	11.99	2.926	13.98	5.194	7.284	10.60	5.321	2.034	0.879	—
Open-VINS	VIO	RGB	22.80	36.15	37.15	5.165	—	3.896	—	3.573	—	—	0.475	0.700
Open-VINS		Thermal	14.43	30.64	16.12	2.258	—	1.562	—	5.311	—	—	1.073	—
Trajectory length (m)			1533	1326	1253	153	1104	642	310	65	174	137	88	80

Baselines: We evaluate VINS-Fusion [53] (graph optimization-based) and Open-VINS [21] (filtering-based). We test VINS-Fusion in VIO and SLAM modes, with and without loop closure constraints, respectively, and report Absolute Trajectory Error [62] averaged over four runs.

Performance analysis: This benchmark reveals challenges due to fast motion at altitude and periodic feature sparsity in littoral environments. In such scenes, motion tracking frequently failed due to loss of features, requiring sequences to be clipped for quantifiable evaluations (Table 6.5). VINS-Fusion outperformed Open-VINS mainly due to higher feature tracking reliability. VINS-Fusion works better on RGB images than on thermal images in general. However, VINS-Fusion demonstrates superior feature tracking on thermal images compared to RGB on the C2 sequence of Castaic Lake, where the thermal contrast of stone grains on the lake coast was particularly pronounced in the late afternoon. This observation motivates future research on integrating thermal and RGB cameras for motion tracking with enhanced robustness and versatility. Our benchmark presents a difficult challenge for VIO and SLAM algorithms in extreme environments where water and reflections off water surfaces dominate image scenes, providing a unique test bed to invigorate interest in robust feature matching and motion tracking in natural texture-deficient scenarios.

Further Analysis

Zero-shot foundation models on thermal imagery: Current foundation models work well on RGB imagery but have not been tested in the thermal domain, especially in non-urban environments. To evaluate foundation models on thermal imagery, specifically for zero-shot segmentation and semantic segmentation, we compare RGB and thermal performance using the aligned set (Section 6.4). With this set, we isolate modality as the root cause for any performance difference. In the following experiments, our metrics do not include the rare *object* classes (*vehicle* and *person*) as they drop metrics to near-uniform values for both modalities, making it difficult to compare.

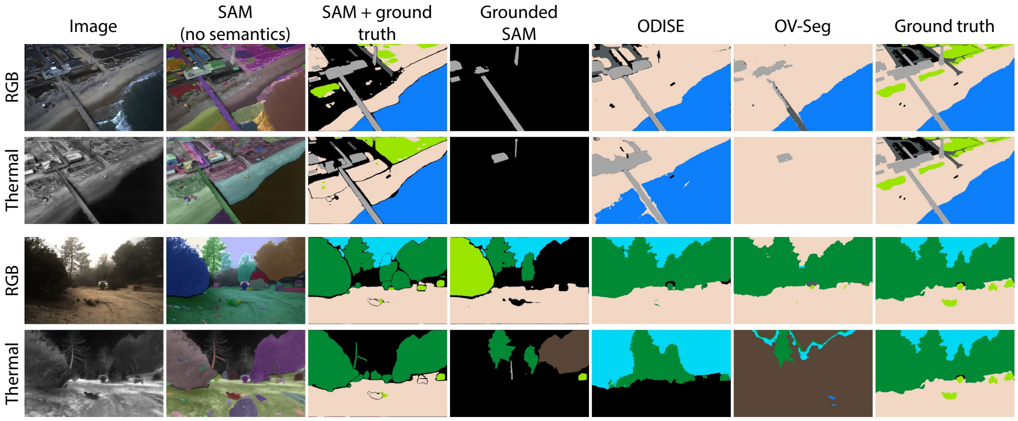


Figure 6.7: Zero-shot foundation models on RGB-T pairs. Oversegmented SAM outputs differ across modalities but are similar with ground truth semantics added. Semantic segmentation models (prompted with all classes) perform better on RGB imagery.

We quantify Segment Anything's (SAM) [33] zero-shot segmentation sensitivity to modality change by measuring the spatial alignment of its mask outputs. We apply it to the RGB set and the 8-bit thermal image set (Section 6.4) with the default 32×32 grid points to segment everything (Fig. 6.7). We compute the average precision (AP) of the predicted thermal masks, using the predicted RGB masks as ground truth, and compare the thermal AP against the APs of a color-jittered RGB set and a grayscale set. Results show that SAM is sensitive to photometric changes, with sizable performance drops from color-jitter to grayscale to thermal (Table 6.6a).

Next, we test large vision-language models (Grounded-SAM [42], OV-Seg [37], and ODISE [75]) in zero-shot semantic segmentation. We prompt the models with a list of classes in our dataset; generate semantic segmentation masks for RGB, color-jittered RGB, grayscale, and thermal image sets; and compute the

Table 6.6: RGB foundation model performances on thermal imagery

(a) **Instance** segmentation via SAM on thermal and augmented color images using coregistered color image segmentations as ground truth

Modality	$AP^{\dagger}_{all}$	AP_{small}	AP_{med}	AP_{large}
Thermal	0.018	0.007	0.022	0.204
Color (grayscale)	0.538	0.536	0.531	0.601
Color (col. jitter)	0.800	0.792	0.801	0.841

†refers to AP@0.5::0.95, following MSCOCO [39]

(b) Zero-shot **semantic** segmentation (mIoU) on coregistered color and thermal imagery

Method	**Color**	**Color (gray)**	**Color (jitter)**	**Thermal**
SAM [33] + GT ann.‡	0.704	0.698	0.700	0.653
Grounded SAM [42]	0.380	0.351	0.361	0.193
OV-Seg [37]	0.362	0.340	0.365	0.217
ODISE [75]	0.504	0.450	0.486	0.232

‡ Most frequent ground truth class label in a SAM mask assigned as label for entire mask

mIoU of their outputs using our ground truth annotations. Although RGB, RGB-variants, and thermal modalities all yield poor semantic segmentation performance (Table 6.6b), all RGB variants outperform thermal by at least 0.13 mIoU, indicating low generalization to the thermal modality (Fig. 6.7).

Finally, we create semantic SAM masks by assigning class labels to SAM instances based on the most frequent ground truth class within an instance (Fig. 6.7). We find that semantic SAM masks attain much higher mIoU score (0.653) compared to the other zero-shot methods. Based on the overall results (Table 6.6), we conclude that SAM can delineate object boundaries in the thermal domain reasonably well but can oversegment instances resulting in poor AP scores when compared to RGB (which can also oversegment). When paired with ground truth semantic labels, this behavior is hidden.

Transfer learning with urban thermal datasets: We test if pretraining on thermal urban datasets can assist transfer to field settings. We sample 40k unlabeled thermal images from FLIR ADAS [17], Freiburg [69], HIT UAV [65], KAIST Pedestrian [11], M^{3}FD [40], MFN [22], MS^2 [60], and SCUT-Seg [74] and pretrain networks of various sizes using MoCoV2 [9]. To assess their usefulness, we train

Table 6.7: Network pretraining methods for downstream thermal segmentation.

Model	mIoU			#
	None	ImageNet	Therm. Urban	Params. (M)
FastSCNN	0.640	0.690	0.688	1.1
EfficientViT	0.687	0.725	0.714	4.8
ResNet18	0.702	0.713	0.706	12.3
ResNet50	0.682	0.713	0.728	26.7
ConvNext-B	0.625	0.717	0.697	89.4

segmentation networks starting from the pretrained weights and evaluate on our test set. Overall, we find sparse evidence suggesting any advantage of pretraining with existing urban thermal datasets (Table 6.7). Instead, off-the-shelf ImageNet weights can provide strong performance in the thermal field domain, especially over training from scratch, without the effort and costs of pretraining.

6.6 Conclusion

We presented the Aerial RGB-Thermal Dataset, the first publicly available dataset specifically tailored towards advancing thermal semantic perception and motion tracking algorithms in natural environments. We created four distinct benchmarks encompassing semantic segmentation in both thermal and RGB-T domains, RGB-T image translation, and motion tracking, and use these benchmarks to demonstrate the current challenges faced by thermal-based robot perception and localization algorithms. Current semantic segmentation and image translation methods were particularly affected by geographical domain shifts, reflecting the diverse intra-class and inter-scenery distributions in our dataset, as well as temporal domain shifts due to thermal inversion and crossover. Motion tracking algorithms, on the other hand, suffered from poor feature tracking in the thermal modality due to reduced spatial quality as compared to RGB and encountered failures when faced with challenging, feature-sparse scenarios. However, such scenarios are commonly encountered during real-world deployments with failures of significant cost and consequence [1]. Our motion tracking benchmark greatly penalizes current VIO/SLAM algorithms for assuming ideal feature-tracking conditions, serving as a unique testbed for algorithm improvements in this area. This dataset, along with the benchmarks, can be used into the future to help develop algorithms that help expand the operating domains of both robotics and computer vision in general.

References

[1] E. Ackerman. *Blade strike on landing ends Mars helicopter's epic journey*. 2024.

[2] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart. "Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback". In: *The International Journal of Robotics Research* 36.10 (2017), pp. 1053–1072.

[3] E. Bondi, R. Jain, P. Aggrawal, S. Anand, R. Hannaford, A. Kapoor, J. Piavis, S. Shah, L. Joppa, B. Dilkina, et al. "BIRDSAI: A dataset for detection and tracking in aerial thermal infrared videos". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2020, pp. 1747–1756.

[4] K. L. Brodie, B. L. Bruder, R. K. Slocum, and N. J. Spore. "Simultaneous Mapping of Coastal Topography and Bathymetry From a Lightweight Multicamera UAS". In: *IEEE Transactions on Geoscience and Remote Sensing* 57.9 (2019), pp. 6844–6864. doi: 10.1109/TGRS.2019.2909026.

[5] H. Cai, C. Gan, and S. Han. "Efficientvit: Enhanced linear attention for high-resolution low-computation visual recognition". In: *arXiv preprint arXiv:2205.14756* (2022).

[6] W. Cai, K. Jin, J. Hou, C. Guo, L. Wu, and W. Yang. "VDD: Varied Drone Dataset for Semantic Segmentation". In: *arXiv preprint arXiv:2305.13608* (2023).

[7] A. F. S. Center. *A Dataset for Machine Learning Algorithm Development from 2010-06-15 to 2010-08-15*. NOAA National Centers for Environmental Information. 2019.

[8] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. "Encoder-decoder with atrous separable convolution for semantic image segmentation". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 801–818.

[9] X. Chen, H. Fan, R. Girshick, and K. He. "Improved Baselines with Momentum Contrastive Learning". In: *arXiv preprint arXiv:2003.04297* (2020).

[10] Y. Chen, Y. Wang, P. Lu, Y. Chen, and G. Wang. "Large-scale structure from motion with semantic constraints of aerial images". In: *Pattern Recognition and Computer Vision: First Chinese Conference, PRCV 2018, Guangzhou, China, November 23-26, 2018, Proceedings, Part I 1*. Springer. 2018, pp. 347–359.

[11] Y. Choi, N. Kim, S. Hwang, K. Park, J. S. Yoon, K. An, and I. S. Kweon. "KAIST multi-spectral day/night data set for autonomous and assisted driving". In: *IEEE Transactions on Intelligent Transportation Systems* 19.3 (2018), pp. 934–948.

[12] S. A. Deevi, C. Lee, L. Gan, S. Nagesh, G. Pandey, and S.-J. Chung. “RGB-X Object Detection via Scene-Specific Fusion Modules”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2024, pp. 7366–7375.

[13] J. Delaune, R. Hewitt, L. Lytle, C. Sorice, R. Thakker, and L. Matthies. “Thermal-inertial odometry for autonomous flight throughout the night”. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2019, pp. 1122–1128.

[14] C. Doer and G. F. Trommer. “Radar visual inertial odometry and radar thermal inertial odometry: Robust navigation even in challenging visual conditions”. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2021, pp. 331–338.

[15] P. Esser, R. Rombach, and B. Ommer. “Taming transformers for high-resolution image synthesis”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 12873–12883.

[16] D. Feng, C. Haase-Schütz, L. Rosenbaum, H. Hertlein, C. Glaeser, F. Timm, W. Wiesbeck, and K. Dietmayer. “Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges”. In: *IEEE Transactions on Intelligent Transportation Systems* 22.3 (2020), pp. 1341–1360.

[17] *Teledyne FLIR ADAS Dataset*. https://www.flir.com/oem/adas/adas-dataset-form/, Last accessed on 2023-10-27.

[18] S. Fountas, N. Mylonas, I. Malounas, E. Rodias, C. Hellmann Santos, and E. Pekkeriet. “Agricultural robotics for field operations”. In: *Sensors* 20.9 (2020), p. 2672.

[19] R. Gade and T. B. Moeslund. “Thermal cameras and applications: a survey”. In: *Machine vision and applications* 25 (2014), pp. 245–262.

[20] L. Gan, C. Lee, and S.-J. Chung. “Unsupervised rgb-to-thermal domain adaptation via multi-domain attention network”. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE. 2023, pp. 6014–6020.

[21] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang. “Openvins: A research platform for visual-inertial estimation”. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 4666–4672.

[22] Q. Ha, K. Watanabe, T. Karasawa, Y. Ushiku, and T. Harada. “MFNet: Towards real-time semantic segmentation for autonomous vehicles with multi-spectral scenes”. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2017, pp. 5108–5115.

[23] K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 770–778.

[24] M. He and R. R. Rajkumar. "Using Thermal Vision for Extended VINS-Mono to Localize Vehicles in Large-Scale Outdoor Road Environments". In: *2021 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2021, pp. 953–960.

[25] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, et al. "Searching for mobilenetv3". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 1314–1324.

[26] T. Hua, L. Pei, T. Li, Q. Wu, R. Wang, and W. Yu. "I2-SLAM: Fusing Infrared Camera and IMU for Simultaneous Localization and Mapping". In: *International Conference on Autonomous Unmanned Systems*. Springer. 2021, pp. 2834–2844.

[27] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz. "Multimodal Unsupervised Image-to-image Translation". In: *European Conference on Computer Vision (ECCV)*. 2018.

[28] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. "Image-to-Image Translation with Conditional Adversarial Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017.

[29] X. Jia, C. Zhu, M. Li, W. Tang, and W. Zhou. "LLVIP: A visible-infrared paired dataset for low-light vision". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 3496–3504.

[30] A. Jong, M. Yu, D. Dhrafani, S. Kailas, B. Moon, K. Sycara, and S. Scherer. "WIT-UAS: A Wildland-fire Infrared Thermal Dataset to Detect Crew Assets From Aerial Views". In: *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2023, pp. 11464–11471.

[31] S. Khattak, C. Papachristos, and K. Alexis. "Keyframe-based direct thermal–inertial odometry". In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 3563–3569.

[32] Y.-H. Kim, U. Shin, J. Park, and I. S. Kweon. "MS-UDA: Multi-Spectral Unsupervised Domain Adaptation for Thermal Image Semantic Segmentation". In: *IEEE Robotics and Automation Letters* 6.4 (2021), pp. 6497–6504. DOI: 10.1109/LRA.2021.3093652.

[33] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick. "Segment Anything". In: *arXiv:2304.02643* (2023).

[34] C. Lee, J. G. Frennert, L. Gan, M. Anderson, and S.-J. Chung. "Online Self-Supervised Thermal Water Segmentation for Aerial Vehicles". In: *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2023, pp. 7734–7741.

[35] D.-.-G. Lee, M.-.-H. Jeon, Y. Cho, and A. Kim. "Edge-guided multi-domain rgb-to-tir image translation for training vision tasks with challenging labels". In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2023, pp. 8291–8298.

[36] C. Li, W. Xia, Y. Yan, B. Luo, and J. Tang. "Segmenting objects in day and night: Edge-conditioned cnn for thermal image semantic segmentation". In: *IEEE Transactions on Neural Networks and Learning Systems* 32.7 (2020), pp. 3069–3082.

[37] F. Liang, B. Wu, X. Dai, K. Li, Y. Zhao, H. Zhang, P. Zhang, P. Vajda, and D. Marculescu. "Open-vocabulary semantic segmentation with mask-adapted clip". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 7061–7070.

[38] M. Liang, J. Hu, C. Bao, H. Feng, F. Deng, and T. L. Lam. "Explicit Attention-Enhanced Fusion for RGB-Thermal Perception Tasks". In: *IEEE Robotics and Automation Letters* (2023).

[39] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. "Microsoft COCO: Common objects in context". In: *Proceedings of the European Conference on Computer Vision*. Springer. 2014, pp. 740–755.

[40] J. Liu, X. Fan, Z. Huang, G. Wu, R. Liu, W. Zhong, and Z. Luo. "Target-aware Dual Adversarial Learning and a Multi-scenario Multi-Modality Benchmark to Fuse Infrared and Visible for Object Detection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2022, pp. 5802–5811.

[41] M.-Y. Liu, T. Breuel, and J. Kautz. "Unsupervised Image-to-Image Translation Networks". In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2017.

[42] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, et al. "Grounding dino: Marrying dino with grounded pre-training for open-set object detection". In: *arXiv preprint arXiv:2303.05499* (2023).

[43] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie. "A convnet for the 2020s". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 11976–11986.

[44] Y. Lyu, G. Vosselman, G.-S. Xia, A. Yilmaz, and M. Y. Yang. "UAVid: A semantic segmentation dataset for UAV imagery". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 165 (2020), pp. 108–119. ISSN: 0924-2716. DOI: https://doi.org/10.1016/j.isprsjprs.2020.05.009.

[45] S. Mehta and M. Rastegari. "Separable self-attention for mobile vision transformers". In: *arXiv preprint arXiv:2206.02680* (2022).

[46] C. Mostegel, M. Maurer, N. Heran, J. Pestana Puerta, and F. Fraundorfer. *Semantic Drone Dataset*. http://dronedataset.icg.tugraz.at/, Last accessed on 2023-10-27. 2019.

[47] I. Nigam, C. Huang, and D. Ramanan. "Ensemble knowledge transfer for semantic segmentation". In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2018, pp. 1499–1508.

[48] S. Nirgudkar, M. DeFilippo, M. Sacarny, M. Benjamin, and P. Robinette. "MassMIND: Massachusetts Maritime INfrared Dataset". In: *International Journal of Robotics Research* 42.1-2 (2023), pp. 21–32.

[49] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, R. Howes, P.-Y. Huang, H. Xu, V. Sharma, S.-W. Li, W. Galuba, M. Rabbat, M. Assran, N. Ballas, G. Synnaeve, I. Misra, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski. *DINOv2: Learning Robust Visual Features without Supervision*. 2023.

[50] K. Panetta, K. M. Shreyas Kamath, S. Rajeev, and S. S. Agaian. "FTNet: Feature Transverse Network for Thermal Image Semantic Segmentation". In: *IEEE Access* 9 (2021), pp. 145212–145227. DOI: 10.1109/ACCESS.2021.3123066.

[51] T. Park, A. A. Efros, R. Zhang, and J.-Y. Zhu. "Contrastive learning for unpaired image-to-image translation". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*. Springer. 2020, pp. 319–345.

[52] R. P. Poudel, S. Liwicki, and R. Cipolla. "Fast-scnn: Fast semantic segmentation network". In: *arXiv preprint arXiv:1902.04502* (2019).

[53] T. Qin, P. Li, and S. Shen. "Vins-mono: A robust and versatile monocular visual-inertial state estimator". In: *IEEE Transactions on Robotics* 34.4 (2018), pp. 1004–1020.

[54] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. "Learning transferable visual models from natural language supervision". In: *International conference on machine learning*. PMLR. 2021, pp. 8748–8763.

[55] S. P. Retief, C. Willers, and M. Wheeler. "Prediction of thermal crossover based on imaging measurements over the diurnal cycle". In: *Geo-Spatial and Temporal Image and Data Exploitation III*. Vol. 5097. SPIE. 2003, pp. 58–69.

[56] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. "High-Resolution Image Synthesis With Latent Diffusion Models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 10684–10695.

[57] C. Saharia, W. Chan, H. Chang, C. Lee, J. Ho, T. Salimans, D. Fleet, and M. Norouzi. "Palette: Image-to-image diffusion models". In: *ACM SIGGRAPH 2022 Conference Proceedings*. 2022, pp. 1–10.

[58] M. R. U. Saputra, C. X. Lu, P. P. B. de Gusmao, B. Wang, A. Markham, and N. Trigoni. "Graph-based thermal–inertial SLAM with probabilistic neural networks". In: *IEEE Transactions on Robotics* 38.3 (2022), pp. 1875–1893.

[59] U. Shin, K. Lee, and I. S. Kweon. "Complementary Random Masking for RGB-Thermal Semantic Segmentation". In: *IEEE International Conference on Robotics and Automation*. 2024.

[60] U. Shin, J. Park, and I. S. Kweon. "Deep Depth Estimation From Thermal Image". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 1043–1053.

[61] S. S. Shivakumar, N. Rodrigues, A. Zhou, I. D. Miller, V. Kumar, and C. J. Taylor. "Pst900: Rgb-thermal calibration, dataset and segmentation network". In: *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2020, pp. 9441–9447.

[62] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. "A benchmark for the evaluation of RGB-D SLAM systems". In: *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE. 2012, pp. 573–580.

[63] Y. Sun, W. Zuo, and M. Liu. "RTFNet: RGB-Thermal Fusion Network for Semantic Segmentation of Urban Scenes". In: *IEEE Robotics and Automation Letters* 4.3 (2019), pp. 2576–2583. DOI: 10.1109/LRA.2019.2904733.

[64] Y. Sun, W. Zuo, P. Yun, H. Wang, and M. Liu. "FuseSeg: Semantic Segmentation of Urban Scenes Based on RGB and Thermal Data Fusion". In: *IEEE Transactions on Automation Science and Engineering* 18.3 (2021), pp. 1000–1011. DOI: 10.1109/TASE.2020.2993143.

[65] J. Suo, T. Wang, X. Zhang, H. Chen, W. Zhou, and W. Shi. "HIT-UAV: A high-altitude infrared thermal dataset for Unmanned Aerial Vehicle-based object detection". In: *Scientific Data* 10 (2023), p. 227.

[66] M. Tan and Q. Le. "Efficientnet: Rethinking model scaling for convolutional neural networks". In: *International conference on machine learning*. PMLR. 2019, pp. 6105–6114.

[67] B. Ustun, A. K. Kaya, E. C. Ayerden, and F. Altinel. "Spectral Transfer Guided Active Domain Adaptation for Thermal Imagery". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 2023, pp. 449–458.

[68] J. Vargas, S. Alsweiss, O. Toker, R. Razdan, and J. Santos. "An overview of autonomous vehicles sensors and their vulnerability to weather conditions". In: *Sensors* 21.16 (2021), p. 5397.

[69] J. Vertens, J. Zürn, and W. Burgard. “HeatNet: Bridging the day-night domain gap in semantic segmentation with thermal images”. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2020, pp. 8461–8468.

[70] V. VS, D. Poster, S. You, S. Hu, and V. M. Patel. “Meta-UDA: Unsupervised Domain Adaptive Thermal Object Detection Using Meta-Learning”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2022, pp. 1412–1423.

[71] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. “High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.

[72] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo. “SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers”. In: *Neural Information Processing Systems (NeurIPS)*. 2021.

[73] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. “Aggregated residual transformations for deep neural networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1492–1500.

[74] H. Xiong, W. Cai, and Q. Liu. “MCNet: Multi-level Correction Network for thermal image semantic segmentation of nighttime driving scene”. In: *Infrared Physics & Technology* (2021), p. 103628. ISSN: 1350-4495. DOI: https://doi.org/10.1016/j.infrared.2020.103628.

[75] J. Xu, S. Liu, A. Vahdat, W. Byeon, X. Wang, and S. De Mello. “Open-vocabulary panoptic segmentation with text-to-image diffusion models”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 2955–2966.

[76] S. Yun, M. Jung, J. Kim, S. Jung, Y. Cho, M.-H. Jeon, G. Kim, and A. Kim. “Sthereo: Stereo thermal dataset for research in odometry and mapping”. In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2022, pp. 3857–3864.

[77] J. Zhang, R. Liu, H. Shi, K. Yang, S. Reiß, K. Peng, H. Fu, K. Wang, and R. Stiefelhagen. “Delivering Arbitrary-Modal Semantic Segmentation”. In: *CVPR*. 2023.

[78] S. Zhao, P. Wang, H. Zhang, Z. Fang, and S. Scherer. “Tp-tio: A robust thermal-inertial odometry with deep thermalpoint”. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 4505–4512.

[79] W. Zhou, J. Liu, J. Lei, L. Yu, and J.-N. Hwang. “GMNet: Graded-Feature Multilabel-Learning Network for RGB-Thermal Urban Scene Semantic Segmentation”. In: *IEEE Transactions on Image Processing* 30 (2021), pp. 7790–7802. DOI: 10.1109/TIP.2021.3109518.

[80] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. “Unpaired image-to-image translation using cycle-consistent adversarial networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 2223–2232.

Chapter 7

SEMANTICS FROM SPACE: SATELLITE-GUIDED THERMAL SEMANTIC SEGMENTATION ANNOTATION FOR AERIAL FIELD ROBOTS

[1] C. Lee, S. Soedarmadji, M. Anderson, A. Clark, and S.-J. Chung. “Semantics from Space: Satellite-Guided Thermal Semantic Segmentation Annotation for Aerial Field Robots”. In: *arXiv preprint arXiv:2403.08997* (2024). Available at https://arxiv.org/abs/2403.14056.

7.1 Abstract

We present a new method to automatically generate semantic segmentation annotations for thermal imagery captured from an aerial vehicle by utilizing satellite-derived data products alongside onboard global positioning and attitude estimates. This new capability overcomes the challenge of developing thermal semantic perception algorithms for field robots due to the lack of annotated thermal field datasets and the time and costs of manual annotation, enabling precise and rapid annotation of thermal data from field collection efforts at a massively-parallelizable scale. By incorporating a thermal-conditioned refinement step with visual foundation models, our approach can produce highly-precise semantic segmentation labels using low-resolution satellite land cover data for little-to-no cost. It achieves 98.5% of the performance from using costly high-resolution options and demonstrates between 70-160% improvement over popular zero-shot semantic segmentation methods based on large vision-language models currently used for generating annotations for RGB imagery.

7.2 Introduction

Uninhabited Aerial Vehicles (UAVs) have been extensively used in field robotic applications, including precision agriculture [36], wildlife conservation [6], coastal mapping [8], and wildfire management [19]. To enable operations during nighttime and adverse weather conditions, UAVs can be equipped with long-wave thermal infrared cameras [13, 25] that provide dense scene perception in such settings. However, developing thermal scene perception for aerial field robotics requires ample data in order to train deep learning models for semantic segmentation [34].

This poses a challenge due to the scarcity of in-domain thermal data capturing typical aerial field robotic operational areas such as deserts [30], forests [19], and coastlines [35, 8].

Although several thermal semantic segmentation datasets of urban scenes have been curated for autonomous driving applications [27, 17, 42], few datasets exist that specifically target natural environments from an aerial viewpoint [26, 25]. To compensate for limited thermal data, existing works leverage large, annotated RGB datasets via domain adaptation techniques like image translation [27] and domain confusion [16, 22], as well as online learning [25] for thermal test-time adaptation. Despite reducing reliance on thermal training data, such methods still require annotated thermal data for comprehensive evaluation and robustness testing. While thermal datasets exist for field environments, most lack annotations relevant for aerial semantic segmentation [19, 35, 39] besides [26]. As a result, collecting and annotating thermal datasets for semantic segmentation is still necessary to further improve thermal scene perception results via supervised training.

Capturing and annotating thermal field data presents unique challenges. Unlike in RGB, publicly-available thermal imagery is scarce due to the high costs and specialized nature of thermal sensors. Consequently, relevant thermal imagery cannot be scraped from the web and field roboticists must travel to various locations for data collection. This process incurs significant time and financial expenses, as it requires extensive travel and permits for flying and data capture. Moreover, annotating thermal imagery adds further costs and delays due to its distinct visual characteristics. This requires multiple rounds of attentive expert review and re-annotation [26], and adds more time to the curation process.

In this study, we propose a method to significantly reduce the time and cost of annotating aerial thermal field imagery for semantic segmentation. We contribute the following:

1. An algorithm that automatically generates high-quality segmentation labels for aerial thermal imagery using estimated camera pose and satellite-derived data.

2. Experiments comparing segmentation labels generated from various satellite-derived data products, demonstrating competitive results with free options.

3. Extensive ablation studies showcasing the robustness of our method to noisy camera pose estimation and temporal misalignments between thermal and satellite

imagery.

4. A demonstration for aerial field robotics perception by training a semantic segmentation network solely on labels generated using our method, yielding promising results.

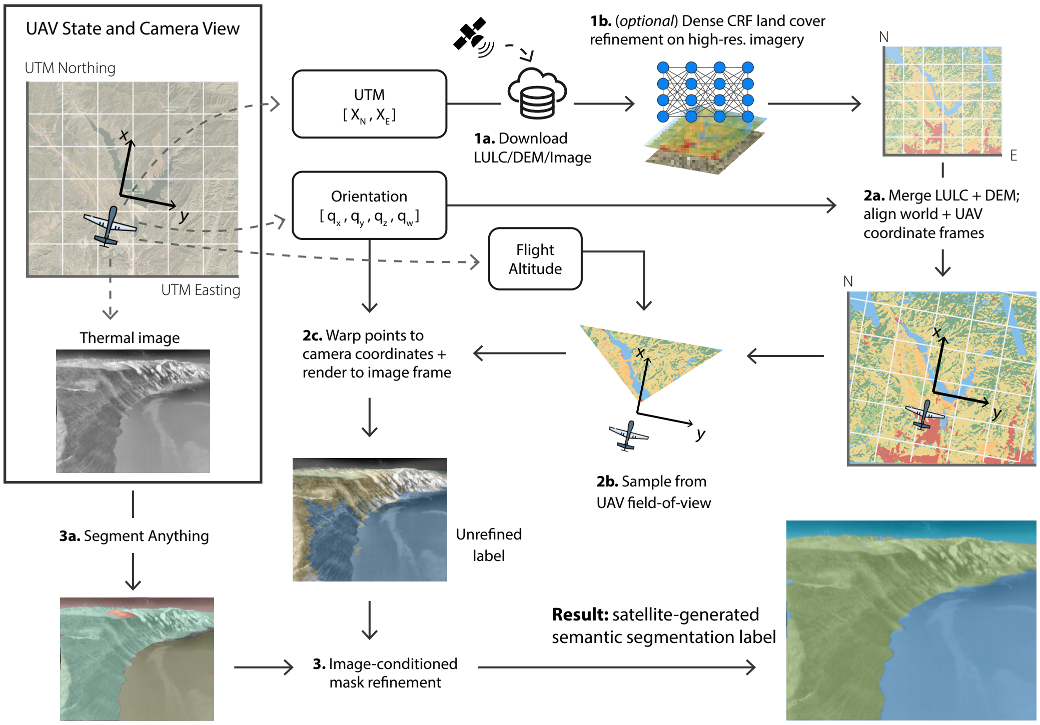

UAV State and Camera View
UTM Northing
x
y
UTM Easting
Thermal image
UTM
[X_N , X_E]
Orientation
[q_x , q_y , q_z , q_w]
Flight Altitude
1a. Download LULC/DEM/Image
1b. (optional) Dense CRF land cover refinement on high-res. imagery
N
E
2a. Merge LULC + DEM; align world + UAV coordinate frames
N
x
y
E
2b. Sample from UAV field-of-view
x
y
2c. Warp points to camera coordinates + render to image frame
Unrefined label
3a. Segment Anything
3. Image-conditioned mask refinement
Result: satellite-generated semantic segmentation label

Figure 7.1: Proposed pipeline for automatically generating semantic segmentation annotations from satellite-derived data. Coarse segmentation labels for thermal images are rendered from Land Use and Land Cover (LULC) datasets and Digital Elevation Maps (DEM). The labels are refined using Segment Anything [23] to capture fine details between segmentation instances.

7.3 Related Work

Semantic Segmentation: Semantic segmentation models perform per-pixel classification and are typically built upon convolutional neural networks and transformer architectures [11, 18, 29]. While conventional fully-supervised models achieve impressive results, they need large annotated training datasets for generalization. In applications like thermal semantic segmentation where labeled data is scarce, unsupervised domain adaptation (UDA) techniques are often employed. UDA methods like [27] synthesize labeled thermal training data from existing RGB datasets via image translation, while other works [16, 22] leverage RGB training for thermal inference by maximizing RGB-thermal domain confusion during training. However, UDA methods still face challenges: they require significant target domain data and thermal annotations for evaluation, and typically exhibit lower performance compared to fully-supervised methods [32].

Alternatively, recent large vision-language models like ODISE [45] and OV-Seg [28] can perform zero-shot semantic segmentation across the RGB spectrum by leveraging user-provided text prompts. Similarly, the Segment Anything Model [23] (SAM) can provide precise segmentations for any object but lacks semantic information. In general, the zero-shot semantic segmentation methods perform worse on thermal imagery compared to RGB [26]. Despite this, [26] finds that SAM can perform well in a semantic segmentation task if its segmentation outputs are assigned ground truth class labels. We leverage this finding in our approach.

Automatic Semantic Segmentation Annotation: Most works using automatic semantic segmentation labeling can be found in self-training and self-supervised learning literature. However, many focus on specialized applications with niche classes [25, 12] and are not relevant for general scene segmentation. For generalized semantic segmentation tasks, [3] self-trains their model using noisy labels predicted by their network for intra-RGB domain adaptation. In contrast, [46] adopts an incremental training approach and utilizes humans to select good network outputs as annotations and manually correct bad ones before retraining. Other works manually annotate a subset of frames in video data, before propagating them to remaining frames using optical flow [31] or learned generative models [4].

As discussed, visual foundation models can also be used for annotation efforts. Zero-shot semantic segmentation models [45, 28, 38] are being used to provide labels, but do not transfer directly to non-RGB domains. [15] generates object detection labels for thermal imagery by using SAM on aligned RGB images and does not work in low-light settings.

In contrast to other works, [7] uses 3D information to generate semantic segmentation labels for construction sites and is most similar to our work. They register Building Information Models with point clouds from photogrammetry and render the labeled 3D points to an image frame. Unlike other works, methods like this operate independently of an image and can work for any imaging modality.

7.4 Preliminaries

In this section, we briefly go over the different satellite-derived data products we use in our approach (Sec. 7.5).

Land Use and Land Cover Datasets: Publicly-available Land Use and Land Cover (LULC) datasets like Dynamic World [9] and Impact Observatory [21] derive from satellite rasters obtained through the Sentinel-2 program. These datasets have a low spatial resolution of 10 m/pixel but have global coverage, and are updated using semantic segmentation networks that use multiple data bands for landcover classification. While daily coverage is possible, availability depends on factors like cloud coverage.

In contrast, high-resolution LULC like the Chesapeake Bay Program [37] and OpenEarthMap [44] offer sub-meter resolution but are limited in geographical and temporal coverage. While segmentation models can be trained on these datasets with high-resolution imagery, they may not generalize to different geographical areas.

High-Resolution Raster Imagery: These include imagery from aerial vehicles and satellites. Aerial imagery providers include the National Agricultural Imagery Program (NAIP) [40] while satellite imagery comes from providers like Planet, Maxar, and Airbus. Image resolutions range from 0.3 m/pixel to 3 m/pixel. Imagery can be available daily at a premium cost while free alternatives are captured triannually.

Lidar-Derived 3D Data Products: Digital surface (DSM) and digital elevation models (DEM) are raster data whose values denote the height at the corresponding geographic location. DSMs consider features above the ground like foliage and rocky terrain while DEMs report bare earth elevation. In this work, we use DEMs and DSMs with 1 m/pixel to 3 m/pixel resolution from the 3D Elevation Program (3DEP) from the United States Geological Survey [41] .

7.5 Approach

We present a three-step method to automatically generate semantic segmentation annotations for thermal images captured from an aerial vehicle using satellite-derived data (Fig. 7.1).

Step 1: Generating 3D Semantic Maps from Satellite Data

We start by downloading relevant satellite data (LULC rasters, DEM or DSM, and high-resolution imagery) around the aerial vehicle's global position and resample them to the highest resolution via bicubic interpolation. To simplify future calculations, we convert to UTM coordinates before merging the DEM and LULC rasters. Since current freely-available LULC data is low resolution (10 m), we optionally refine them by conditioning on high resolution imagery as described below. Alternatively, high-resolution LULC can also be created using a pretrained LULC segmentation network on high resolution imagery (see Sec. 7.7).

Land-Use-Land-Cover Refinement: We use dense conditional random fields [24] (CRF) to refine 10 m resolution LULC rasters with 1 m-3 m resolution aerial imagery (Fig. 7.2). To summarize, a dense CRF is defined by a Boltzmann distribution with energy function

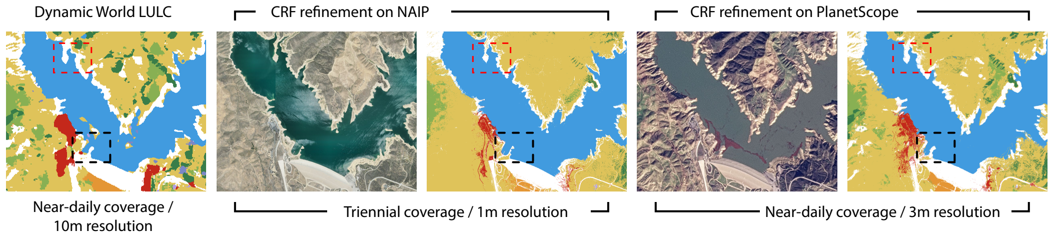

Dynamic World LULC
CRF refinement on NAIP
CRF refinement on PlanetScope
Near-daily coverage / 10m resolution
Triennial coverage / 1m resolution
Near-daily coverage / 3m resolution

Figure 7.2: Dense CRF refinement of Dynamic World land cover raster using NAIP and PlanetScope imagery of Castaic Lake, CA. Results via PlanetScope convey the actual scenery at time of thermal image capture due to its high revisit frequency but at a lower 3 m spatial resolution. NAIP refinement offers 1 m resolution but is susceptible to changes in the terrain (notably, water levels of lakes) due to its triennial capture cycle. Zoom in to see key differences (outlined in dashed boxes).

$$E(\mathbf{X}|\mathbf{I}) = \sum_i \psi_u(x_i|I_i) + \sum_{i<j} \psi_p(x_i, x_j|I_i). \tag{7.1}$$

This function models the relationship between labels $\mathbf{x} \in \mathbf{X}$ and the conditioning image $I \in \mathbb{R}^{H\times W\times C}$. Here, ψ_u is a unary potential taken to be raw logits from a semantic segmentation network and ψ_p is a pairwise potential that encourages label consistency among adjacent pixels with similar intensities.

In our method, we set ψ_u to be the logits from the model that generated our LULC labels. Like [20], we use a generalized ψ_p to condition on multi-band raster images:

$$\begin{aligned}\psi_p(\mathbf{f_i}, \mathbf{f_j}) = \mu \cdot \Bigg[& w^{(1)} \exp\left(-\frac{1}{2}\bar{\mathbf{p}}_{ij}^\top \boldsymbol{\Sigma}_\alpha \bar{\mathbf{p}}_{ij} - \frac{1}{2}\bar{\mathbf{I}}_{ij}^\top \boldsymbol{\Sigma}_\beta \bar{\mathbf{I}}_{ij}\right) \\ & + w^{(2)} \exp\left(-\frac{1}{2}\bar{\mathbf{p}}_{ij}^\top \boldsymbol{\Sigma}_\gamma \bar{\mathbf{p}}_{ij}\right)\Bigg].\end{aligned} \tag{7.2}$$

Here, $\bar{\mathbf{p}}_{ij} = [\mathbf{p}_i - \mathbf{p}_j] \in \mathbb{R}^3$ is the difference between positions of pixel i and j, and $\bar{\mathbf{I}}_{ij} = [\mathbf{I}_i - \mathbf{I}_j] \in \mathbb{R}^C$ is the difference between image features at pixels i and j. We set μ as the standard Potts compatibility function [24].

We optimize the CRF by tuning weight parameters $w^{(1)}$ and $w^{(2)}$, and the Gaussian bandwidth parameters $\mathbf{\Sigma}_\alpha = \theta_\alpha$, $\mathbf{\Sigma}_\gamma = \theta_\gamma$, and $\mathbf{\Sigma}_\beta = \operatorname{diag}\big(\theta_\beta^{(1)}, ..., \theta_\beta^{(C)}\big)$. We minimize the boundary loss [5] which is the complement of the F1 score with precision P and recall R where

$$P = \frac{\sum b(\hat{\mathbf{y}}) \odot b^{\text{ex}}(\mathbf{y})}{\sum b(\hat{\mathbf{y}})} \qquad R = \frac{\sum b(\mathbf{y}) \odot b^{\text{ex}}(\hat{\mathbf{y}})}{\sum b(\mathbf{y})} \tag{7.3}$$

and

$$b(x) = \mathbf{Pool}^{\mathbf{3\times3}}_{\text{max}}[1 - x] - (1 - x) \tag{7.4}$$

$$b^{\text{ex}}(x) = \mathbf{Pool}^{\mathbf{5\times5}}_{\text{max}}\big[b(x)\big]. \tag{7.5}$$

We use this instead of cross entropy to account for imprecise labels at class boundaries due to low LULC resolution.

Step 2: LULC Projection to Aerial Camera Image Frame

To generate an LULC-derived semantic label for an image at time t, we start by transforming the world coordinates of each pixel $\mathbf{X}_w \in \mathbb{R}^3$ into the camera coordinate frame. This requires the position of the host vehicle $\mathbf{x}_t^{\text{uav}} \in \mathbb{R}^3$, taken from the onboard EKF-fused GPS position and barometric altitude, the orientation quaternion $\mathbf{q}_t \in \mathbb{H}$, taken from the EKF-fused IMU readings, and the offset between the aircraft and camera reference points. Using the calibrated camera intrinsic matrix $\mathbf{K}$, we can then project to image coordinates $\mathbf{x}_t^c \in \mathbb{Z}^2$. Formally, this is

$$\begin{bmatrix} \mathbf{x}_t^c \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} \mathbf{R}(\mathbf{q}_t) & \mathbf{T}(\mathbf{x}_t^{\text{uav}}) \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{X}_w \\ 1 \end{bmatrix} \tag{7.6}$$

where $\mathbf{R}$ is a rotation matrix and $\mathbf{T}$ is a translation vector. We use OpenGL to render the projected LULC, using 3D coordinates as vertices, associated class labels as vertex colors, and depth-testing to avoid rendering occluded semantics.

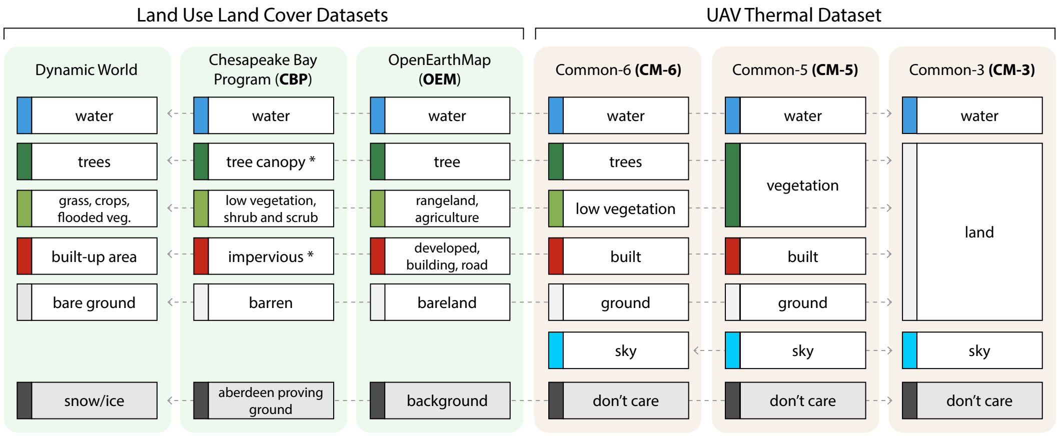
To optimize memory and speed, we only consider 3D semantics within a specified distance in front of and on both sides of the camera. We also exponentially increase spacing between sampled vertices as distance from the camera increases, exploiting the compression of far-field points in the image frame. This enables us to use only 250 × 200 points when rendering within a 10 km×8 km bounding box.

Algorithm 3 SAM-based Label Refinement

```
Input: Projected (unrefined) label mask $L \in \mathbb{N}^{H\times W}$,
       Thermal image $I \in \mathbb{R}^{H\times W}$
Output: Refined semantic segmentation label $M$
Initialize: Segment Anything Model $f_{\text{sam}}$

$\{M^i_{\text{sam}}\}^N_0 \leftarrow f_{\text{sam}}(I)$                ▷ SAM produces binary masks
Initialize zero-array $M$ of size $H \times W$

for $m_{\text{sam}} \in \{M^i_{\text{sam}}\}^N_0$ do
    $x_{idx} \leftarrow [m_{\text{sam}} == 1]$                          ▷ Get mask indices
    $y_{\text{cls}} \leftarrow L[x_{idx}]$.mode()                      ▷ Find most freq. class in mask
    $M[x_{idx}] = y_{\text{cls}}$
end for

return $M$
```

Step 3: Rendered Label Refinement

Though the semantic segmentation labels have been rendered, they do not align well with the thermal images. This is primarily caused by poor spatial resolution and temporal misalignment, but could also stem from errors in LULC label generation and camera pose estimation. To improve alignment, we refine the labels by generating binary segmentation masks of the corresponding thermal image using the Segment Anything Model. Then, we assign each mask a semantic class based on the most prevalent LULC class within it (Alg. 3).



* indicates other variants of this class: *impervious structures, impervious roads, tree canopy above road, etc...*

Figure 7.3: Class mappings between LULC datasets and our ground truth evaluation set. The UAV thermal dataset was created in Chapter 6.

7.6 Low Altitude Aerial Dataset

We test our method using a thermal field robotics dataset, which includes off-nadir (20°-45°) aerial views of rivers (Kentucky River, KY and Colorado River, CA), lakes

(Castaic Lake, CA), and coastal (Duck, NC) areas across the United States [25, 26]. The dataset, captured from a multirotor, comprises 15 flight trajectories ranging from 40 m to 100 m in altitude and contains time-synchronized thermal imagery, GPS, and IMU measurements. Four trajectories are excluded from testing due to GPS data collection errors. While the dataset provides ground truth semantic segmentation annotations for 10 classes, we condense the classes into 6 categories in order to better conform with land cover classes. We end up with ground truth, 6-class semantic segmentation labels for 1304 sub-sampled images (CM-6) and further condense the classes again to create two additional class-sets, CM-5 (5 classes) and CM-3 (3 classes). A mapping of segmentation labels is shown in Fig. 7.3.

7.7 Results

We outline our experimental setup, including data acquisition and generation, and specific method parameters, before presenting our results.

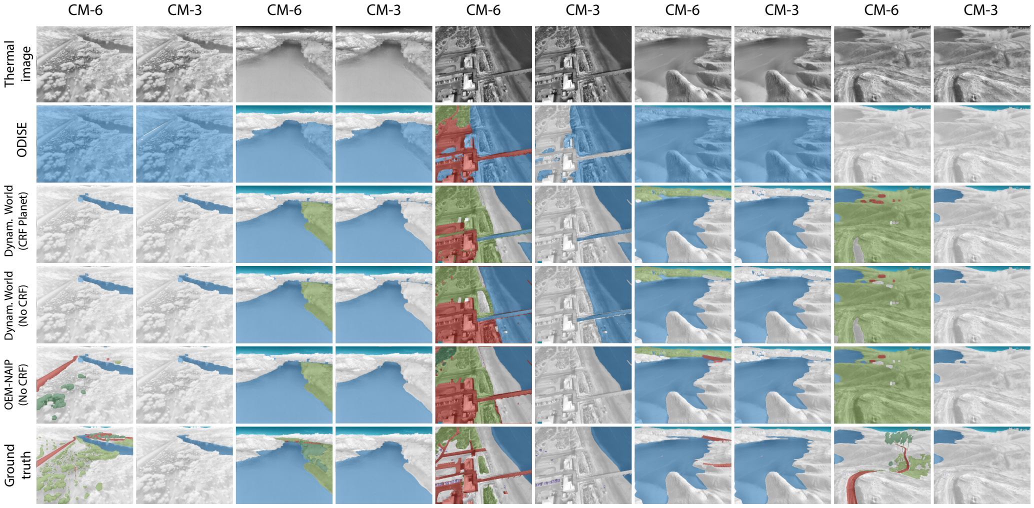


Figure 7.4: Generated segmentations from the baseline (ODISE [45]), our methods, and the ground truth (GT) using class mappings and colors from Fig. 7.3. Mismatches between CM-6 labels and GT can occur depending on the LULC source used but are resolved with CM-3. Segmentations for classes containing small, sparse, and thin instances (CM-6), e.g. *low vegetation* and *built*, are hard to render due to low LULC resolution and low thermal contrast during label refinement.

Experimental Setup

Raster Acquisition: We acquired 10 m resolution Dynamic World LULC, 3D terrain data (3 m DEM, 1 m DEM, 2 m DSM) from USGS 3DEP, and high-resolution nadir imagery from NAIP (1 m) and Planet (3 m). Data was obtained via Microsoft Planetary Computer and Google Earth Engine.

Table 7.1: Evaluation of dense CRF refinement of Dynamic World LULC on NAIP imagery with ground truth labels from Chesapeake Bay Program (see class mapping in Fig. 7.3).

CRF cond. source	Boundary Loss ↓	mIoU ↑	Dense CRF Parameters							
			w_1	w_2	θ_γ	θ_α	$\theta_\beta^{\{0\}}$	$\theta_\beta^{\{1\}}$	$\theta_\beta^{\{2\}}$	$\theta_\beta^{\{3\}}$
None	0.945	0.432	—	—	—	—	—	—	—	—
RGB†	0.914	0.441	1.00	1.00	200	195	7.00	7.00	7.00	—
RGB	0.777	0.452	47.2	2.63	33.3	149	1.14	1.14	1.14	—
RGB-NIR	**0.749**	**0.453**	47.4	0.14	61.5	194	128	0.22	125	2.71

†tuned by minimizing weighted cross entropy instead of boundary loss

LULC from High-Resolution Imagery: We used networks trained on Chesapeake Bay Program (CBP) and OpenEarthMap (OEM) datasets to produce two more high-resolution LULC sources alongside Dynamic World. For OEM-derived LULC, we used the pretrained U-Net model from [44]. To produce CBP-derived LULC, we fine-tuned a geospatial foundation model [33] on the CBP dataset, using the 7-class set from [37].

We trained for 1000 epochs with a batch size of 16, a 1e^{-3} learning rate, and RGB-NIR inputs of size 512×512. To perform inference on large raster images, we use tiles with 50 % overlap and applied flips for test-time augmentation.

LULC Refinement with Dense CRFs: We refined the 10 m Dynamic World LULC rasters on RGB-NIR imagery from NAIP and Planet (Fig. 7.2) using parameters from Tab. 7.1. Parameters were found using Bayesian optimization with Optuna [2]. The search was done using NAIP as conditioning imagery and 1 m resolution labels from CBP as ground truth (see Fig. 7.3 for class mapping). For this use case, boundary loss was superior to standard cross-entropy loss (Tab. 7.1).

Rendered Label Refinement: For SAM refinement of the projected LULC labels, we used the default ViT-H model. We prompted with 32×32 grid points and lowered the box non-maximum suppression threshold to 0.5.

Thermal Image Preprocessing: We rescaled raw 16-bit thermal pixel intensities to sit between the 2nd and 98th percentiles before applying a contrast-limited adaptive histogram equalization with a 0.02 clip limit, following [25]. This was done for both visualization and algorithm input.

Table 7.2: LULC-generated semantic segmentation label assessment (mIoU) when compared to ground truth annotations, with comparisons against zero-shot visual foundation model baselines.

Method /	Dense CRF	3D source	Dataset mIoU ↑			Trajectory avg. mIoU ↑		
LULC source	refinement src.		CM-6	CM-5	CM-3	CM-6	CM-5	CM-3
ODISE [45]	—	—	0.299	0.262	0.330	0.264	0.304	0.413
OV-Seg [28]	—	—	0.201	0.240	0.385	0.183	0.233	0.390
Chesapeake Bay (NAIP)	—	DEM (3m)	0.453	0.481	0.857	0.417	0.478	0.848
Chesapeake Bay (Planet)	—	DEM (3m)	0.236	0.305	0.657	0.201	0.251	0.555
Open Earth Map (NAIP)	—	DEM (3m)	0.549	0.562	0.868	0.440	**0.528**	0.864
Open Earth Map (Planet)	—	DEM (3m)	0.502	0.509	0.825	0.360	0.428	0.816
Dynamic World	—	DEM (3m)	**0.577**	**0.572**	0.876	0.450	0.518	0.860
Dynamic World	NAIP	DEM (3m)	0.556	0.535	0.868	0.441	0.504	0.865
Dynamic World	Planet	DEM (3m)	0.573	0.557	**0.887**	**0.455**	0.510	**0.870**

Table 7.3: Ablation studies

(a) 3D source ablation

Method	3D source	Traj. avg. mIoU		
		CM-6	CM-5	CM-3
Dynamic World + SAM	DEM (3m)	**0.450**	**0.518**	**0.860**
	DEM (1m)	0.441	0.507	0.842
	DSM (2m)	0.439	0.504	0.848

(b) Label refinement ablation

Method	Projected label refine method	Traj. avg. mIoU		
		CM-6	CM-5	CM-3
Dynamic World + DEM (3m)	SAM	**0.450**	**0.518**	**0.860**
	SLIC	0.392	0.452	0.711
	Felzenszwab	0.369	0.426	0.677

Satellite-based Semantic Segmentation Label Generation

We compare our LULC-generated semantic segmentation labels to manually-annotated, ground truth labels. Due to class differences between LULC data and ground truth, we evaluate on three ground truth-derived class sets of increasing generality (CM-6, CM-5, CM-3). We report the overall dataset mIoU and the trajectory-averaged mIoU in Tab. 7.2.

Overall, our method delivers thermal semantic segmentation labels consistent with ground truth (Fig. 7.4). Notably, our best variants greatly outperform the zero-shot semantic segmentation models, ODISE [45], and OV-Seg [28], which were prompted with a list of classes present in the dataset. We note that ODISE and OV-Seg are occasionally effective on thermal images, but lack consistency.

Among our methods without LULC refinement, semantic segmentation label generation using Dynamic World and DEM (3 m) as a 3D source generally outperforms other variants using CBP- and OEM-derived LULC sources. LULC created from the OEM network on NAIP data provides improvements (0.005 - 0.01 mIoU) in trajectory-averaged mIoU over Dynamic World for the CM-5 and CM-3 class sets. Despite marginal gains, this is likely due to the higher resolution (1 m) of OEM/NAIP-derived LULC, which enables segmentation renderings of small or thin classes that are present in CM-5, such as roads (see Fig. 7.4). This is not possible with Dynamic World due to its lower 10 m resolution.

Conversely, LULC generated from Planet imagery provides poor results due to domain differences between OEM/CBP training images and Planet. When used for dense CRF refinement of 10 m Dynamic World rasters, however, Planet imagery uniquely provides ~0.01 boost for both mIoU metrics on the most general CM-3 class set. This behavior is absent when refining on NAIP imagery due to terrain changes between thermal and NAIP acquisition dates.

Furthermore, we note that our method can handle temporal mismatches between satellite and thermal data even as environments naturally evolve. For example, coastal tide patterns and varying lake levels (Fig. 7.5, Castaic Lake) may shift class boundaries within short time periods. Due to SAM's ability to capture entire class instances, our rendered label refinement step (Sec. 7.5) is notably able to overcome such changes as long as most of the true class is still rendered.

Due to its accessibility and competitive performance, we advocate using Dynamic World LULC for satellite-based semantic segmentation label generation efforts, with potential refinement via temporally-relevant, high resolution imagery. However, this will inevitability change with advancements in LULC creation and as sub-meter data products with high temporal coverage become more freely-accessible.

Ablation Study

In these ablations, we use Dynamic World as our semantic source. Unless otherwise specified, we use 3 m DEMs to add 3D context and do not use any CRF refinement.

3D Data Source: First, we compare LULC-based semantic segmentation label generation with 3 m DEMs against two additional 3D data sources: 2 m DSMs and 1 m DEMs. Due to limited coverage, we lack DSMs and 1 m DEMs over the thermal data capture areas of Colorado River and Duck, respectively, and resort to 3 m DEMs in those areas. Our results show that 3 m DEMs provide consistently

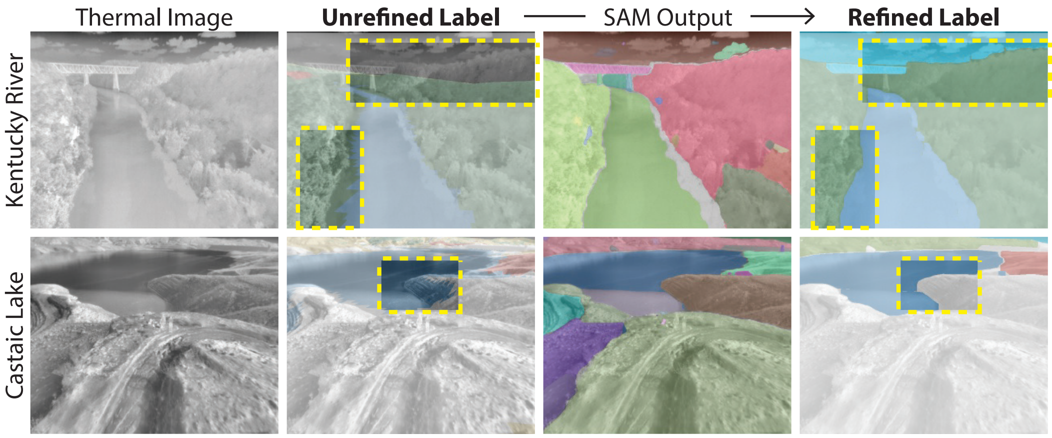


Figure 7.5: Rendered label refinement process with SAM [23].

higher trajectory-averaged mIoU across all three class sets, despite the other two sources supposedly providing more accurate and precise 3D terrain data (Tab. 7.3a). Reasons for this may include temporal differences or spatial misalignment during orthorectification. Nonetheless, all 3D sources generally perform well and any one of these 3D products can be used for our method when the other two are unavailable.

Raster Spatial Resolution: To assess the impact of LULC spatial resolution on label generation, we generate labels from Dynamic World LULC rasters resampled to 10 m (native), 5 m, and 1 m resolution. We use nearest neighbor interpolation on the LULC directly, and CRF refinement on NAIP and Planet rasters (resampled to 10 m, 5 m, and 1 m resolutions via bicubic interpolation).

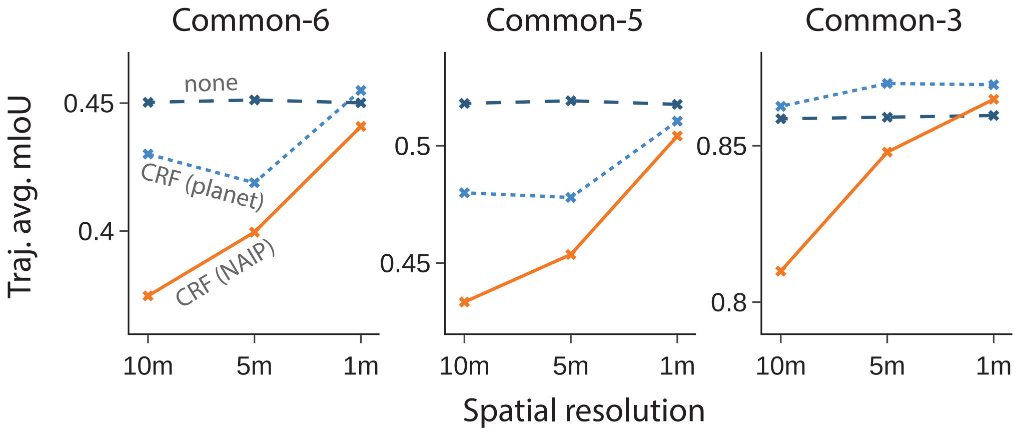


Figure 7.6: Effect of LULC spatial resolution on semantic segmentation label generation.

Our results (Fig. 7.6) suggest that LULC spatial resolution matters more for more specific class sets (CM-6/CM-5), and becomes less critical as class sets generalize (CM-3). Moreover, we find greater benefits from CRFs when conditioning on higher-resolution imagery, especially when dealing with the larger and more specific class sets (CM-6/CM-5). This is likely due to smoothing over small or thin class instances that comprise of a few pixels when refining at lower resolutions.

Segment Anything vs. Classical Methods for Projected LULC Refinement: We compare our choice of SAM for projected LULC label refinement against SLIC [1] and Felzenszwab [14] superpixels. We use implementations from `scikit-image` [43], setting SLIC's number of segments to 100 and compactness to 10, and Felzenszwab's scale parameter to 1e^4. We select these parameters to maximize segmentation area while remaining within class instances.

Overall, SAM consistently outperforms the other methods, with mIoU margins increasing from 0.06 (Tab. 7.3b). This is because SAM can produce semantically distinct masks in the thermal domain, albeit less reliably than in RGB. This allows minor imperfections to be ignored through majority vote (Alg. 3). In contrast, classical methods produce fragmented, semantic-agnostic masks which offer little benefit.

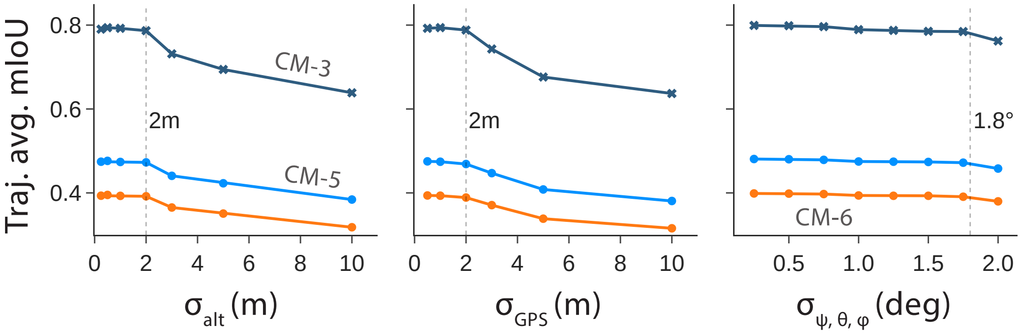


Figure 7.7: Effect of global pose estimate precision on semantic segmentation label generation with SAM refinement.

State Estimate Precision: To quantify the effect of global pose estimation precision on our label generation process, we systematically perturb these measurements by sampling from a normal distribution with increasing variance. Our analysis reveals that, with 95 % confidence, label generation remains robust for global positioning and altitude estimates within roughly 4 m and for orientations within roughly 3.5° (Fig. 7.7). These findings are consistent across class sets. During development, both synchronizing the timing of image capture to the IMU data was shown to be critical, as was the SAM refinement stage for compensation for attitude estimate errors (see Kentucky River in Fig. 7.5).

Application: Semantic Segmentation Model Training

To demonstrate our method for field robot perception, we trained an EfficientViT-B0 semantic segmentation network [10] using the aerial thermal dataset and general train/val/test split from [26]. Three sets of labels (CM-6, CM-5, and CM-3) were generated for training and validation using our method, with ground truth labels converted accordingly for testing and baseline training. All networks were trained following the thermal training procedure from [26].

Our semantic segmentation results (Tab. 7.4a) closely match the mIoU of the generated annotations (Tab. 7.2). Networks trained with CM-3 classes resulted in 0.889 mIoU during testing, compared to 0.962 mIoU for those trained with ground truth labels. Networks trained on CM-5 and CM-6 show larger gaps (Tab. 7.4a) but still show the benefit of our method. We find this is largely due to difficulties in accurately rendering land-based classes, specifically *low vegetation* and *built* (Tab. 7.4b). These classes contain small and thin entities like sparse shrubs or roads, and are not always precisely shown in LULC data. Also, they can be missed during rendered label refinement (Sec. 7.5) due to blurred and low-contrast appearance in thermal imagery. Despite this, our method can effectively train semantic segmentation models, particularly with the CM-3 class set, and support field robotic applications like nighttime river navigation [25].

Table 7.4: Test results (mIoU) of semantic segmentation networks trained on LULC-generated labels and networks trained on manually-annotated ground truth.

(a) Segmentation results after training on CM-6 (least inclusive), CM-5, and CM-3 (most inclusive) class sets.

Annotation Method	Class set		
	CM-6	CM-5	CM-3
LULC-generated	0.542	0.547	0.889
Ground truth	0.819	0.836	0.962

(b) Per-class IoU for networks trained using the CM-6 class set.

Annotation Method	water	trees	low veg.	built	ground	sky
Generated	0.880	0.529	0.165	0.289	0.521	0.868
Ground truth	0.963	0.787	0.702	0.653	0.854	0.955

Computational Costs and Pricing

Our method annotates a single image in 3 s, 2.86 s of which is due to SAM. Annotations are *free* when using only Dynamic World LULC but cost ~\$10 USD/km^2 with CRF refinement due to the price of realtime, high-resolution satellite imagery. With our method, annotating 2 000 images takes 1.6 hours on a single workstation, in contrast with the usual 2-4 week timeframe and \$3 000 to \$8 000 USD outsourcing cost[1]. We note that CRF refinement can be cost-effective for large data volumes in a concentrated area due to its one-time cost, but 98.5 % of its performance (CM-6, CM-3) is achievable with free 10 m resolution LULC (Sec. 7.7).

7.8 Conclusion

We presented a novel method for automatically generating high-quality semantic segmentation annotations for classes often encountered by aerial robots in field settings. Our approach leverages satellite data products and employs refinement steps to achieve fine precision at class boundaries even with low-resolution satellite data, achieving 98.5% of the performance of costly high-resolution options. We demonstrated the robustness of our method to global positioning and attitude estimation errors, indicating that it can provide good segmentations even with inexpensive sensors and slight data desynchronization, and identified limitations due to small and thin class instances. Lastly, we demonstrated its application to field robot perception by successfully training a semantic segmentation network solely with generated labels. This method enables rapid training of thermal perception stacks using incremental learning as new field data is collected.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. “SLIC superpixels compared to state-of-the-art superpixel methods”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.11 (2012), pp. 2274–2282.

[2] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. “Optuna: A Next-generation Hyperparameter Optimization Framework”. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2019.

[1] \$1.50 to \$4.00 per image for 1-10 semantic segmentation classes, based on pricing from Scale AI at the time of writing: https://scale.com/pricing

[3] N. Araslanov and S. Roth. “Self-supervised augmentation consistency for adapting semantic segmentation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2021, pp. 15384–15394.

[4] A. Berg, J. Johnander, F. Durand de Gevigney, J. Ahlberg, and M. Felberg. “Semi-Automatic Annotation of Objects in Visual-Thermal Video”. In: *Proc. IEEE Int. Conf. Comput. Vis. Workshop*. 2019, pp. 2242–2251. DOI: 10.1109/ICCVW.2019.00277.

[5] A. Bokhovkin and E. Burnaev. “Boundary loss for remote sensing imagery semantic segmentation”. In: *Proceedings of the International Symposium on Neural Networks*. Springer. 2019, pp. 388–401.

[6] E. Bondi, R. Jain, P. Aggrawal, S. Anand, R. Hannaford, A. Kapoor, J. Piavis, S. Shah, L. Joppa, B. Dilkina, et al. “BIRDSAI: A dataset for detection and tracking in aerial thermal infrared videos”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2020, pp. 1747–1756.

[7] A. Braun and A. Borrmann. “Combining inverse photogrammetry and BIM for automated labeling of construction site images for machine learning”. In: *Automation in Construction* 106 (2019), p. 102879.

[8] K. L. Brodie, B. L. Bruder, R. K. Slocum, and N. J. Spore. “Simultaneous Mapping of Coastal Topography and Bathymetry From a Lightweight Multicamera UAS”. In: *IEEE Transactions on Geoscience and Remote Sensing* 57.9 (2019), pp. 6844–6864. DOI: 10.1109/TGRS.2019.2909026.

[9] C. F. Brown, S. P. Brumby, B. Guzder-Williams, T. Birch, S. B. Hyde, J. Mazzariello, W. Czerwinski, V. J. Pasquarella, R. Haertel, S. Ilyushchenko, et al. “Dynamic World, Near real-time global 10 m land use land cover mapping”. In: *Scientific Data* 9.1 (2022), p. 251.

[10] H. Cai, J. Li, M. Hu, C. Gan, and S. Han. “Efficientvit: Lightweight multi-scale attention for high-resolution dense prediction”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 17302–17313.

[11] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. “Rethinking atrous convolution for semantic image segmentation”. In: *arXiv preprint arXiv:1706.05587* (2017).

[12] S. Daftry, Y. Agrawal, and L. Matthies. “Online Self-Supervised Long-Range Scene Segmentation for MAVs”. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2018, pp. 5194–5199.

[13] J. Delaune, R. Hewitt, L. Lytle, C. Sorice, R. Thakker, and L. Matthies. “Thermal-inertial odometry for autonomous flight throughout the night”. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2019, pp. 1122–1128.

[14] P. F. Felzenszwalb and D. P. Huttenlocher. “Efficient graph-based image segmentation”. In: *International Journal of Computer Vision* 59 (2004), pp. 167–181.

[15] J. E. Gallagher, A. Gogia, and E. J. Oughton. “A Multispectral Automated Transfer Technique (MATT) for machine-driven image labeling utilizing the Segment Anything Model (SAM)”. In: *arXiv preprint arXiv:2402.11413* (2024).

[16] L. Gan, C. Lee, and S.-J. Chung. “Unsupervised RGB-to-thermal domain adaptation via multi-domain attention network”. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. 2023, pp. 6014–6020.

[17] Q. Ha, K. Watanabe, T. Karasawa, Y. Ushiku, and T. Harada. “MFNet: Towards real-time semantic segmentation for autonomous vehicles with multi-spectral scenes”. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2017, pp. 5108–5115.

[18] K. He, G. Gkioxari, P. Dollár, and R. Girshick. “Mask R-CNN”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 2961–2969.

[19] A. Jong, M. Yu, D. Dhrafani, S. Kailas, B. Moon, K. Sycara, and S. Scherer. “WIT-UAS: A Wildland-fire Infrared Thermal Dataset to Detect Crew Assets From Aerial Views”. In: *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2023, pp. 11464–11471.

[20] K. Kamnitsas, C. Ledig, V. F. Newcombe, J. P. Simpson, A. D. Kane, D. K. Menon, D. Rueckert, and B. Glocker. “Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation”. In: *Medical Image Analysis* 36 (2017), pp. 61–78.

[21] K. Karra, C. Kontgis, Z. Statman-Weil, J. C. Mazzariello, M. Mathis, and S. P. Brumby. “Global land use / land cover with Sentinel 2 and deep learning”. In: *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium*. 2021, pp. 4704–4707. DOI: 10.1109/IGARSS47720.2021.9553499.

[22] Y.-H. Kim, U. Shin, J. Park, and I. S. Kweon. “MS-UDA: Multi-Spectral Unsupervised Domain Adaptation for Thermal Image Semantic Segmentation”. In: *IEEE Robotics and Automation Letters* 6.4 (2021), pp. 6497–6504. DOI: 10.1109/LRA.2021.3093652.

[23] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick. “Segment Anything”. In: *arXiv:2304.02643* (2023).

[24] P. Krähenbühl and V. Koltun. “Efficient inference in fully connected crfs with gaussian edge potentials”. In: *Proceedings of the Advances in Neural Information Processing Systems Conference* 24 (2011).

[25] C. Lee, J. G. Frennert, L. Gan, M. Anderson, and S.-J. Chung. “Online Self-Supervised Thermal Water Segmentation for Aerial Vehicles”. In: *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2023, pp. 7734–7741.

[26] C. Lee*, M. Anderson*, N. Raganathan, X. Zuo, K. Do, G. Gkioxari, and S.-J. Chung. “CART: Caltech Aerial RGB-Thermal Dataset in the Wild”. In: *arXiv preprint arXiv:2403.08997* (2024). Available at https://arxiv.org/abs/2403.08997.

[27] C. Li, W. Xia, Y. Yan, B. Luo, and J. Tang. “Segmenting objects in day and night: Edge-conditioned cnn for thermal image semantic segmentation”. In: *IEEE Transactions on Neural Networks and Learning Systems* 32.7 (2020), pp. 3069–3082.

[28] F. Liang, B. Wu, X. Dai, K. Li, Y. Zhao, H. Zhang, P. Zhang, P. Vajda, and D. Marculescu. “Open-vocabulary semantic segmentation with mask-adapted clip”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 7061–7070.

[29] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. “Swin transformer: Hierarchical vision transformer using shifted windows”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2021, pp. 10012–10022.

[30] G. Loianno, V. Spurny, J. Thomas, T. Baca, D. Thakur, D. Hert, R. Penicka, T. Krajnik, A. Zhou, A. Cho, M. Saska, and V. Kumar. “Localization, Grasping, and Transportation of Magnetic Objects by a Team of MAVs in Challenging Desert-Like Environments”. In: *IEEE Robotics and Automation Letters* 3.3 (2018), pp. 1576–1583. DOI: 10.1109/LRA.2018.2800121.

[31] A. Marcu, V. Licaret, D. Costea, and M. Leordeanu. “Semantics through time: Semi-supervised segmentation of aerial videos with iterative label propagation”. In: *Proceedings of the Asian Conference on Computer Vision*. 2020.

[32] A. Mehra, B. Kailkhura, P.-Y. Chen, and J. Hamm. “Understanding the limits of unsupervised domain adaptation via data poisoning”. In: *Proceedings of the Advances in Neural Information Processing Systems Conference* 34 (2021), pp. 17347–17359.

[33] M. Mendieta, B. Han, X. Shi, Y. Zhu, and C. Chen. “Towards Geospatial Foundation Models via Continual Pretraining”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2023, pp. 16806–16816.

[34] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos. “Image segmentation using deep learning: A survey”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.7 (2021), pp. 3523–3542.

[35] S. Nirgudkar, M. DeFilippo, M. Sacarny, M. Benjamin, and P. Robinette. "MassMIND: Massachusetts Maritime INfrared Dataset". In: *International Journal of Robotics Research* 42.1-2 (2023), pp. 21–32.

[36] A. Pretto, S. Aravecchia, W. Burgard, N. Chebrolu, C. Dornhege, T. Falck, F. Fleckenstein, A. Fontenla, M. Imperoli, R. Khanna, et al. "Building an aerial–ground robotics system for precision farming: an adaptable solution". In: *IEEE Robotics and Automation Magazine* 28.3 (2020), pp. 29–49.

[37] C. Robinson, L. Hou, K. Malkin, R. Soobitsky, J. Czawlytko, B. Dilkina, and N. Jojic. "Large scale high-resolution land cover mapping with multi-resolution data". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 12726–12735.

[38] Scale AI. *Introducing Scale's Automotive Foundation Model*. https://scale.com/blog/afm1. 2023.

[39] S. S. Shivakumar, N. Rodrigues, A. Zhou, I. D. Miller, V. Kumar, and C. J. Taylor. "Pst900: Rgb-thermal calibration, dataset and segmentation network". In: *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2020, pp. 9441–9447.

[40] U.S. Department of Agriculture Farm Service Agency, Aerial Photography Field Office. *National Agricultural Imagery Program*. earthexplorer.usgs.gov. 2012.

[41] U.S. Geological Survey. *3D Elevation Program*. https://usgs.gov/3d-elevation-program.

[42] J. Vertens, J. Zürn, and W. Burgard. "Heatnet: Bridging the day-night domain gap in semantic segmentation with thermal images". In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2020, pp. 8461–8468.

[43] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, and the scikit-image contributors. "scikit-image: image processing in Python". In: *PeerJ* 2 (2014), e453. ISSN: 2167-8359. DOI: 10.7717/peerj.453.

[44] J. Xia, N. Yokoya, B. Adriano, and C. Broni-Bediako. "OpenEarthMap: A benchmark dataset for global high-resolution land cover mapping". In: *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*. 2023, pp. 6254–6264.

[45] J. Xu, S. Liu, A. Vahdat, W. Byeon, X. Wang, and S. De Mello. "Open-vocabulary panoptic segmentation with text-to-image diffusion models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 2955–2966.

[46] B. Yu, R. Tibbetts, T. Barna, A. Morales, I. Rekleitis, and M. J. Islam. "Weakly Supervised Caveline Detection For AUV Navigation Inside Underwater Caves". In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2023, pp. 9933–9940.

Chapter 8

RGB-X OBJECT DETECTION VIA SCENE-SPECIFIC FUSION MODULES

[1] S. A. Deevi*, C. Lee*, L. Gan*, S. Nagesh, G. Pandey, and S.-J. Chung. “RGB-X Object Detection via Scene-Specific Fusion Modules”. In: *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2024, pp. 7351–7360. DOI: 10.1109/WACV57701.2024.00720.

8.1 Abstract

Multimodal deep sensor fusion has the potential to enable autonomous vehicles to visually understand their surrounding environments in all weather conditions. However, existing deep sensor fusion methods usually employ convoluted architectures with intermingled multimodal features, requiring large coregistered multimodal datasets for training. In this work, we present an efficient and modular RGB-X fusion network that can leverage and fuse pretrained single-modal models via scene-specific fusion modules, thereby enabling joint input-adaptive network architectures to be created using small, coregistered multimodal datasets. Our experiments demonstrate the superiority of our method compared to existing works on RGB-thermal and RGB-gated datasets, performing fusion using only a small amount of additional parameters.

8.2 Introduction

Autonomous vehicles rely on object detection algorithms to understand and interact with their surrounding environments. In order to be robust against different driving conditions, these algorithms operate on data from various sensor modalities ranging from optical cameras to LiDAR, each with their own advantages and disadvantages. Because no single sensor modality is robust to all possible conditions that may be encountered during driving, multiple sensor modalities are often used in conjunction via *deep sensor fusion (DSF)* to boost performance during normal driving operations, as well as to ensure segmentation and object detection reliability in adverse weather conditions [8].

Unlike traditional sensor fusion which merges processed sensor data outputs coming from independent pipelines, current works in DSF generally require joint end-to-end

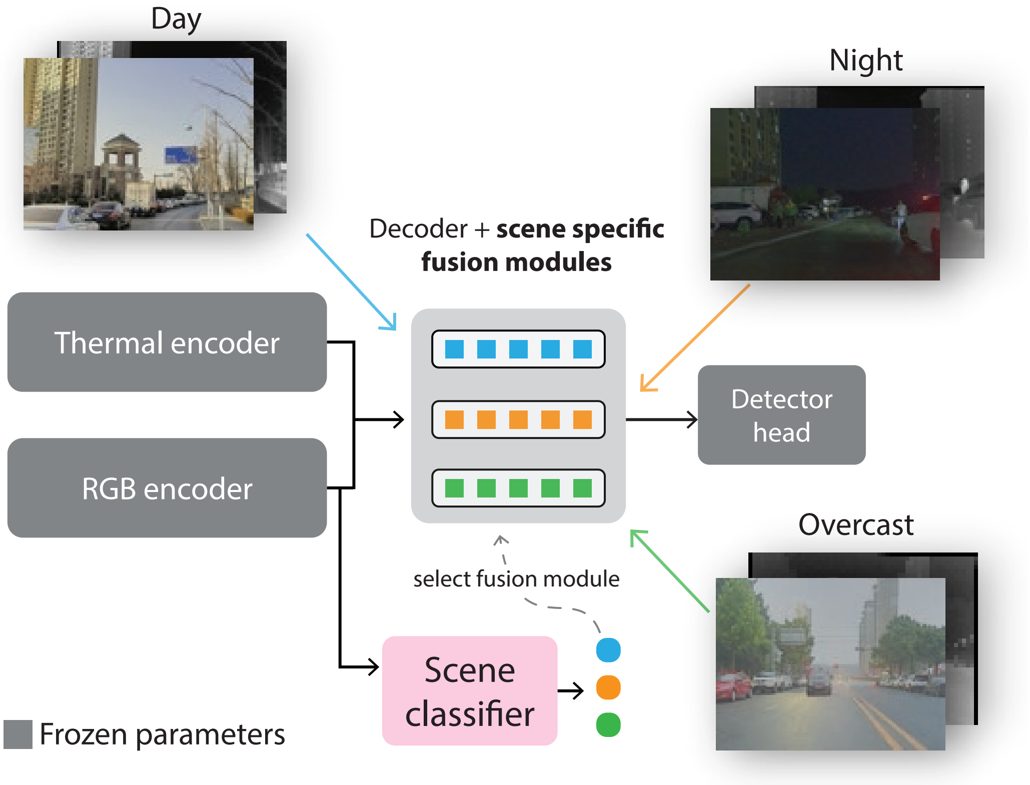


Figure 8.1: Our multimodal object detection approach combines RGB and thermal pretrained networks using lightweight, scene-specific fusion modules. Fusion modules are trained using categorized scene images and are used adaptively during inference with an auxiliary scene classifier.

training of multi-branch sensor networks on large multimodal datasets [1, 3, 39, 17, 25, 40] such as NuScenes [2], Berkeley Deep Drive [38], and Waymo [28] prior to deployment in the wild [8]. This means that fusion architectures must undergo time-consuming and potentially expensive retraining (in cost and carbon emissions) anytime a sensor modality is removed or added [24], and that they fail to take full advantage of state-of-the-art RGB pretrained networks.

In this paper, we propose the use of existing, well-known attention blocks as lightweight, scene-specific attention modules in order to easily fuse pretrained networks and to better adapt to common weather disturbances. We demonstrate our approach (Fig. 8.1) for object detection applications, training RGB-thermal and RGB-gated fusion models on RGB, thermal, and gated imagery collected in adverse driving conditions such as night, fog, snow, and rain [1, 20, 9]. We also leverage the attention modules as a method to visually interpret the contributions of each sensor modality. Compared to prior works, our approach takes us another step closer to enabling a modular, *drag-and-drop* design for deep sensor fusion that absolves the need for extensive and expensive retraining while delivering on-par or better performance. Our contributions are as follows:

1. A lightweight, modular RGB-X fusion network for object detection that leverages pretrained single-modality networks.
2. A scene-adaptive fusion approach that selectively uses different fusion modules for different scene/weather conditions.
3. Extensive experiments on publicly available RGB-X datasets that demonstrate the superiority of our approach in terms of detection performance and computational efficiency.

8.3 Related Work

Object Detection: Most modern methods for detecting objects utilize convolutional neural networks (CNN) or transformers. CNN object detectors include two-stage and single-shot detectors [27, 26, 21, 31]. A two-stage detector has an additional region proposal step while a single-shot detector relies only on a feature extractor and a detection head that directly predicts bounding boxes and classes, resulting in faster inference [43]. To deploy on mobile devices, neural architecture search (NAS) has been used to develop faster and lighter networks and detection architectures [14, 31]. In this work, we adopt the EfficientDet [31] detection architecture to target self-driving car applications that operate on mobile computing devices. Recent large vision transformer models have achieved state-of-the-art object detection results, but are not suitable for real-time use on robotic platforms [4, 22].

Deep Sensor Fusion: Robotic perception applications, notably for self-driving cars, rely on DSF to add sensor redundancy and to increase perception robustness and performance in both common and adverse operating scenarios. Current DSF algorithms consume multimodal data using deep networks and are trained end-to-end, combining different features at various points throughout a network depending on their particular fusion policy [8, 7]. Early fusion policies aggregate raw inputs or features extracted early on in the network [32, 19] while mid-fusion approaches [16, 35] operate on deeper, intermediate representations. Late fusion methods operate directly on bounding box outputs and can be used directly with pretrained detectors, but are subject to the performance of pretrained models [5]. In our work, we opt for a mid-fusion approach in order to take full advantage of the different feature modalities at various stages.

Regardless of fusion policies, current DSF algorithms and datasets for self-driving cars mainly focus on incorporating sensors like LiDAR and radar with RGB cameras [35, 38, 2, 28, 12]. In our work, we are interested in supplementing RGB with 2D image data from thermal and gated cameras due to the rich semantic information they provide and their robustness to fog and lighting in driving scenarios [8].

RGB-Thermal Object Detection: Current RGB-thermal (RGB-T) object detection methods typically operate on aligned RGB-thermal image pairs and utilize some form of attention-based modules to perform mid-fusion on RGB and thermal image features. [40] utilizes intra-modality and inter-modality spatial attention modules to enhance and adaptively fuse intermediate features, respectively, prior to passing downstream to a detection head. Recently, [3] proposed mid-fusion modules that utilize channel attention to dynamically swap RGB and thermal feature channels. This helps to maximize feature usefulness before enhancing local features via parameter-free spatial attention. Other works including [10, 42, 25] fuse multi-modal data in a similar fashion but instead leverage transformer-based attention modules that increase model size and computational cost. [1] does not use thermal images, but similarly fuses RGB, gated, and projected LiDAR and radar data using local entropy masks in lieu of attention. In our work, we demonstrate that pretrained, single-modality detectors can be fused using simple, scene-specific channel and spatial attention modules to achieve strong RGB-T object detection performance.

8.4 Approach

We propose a modular RGB-X fusion network for object detection that is built upon pretrained single-modal detection architecture and multi-stage convolutional block attention modules (CBAM) [34] for cross-modal feature fusion. This modularity separates the training of single-modal backbones that contain the majority of network parameters and the training of a small fusion module, mitigating the requirement of large-scale multi-modal training data. The overall architecture for RGB-X fusion is shown in Fig. 8.2 using RGB-T as an example. We have an individual EfficientDet [31] for each image modality consisting of an EfficientNet [30] backbone network, a bidirectional feature pyramid network (BiFPN) and a detector head. While we choose to use EfficientDet to demonstrate our approach, we note that this architecture can be built using any single-modal detection network.

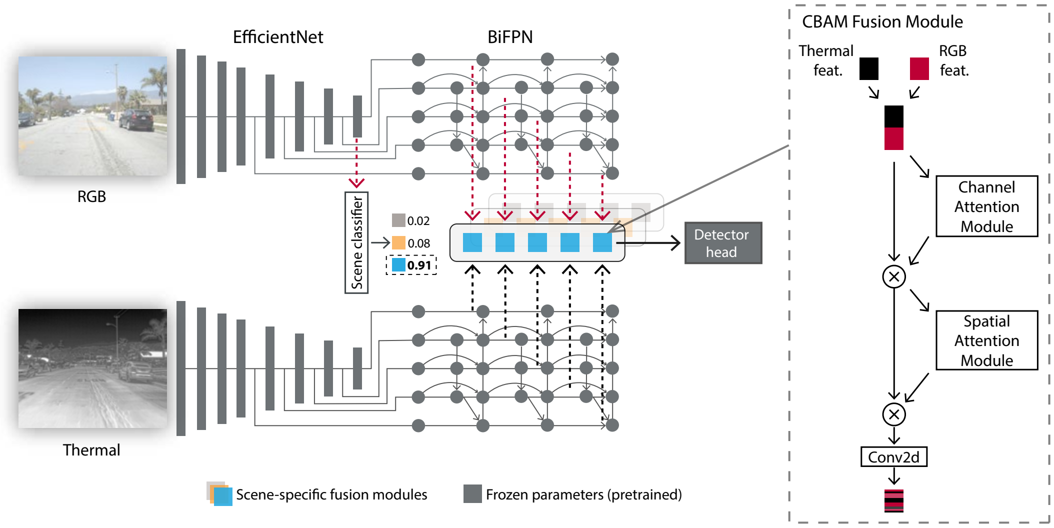


Figure 8.2: Overall framework of our scene-adaptive CBAM model for RGB-X fusion illustrated by RGB-T fusion. RGB and thermal images are processed by separate EfficientNet backbones, followed by BiFPNs. The features from BiFPNs are used for cross-modal feature fusion using modules selected by the scene classifier. The detector head utilizes these fused features to obtain the final detection results. The right side of the figure illustrates the CBAM fusion module, consisting of channel and spatial attention blocks, for feature fusion.

We employ CBAM to fuse the RGB and thermal features output from the respective BiFPN at various stages. Each CBAM fuses features at the same scale, resulting in 5 CBAM fusion modules. During training, only CBAM parameters are updated while pretrained object detector weights are frozen. CBAM modules are trained per scene category and are selected for use during inference time using an auxiliary scene classifier. In the rest of this section, we go over the details of our fusion mechanism and the auxiliary scene classifier, before describing the overall scene-adaptive fusion algorithm for RGB-X object detection.

Convolutional Block Attention Fusion

We use CBAM to fuse RGB and thermal (or gated) CNN feature maps $\mathbf{F_{rgb}}$ and $\mathbf{F_x}$, respectively. We concatenate features from both modalities across the channel dimension to create an input feature map $\mathbf{F}$ for CBAM:

$$\mathbf{F} = [\mathbf{F_{rgb}}; \mathbf{F_x}] \in \mathbf{R}^{B\times C\times H\times W}, \tag{8.1}$$

where B denotes the batch size, and C, H, W denote the channel and spatial dimensions of the feature, respectively. Following the notation in [34], a CBAM module takes the feature map $\mathbf{F}$ and masks it using channel and spatial attention operators

M_c, M_s such that

$$\mathbf{F}' = M_c(\mathbf{F}) \otimes \mathbf{F}, \tag{8.2}$$

$$\mathbf{F}'' = M_s(\mathbf{F}') \otimes \mathbf{F}', \tag{8.3}$$

where $\otimes$ denotes element-wise multiplication. We further convolve $\mathbf{F}''$ with $C/2$ kernels resulting in $C/2$ channels which is the original feature dimension.

Channel attention operator M_c is computed via

$$M_c(\mathbf{F}) = \sigma(\mathbf{W_1 W_0 F^c_{avg}} + \mathbf{W_1 W_0 F^c_{max}}), \tag{8.4}$$

where σ, $\mathbf{W}$, $\mathbf{F^c_{avg}}$, $\mathbf{F^c_{max}}$ denotes the sigmoid function, linear layer weights, the global average and max pooled features, respectively. Spatial attention is computed via

$$M_s(\mathbf{F}) = \sigma(f^{7\times7}([\mathbf{F^c_{avg}}; \mathbf{F^c_{max}}])), \tag{8.5}$$

where $\mathbf{F^c_{avg}}$, $\mathbf{F^c_{max}}$ are computed via channel-wise mean and max operations and $f^{7\times7}$ denotes convolution with a kernel size of 7.

Auxiliary Scene Classification

We utilize a simple scene classifier during inference time to adaptively select the most suitable set of fusion modules for the current setting, based on the intuition that the fusion module should attend different modalities under different scene/weather conditions. The scene classifier consists of a 2D adaptive average pooling operator followed by a fully connected layer, taking in the features from the RGB object detector encoder and outputs probabilities of possible scene categories. We choose RGB features as the input for scene classification due to their high variance in different scenes.

Scene-Specific Fusion

We train different CBAM fusion modules for various scenes by considering scene-specific dataset splits (Table 8.2). The number of parameters in different parts of the proposed fusion model is shown in Table 8.1. The total number of trainable parameters per scene is significantly less than the total number of parameters, making our approach expeditious. During inference of scene-adaptive fusion, we use the CBAM fusion modules trained on the scene with the highest probability, as indicated by the scene classifier.

Table 8.1: Parameter statistics of the proposed RGB-X fusion model.

Network Part	# Parameters
Backbones (RGB + X)	24.8 M
BiFPNs (RGB + X)	0.12 M
Detection Head	1.60 M
Fusion Modules (one per fusion level)	0.21 M
Total	26.7 M
Total Trainable (per scene)	0.21 M

8.5 Results

Implementation and Training Details

Our code is written in PyTorch and based on the EfficientDet[1] repository. Pretrained RGB detectors on COCO dataset [18] were taken from the same repository. All other networks were trained using the Adam optimizer, a batch size of 8, an initial learning rate of $1e^{-3}$ with an exponential learning rate schedule, and a L_2 weight decay of $1e^{-3}$. The maximum number of epochs is set to 300 and 50 for pretraining single modality networks and fine-tuning RGB-X fusion networks, respectively. The scene classifier is trained for 50 epochs while the RGB backbone remains frozen. Networks were trained using an Nvidia P100 GPU.

Datasets

We use the following RGB-X datasets to validate our method and compare against state-of-the-art baselines. The train/val/test split statistics we use for various datasets and scene conditions are shown in Table 8.2.

FLIR Aligned: The FLIR Aligned dataset [39] consists of 5,142 aligned RGB-thermal image pairs from the original FLIR ADAS object detection dataset [9]. This derived dataset consists of bounding box annotations for *person*, *bicycle* and *car* classes. The provided train and test splits contain 4,129 and 1,013 image pairs, respectively. We manually divided them into *day* and *night* scene categories based on the appearance.

M^{3}FD: The M^{3}FD object detection dataset consists of 4,200 coregistered, time-synchronized RGB-thermal image pairs [20]. Bounding box annotations for *people, car, bus, motorcycle, truck,* and *lamp* classes are provided. The data is also split into four scene categories *(day, night, overcast, challenge)* in [20] according to

[1] https://github.com/rwightman/efficientdet-pytorch

Table 8.2: Dataset scene and train/val/test splits in our experiments.

(a) Seeing Through Fog (STF)

Split	Clear		Fog		Snow	
	Day	Night	Day	Night	Day	Night
Train	2147	1572	712	438	1365	1455
Val	537	393	438	110	342	364
Test	895	655	297	183	570	607

(b) FLIR Aligned [39]

Split	Scene Condition	
	Day	Night
Train	3476	653
Val	—	—
Test	702	311

(c) M^{3}FD [20]

Split	Scene Condition			
	Day	Night	Overcast	Challenge
Train	992	488	746	484
Val	216	108	190	122
Test	323	140	205	156

environment characteristics. We use the train/val/test splits provided by [17] due to the unavailability in the original dataset.

Seeing Through Fog: The Seeing Through Fog (STF) multispectral object detection dataset [1] consists of synchronized RGB/gated/LiDAR/radar/unaligned thermal data for a variety of weather conditions. The dataset also provides bounding box annotations for *pedestrian*, *truck*, *car*, *cyclist*, and *dontcare* classes. For training our scene-adaptive model, we considered the scene splits in Table 8.2a due to overlaps in original splits. For evaluation, we follow the original scene splits including *clear*, *light fog*, *dense fog*, and *snow/rain*. We use pairs of aligned 12-bit RGB and 10-bit gated images throughout this work.

Performance Evaluation

In this section, we validate the proposed method on the three datasets for RGB-X object detection. Auxiliary scene classification is employed to adaptively select suitable fusion modules per input image.

Auxiliary Scene Classification: We train our scene classifiers using ground truth scene labels provided in Table 8.2 by minimizing the standard cross-entropy loss for image classification. Top-1 accuracy of the scene classification is reported in Table 8.3, where the classifier attains high accuracy for categorizing various scenes in M^{3}FD, FLIR, and STF-Clear (the subset of STF dataset consists of *clear*-

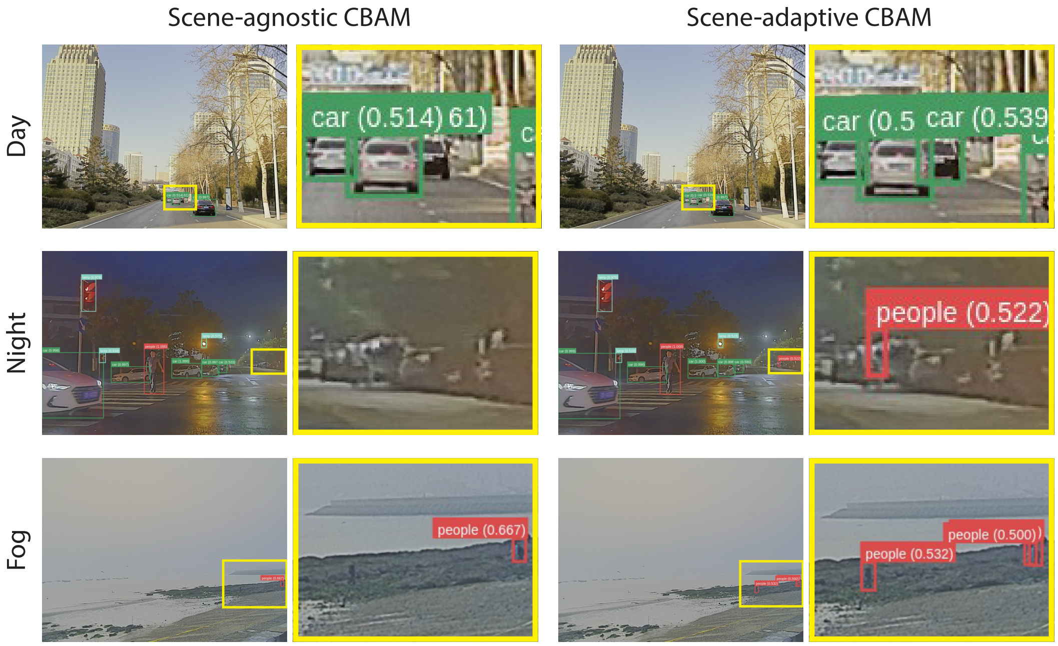


Figure 8.3: Qualitative detection results on M^{3}FD dataset. Zoomed-in images (yellow rectangle) are shown on the right of the original images for better visualization.

Table 8.3: Top-1 Accuracy (%) of our scene classifier on the test set of three datasets.

Dataset	M^{3}FD	FLIR	STF (Clear)	STF (Full)
Accuracy	91.42	96.35	96.01	77.02

day and *clear-night*) datasets. The classifier does not perform as high for STF-Full, possibly because a large portion of *snow* images are also foggy and confused the classifier.

Scene-Adaptive Object Detection: This subsection reports quantitative and qualitative object detection results of our proposed methods, compared with existing works. From Table 8.4, our scene-adaptive CBAM model outperforms existing methods on the M^{3}FD dataset using the mean Average Precision IoU = 0.5 (mAP@0.5) metric used in [17, 20]. On the *full* test set, it outperforms EAEFNet [17] by 1.4% and the scene-agnostic CBAM model (in which only one set of CBAM fusion modules are trained using all training images) by 1%. A comparison of qualitative detection results on M^{3}FD dataset between the scene-agnostic and scene-adaptive models is shown in Fig. 8.3. From the zoomed-in area of the figures, we can see that the scene-adaptive model detects some occluded, blurred objects that the scene-agnostic model fails to detect. Note that the single-modality models used in this experiment are pretrained on COCO and further fine-tuned on

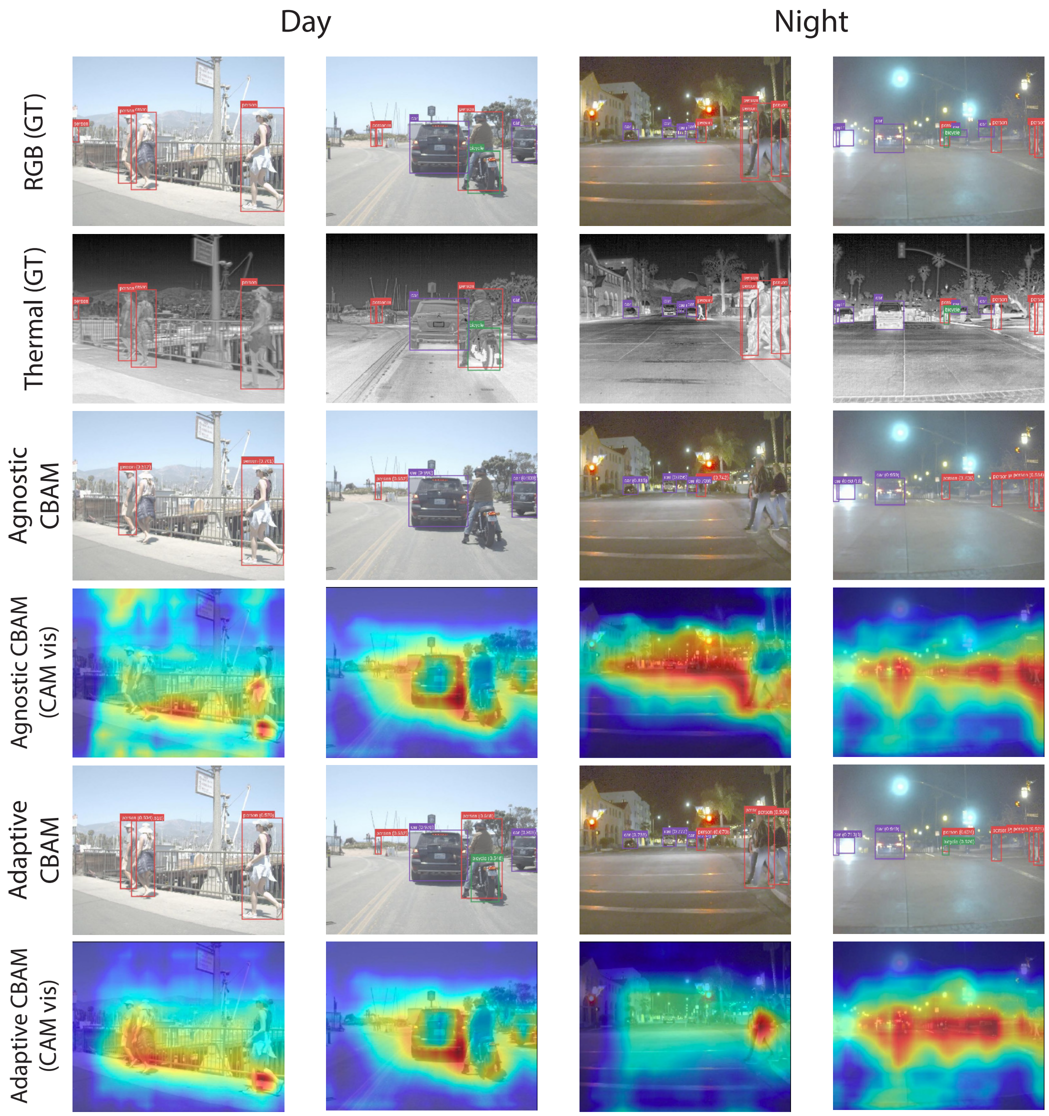


Figure 8.4: Qualitative detection results on FLIR Aligned dataset with *day* examples in the upper rows and *night* examples in the lower rows. The input RGB and thermal images are overlaid with ground truth (GT) bounding boxes. For each fusion model, we plot the detected bounding boxes and Eigen-CAM [23] visualizations of the CBAM fusion module. (d) and (f) are visualizations of (c) and (e), respectively.

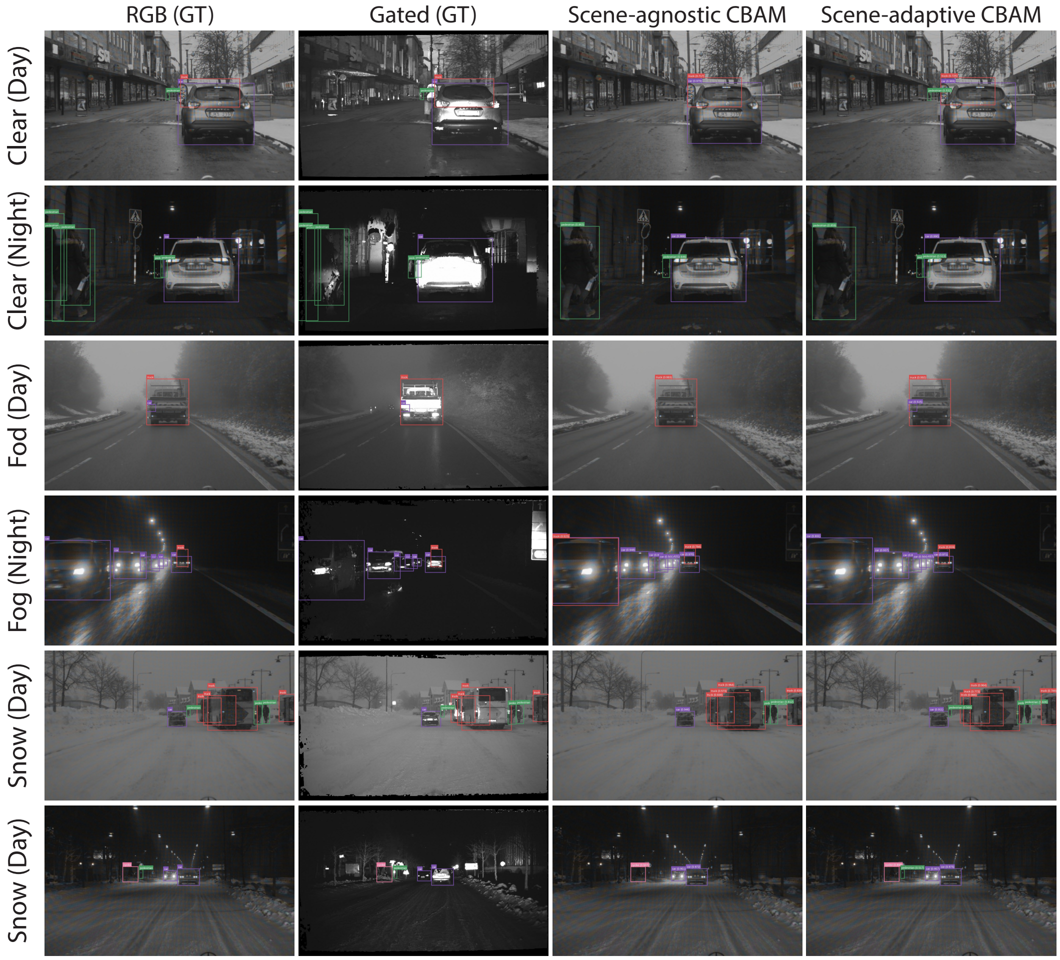


Figure 8.5: Object detection results on STF dataset in various scene conditions. From top to bottom: RGB images, gated images, scene-agnostic CBAM detections, and scene-adaptive CBAM detections. RGB and gated images are overlaid with ground truth bounding boxes.

Table 8.4: Object detection results (mAP@0.5) and speed (s) on M^{3}FD dataset. Due to the difference in scene splits between baselines and our models, only results on the *full* test set are comparable across all methods.

Method	Test Scene					Inference Speed (s)
	Day	Night	Overcast	Challenge	Full	
RGB only	71.59	91.06	81.55	80.03	77.79	0.016
Thermal only	65.68	89.17	79.66	76.39	74.64	0.016
U2F [36]	73.80	86.8	73.10	97.6	77.5	0.129†
TarDAL [20]	74.50	89.30	74.10	98.30	77.80	0.047†
EAEFNet [17]	78.30	89.50	78.60	97.90	80.10	—
Scene-Agnostic CBAM **(ours)**	74.53	**93.09**	84.11	81.06	80.46	0.028
Scene-Adaptive CBAM **(ours)**	**75.92**	92.55	**85.14**	**82.72**	**81.46**	0.032

† Includes image fusion and object detection inference time.

Table 8.5: Object detection results and speed (s) on FLIR Aligned dataset. AP@0.5 for each object category is reported.

Method	Person	Bicycle	Car	mAP@0.5	mAP@0.75	mAP[†]	Inference Speed (s)
RGB only	60.79	37.25	73.94	57.32	17.6	24.7	0.016
Thermal only	82.86	50.80	82.83	72.16	33.4	37.0	0.016
GAFF [40]	76.60	59.40	85.50	72.9	32.9	37.5	0.061
CFR_3[39]	74.49	57.77	84.91	72.93	—	—	0.050
RetinaNet + MFPT[42]	78.1	65.0	87.3	76.80	—	—	0.050
UA-CMDet[29]	83.20	64.30	88.40	78.60	—	—	—
CFT [25]	—	—	—	78.7	35.5	40.2	0.026
CSAA[3]	—	—	—	79.20	37.4	41.3	0.031
FasterRCNN + MFPT[42]	83.2	67.7	89.0	80.00	—	—	0.080
LRAF-Net[10]	—	—	—	80.50	—	42.8	—
Scene-agnostic CBAM **(ours)**	88.26	77.43	90.68	85.45	**43.3**	46.8	0.028
Scene-adaptive CBAM **(ours)**	**88.92**	**78.61**	**90.94**	**86.16**	43.0	**47.1**	0.032

[†] mAP refers to mAP@0.5:0.95

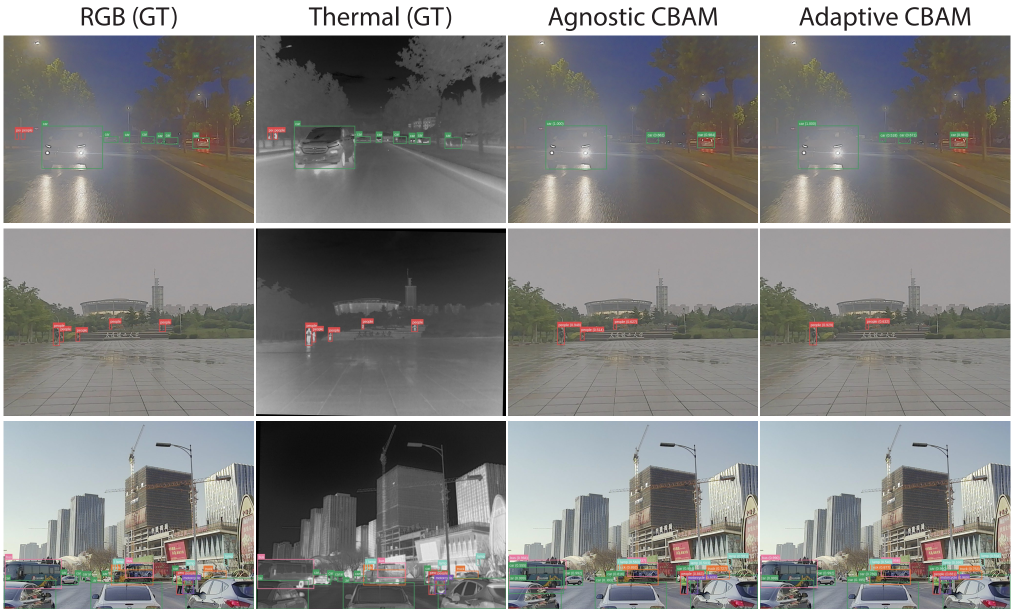
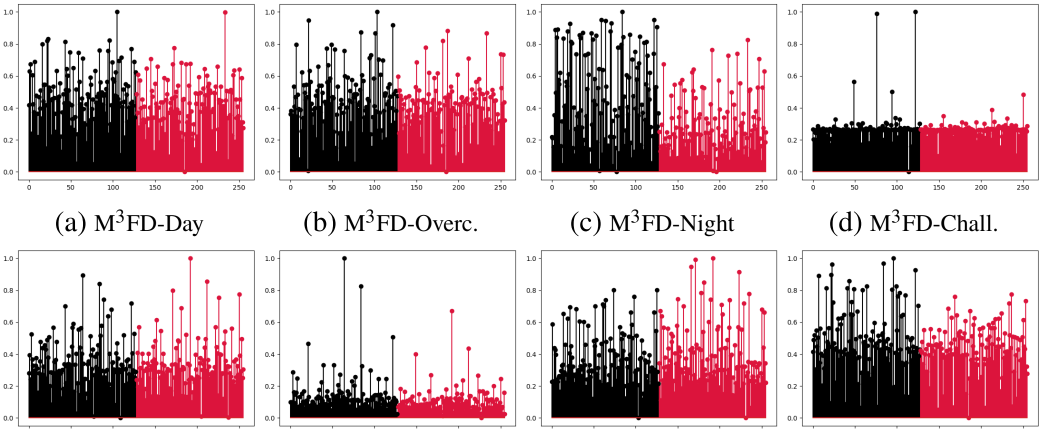


Figure 8.6: Example of failure cases on M^{3}FD dataset. Both models struggled with distant small objects in *night* and *overcast* images and cluttered objects in *day* images.

the M^{3}FD training set for better performance. We also show some failure cases on M^{3}FD in Fig. 8.6 where both fusion models struggled with distant small objects in *overcast* and *night* scenes, and cluttered objects under daylight.

For the FLIR Aligned dataset, we evaluate fusion networks built from an RGB network pretrained on COCO and a thermal network trained on the unaligned FLIR thermal training set. In general, both our scene-agnostic and scene-adaptive fusion



(a) M^{3}FD-Day (b) M^{3}FD-Overc. (c) M^{3}FD-Night (d) M^{3}FD-Chall.

(e) FLIR-Day (f) FLIR-Night (g) FLIR-Full (h) M^{3}FD-Full

Figure 8.7: Normalized attention weights for 256 feature channels in CBAM fusion module trained on different scenes. Thermal channels are in black, and RGB channels in crimson. The fusion module trained on the entire dataset (g-h) exhibits similar attention patterns across all scene/weather conditions, whereas from *day* to *overcast* to *night*, the scene-specific fusion module (a-f) attends increasingly on thermal features.

Table 8.6: Quantitative detection AP on the *clear* scene and unseen scenes for *car* following the KITTI evaluation [13] used in [1]. Models are all trained on the training set of the *clear* scene. Our scene-adaptive CBAM model is trained on *clear-day* and *clear-night* splits.

Method	Clear			Light Fog			Dense Fog			Snow/Rain		
	easy	mod.	hard	easy	mod.	hard	easy	mod.	hard	easy	mod.	hard
RGB only	90.14	87.56	80.87	91.19	88.47	82.02	90.43	85.59	80.79	89.44	82.87	77.81
Gated only	88.51	80.09	74.65	87.98	78.92	73.59	80.52	75.86	70.42	80.58	75.59	69.52
Fusion SSD [1]	87.73	78.02	69.49	88.33	78.65	76.54	74.07	68.46	63.23	85.49	75.28	67.48
Deep Fusion [1]	90.07	80.31	77.82	90.60	81.08	79.63	86.77	77.28	73.93	89.25	79.09	70.51
Deep Entropy Fusion [1]	89.84	85.57	79.46	90.54	87.99	84.90	87.68	81.49	76.69	88.99	83.71	77.85
Scene-agnostic CBAM **(ours)**	**90.33**	88.53	**81.16**	**91.43**	89.05	**84.94**	90.75	**88.66**	**82.07**	**89.99**	**86.57**	**79.79**
Scene-adaptive CBAM **(ours)**	90.29	**88.53**	81.07	91.13	**89.13**	84.20	**90.77**	88.37	81.68	89.96	86.30	79.74

Table 8.7: Quantitative detection AP on all scenes for *pedestrian*, *truck*, *car*, and *cyclist* following the KITTI evaluation [13] used in [1]. Models are trained on the training set of all scenes. The last column shows mAP@0.5 for all objects on all test images.

Method	Clear			Light Fog			Dense Fog			Snow/Rain			Full			All
	easy	mod.	hard	easy	mod.	hard	easy	mod.	hard	easy	mod.	hard	easy	mod.	hard	
RGB only	87.05	83.88	82.93	89.68	88.88	87.99	88.61	88.28	87.90	88.92	86.01	83.73	84.22	79.94	76.30	80.85
Gated only	81.69	76.19	74.57	85.63	84.01	80.19	83.40	82.00	79.88	84.03	79.54	77.38	80.70	73.58	70.13	75.15
Scene-agnostic CBAM **(ours)**	**88.65**	85.12	**84.25**	90.30	**89.68**	**88.95**	89.78	89.18	88.82	89.25	87.01	**85.77**	86.11	81.84	**78.52**	83.01
Scene-adaptive CBAM **(ours)**	88.60	**85.24**	84.22	**90.53**	89.39	88.89	**89.79**	**89.33**	**89.03**	**89.37**	**87.46**	85.69	**86.13**	**81.85**	78.48	**83.11**

models outperform the baselines by a large margin (Table 8.5), due to the increase in data the thermal and RGB networks had access to. Some qualitative detection results on FLIR test images along with attention visualizations are given in Fig. 8.4. We observe that scene-adaptive model tends to detect bicycles more successfully than scene-agnostic model, especially when the bicycle is rode by a person (see row 2 and 6 in Fig. 8.4). The higher margin of AP@0.5 for *bicycle* in Table 8.5 also aligns with this observation. In order to exam the effects of scene-adaptive CBAM, we visualize the CBAM using class activation map (CAM) [23] where the spatial attention is shown by a heat map. From the visualization, we can see there is generally no difference between scene-agnostic CBAM and scene-adaptive CBAM for day images. However, the spatial attention in scene-adaptive CBAM attend more on small areas.

We visualize the channel attention of the scene-specific fusion module by plotting the normalized attention weights of thermal (black) and RGB (crimson) features for various scenes in Fig. 8.7. Higher value implies CBAM attends more on that feature channel. We find that scene-agnostic CBAM exhibits similar channel attention patterns across all scenes, while scene-adaptive CBAM shows tailored attention patterns per scene. Moreover, we observe attention weight increases on thermal features compared with RGB features from day to overcast to night images, likely as RGB images contain less information under lower illumination.

For the STF dataset, we first follow [1] and train our fusion modules only on *clear-day* and *clear-night* RGB-gated image pairs for fair comparison. As shown in Table 8.6, the scene-agnostic and scene-adaptive CBAM models achieve similar performance on different scenes and outperform the baseline models using even more modalities than RGB-gated images [1]. When training on all scenes in Table 8.7, we can see that our scene-adaptive model outperforms the scene-agnostic model by 0.1% on mAP@0.5. Single-modality models used for this experiment are also further trained on STF training data, due to their use of 10 and 12 bit gated and RGB imagery. Fig. 8.5 presents a few examples of the qualitative detection results in various scenes.

Computational Benchmarks: We compiled our CBAM fusion models using TorchInductor and conducted benchmarks on a Titan RTX. The inference time for the scene-adaptive fusion model is 0.032 seconds per individual image pair, while the scene-agnostic variant clocks in at 0.028 seconds. These times are comparable with other recent multimodal object detection approaches (Table 8.4, 8.5) and meet the speed requirements for real-time autonomous driving applications.

Table 8.8: Ablation study on different fusion modules. Object detection results (mAP@0.5) on M^{3}FD dataset are reported.

Fusion Module	Train/Search Scene	Test Scene				
		Day	Night	Overcast	Challenge	Full
RGB only	Full	71.59	91.06	81.55	80.03	77.79
Thermal only		65.68	89.17	79.66	76.39	74.63
ECAAttn (Tr)		73.38	93.39	83.55	82.28	80.17
ECAAttn (RH)		72.25	92.83	81.98	80.53	78.81
ECAAttn (TH)		74.02	93.38	**84.25**	**81.48**	80.32
ShuffleAttn (Tr)		73.47	**94.56**	84.61	80.91	80.17
ShuffleAttn (RH)		72.78	92.63	83.61	80.37	79.28
ShuffleAttn (TH)		74.07	93.21	84.19	81.43	80.34
CBAM (Tr)		73.11	93.01	83.11	80.17	79.33
CBAM (RH)		72.85	92.46	83.54	80.73	79.21
CBAM (TH)		**74.53**	93.09	84.11	81.06	**80.46**
ECAAttn (TH)	Day	**74.75**	**94.51**	84.16	81.09	**80.65**
	Night	72.00	91.84	83.56	79.74	78.74
	Overcast	71.96	92.67	**84.44**	80.09	79.18
	Challenge	73.25	93.11	83.78	**81.88**	80.14
ShuffleAttn (TH)	Day	**75.28**	**94.64**	**84.72**	**81.85**	**81.04**
	Night	71.79	92.21	83.30	78.95	78.21
	Overcast	73.57	92.42	84.32	80.95	80.00
	Challenge	72.42	92.90	84.21	81.27	79.62
CBAM (TH)	Day	**76.04**	94.07	84.89	80.78	**81.07**
	Night	72.68	92.55	83.20	78.77	78.62
	Overcast	73.30	92.53	**85.15**	80.67	79.94
	Challenge	74.10	**94.28**	82.70	**82.61**	80.93
DSF-NAS	Day	**75.68**	**94.25**	**84.35**	**81.85**	**81.03**
	Night	72.32	91.94	83.85	80.51	79.12
	Overcast	73.15	93.46	83.79	80.60	79.52
	Challenge	72.90	93.44	83.29	81.59	79.81
	Full	74.68	92.65	83.90	81.67	80.56

Tr – Trained head TH – Thermal head RH – RGB head

Table 8.9: Object detection results (mAP@0.5) of our scene-adaptive CBAM model trained using decreasing amounts of data.

Dataset	% of Original Training Set			
	100%	50%	25%	1%
FLIR	86.16	85.70	84.60	75.72
M^{3}FD	81.46	78.34	77.65	41.94
STF-Clear	80.65	80.73	80.10	73.76
STF-Full	83.11	83.06	82.99	75.64

Ablation Studies

Fusion Module Design: We conduct an ablation study using the M^{3}FD dataset to explore the effects of different fusion modules and architectures (Table 8.8). We compare our CBAM-based RGB-X fusion approach against two other attention modules: ECAAttn [33, 6] and ShuffleAttn [41]. Furthermore, we also compare against custom fusion modules (DSF-NAS) designed purposely for this fusion task via neural architecture search. In particular, we use Bilevel Multimodal Neural Architecture Search [37] (BM-NAS) to automate this design as its gradient-based optimization approach makes it faster compared to other NAS methods based on reinforcement learning and genetic algorithms. Specifically, we allow BM-NAS to optimize over sequential applications of two operations chosen from sum, spatial attention, channel attentions from CBAM and ECAAttn, and 2D convolution of concatenated features.

We first train for fusion using scene-agnostic CBAM, ECAAttn, and ShuffleAttn modules along with either a trainable, frozen thermal, or frozen RGB detector head. We find that training with a frozen detector head initialized with thermal weights performs the best in Table 8.8, possibly due to the lower variance of thermal data across different scenes. We repeat the study under the scene-adaptive regime, with the previous three attention modules and frozen thermal detection heads, along with DSF-NAS fusion modules. Overall, we find similar performance between DSF-NAS and CBAM-based fusion networks. However, CBAM fusion models exhibit better performance on scene-specific data verifying its use in our proposed modular framework.

Effect of Training Dataset Size on Fusion: As our proposed fusion method looks to fuse pretrained networks with lightweight fusion modules, the fusion process should still be effective and be able to generalize even when done with limited

Table 8.10: Object detection results (mAP@0.5) of our scene-adaptive CBAM model on unknown scenes in M^{3}FD dataset.

Test	Excluded Training Scene			
	Day	Night	Overcast	Challenge
Excluded Scene	73.17	93.50	84.18	80.74
All Scenes	80.48	81.30	81.27	80.98

amounts of training data. To determine the extent of this, we perform fusion using 100%, 50%, 25%, and 1% of the original datasets in Table 8.9. Overall, we find that competitive results can still be achieved using only 25% of the original training data results, with the exception of M^{3}FD which decays quicker than the rest.

Performance on Unknown Scenes: As our proposed method requires scenes to be known during training, we further investigate the performance of our method on unknown/unexpected scenes. In this experiment, our scene-specific CBAM fusion modules and scene classifiers are trained with one scene data excluded, and tested on that excluded scene and all test images. We observed minor regression in overall performance (row 2 in Table 8.10) compared with our scene-adaptive model trained on all scenes (81.46 in Table 8.4), which is expected as there is no fusion module trained specifically for that unknown scene. However, the overall mAP@0.5 in all cases is still higher than scene-agnostic model trained on all scenes (80.46 in Table 8.4). Specifically, in the case of *night* or *overcast* scene excluded, the object detection performance on the unknown scene (row 1 in Table 8.10) is higher than the scene-agnostic model. This is possibly because our scene classifier tends to select a fusion module trained on a similar scene, for instance, classifying *night* image as *overcast* and vice versa.

8.6 Limitations and Future Work

Our method requires aligned RGB-X data, which is not always available. The scene-specific modules require scenes to be known during training, and the approach is not expected to work as well in unexpected weather conditions. Future work looks to incorporate unsupervised [11] and online learning [15] to adapt to unexpected conditions.

8.7 Conclusion

We presented a novel RGB-X object detection model that improves autonomous vehicle perception in different weather and lighting conditions. We showed that our method is superior compared to existing works on two RGB-T and one RGB-gated object detection benchmarks, demonstrating the robustness of our scene-adaptive models and generalizability to different modalities. Furthermore, our use of lightweight fusion modules brings us closer to achieving a more modular design for deep sensor fusion. For future work, we look to train and leverage larger pretrained models for both RGB and thermal modalities via multitask learning and to incorporate into an online learning framework to adapt to unexpected weather patterns.

References

[1] M. Bijelic, T. Gruber, F. Mannan, F. Kraus, W. Ritter, K. Dietmayer, and F. Heide. “Seeing Through Fog Without Seeing Fog: Deep Multimodal Sensor Fusion in Unseen Adverse Weather”. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[2] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. “nuscenes: A multimodal dataset for autonomous driving”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2020, pp. 11621–11631.

[3] Y. Cao, J. Bin, J. Hamari, E. Blasch, and Z. Liu. “Multimodal Object Detection by Channel Switching and Spatial Attention”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 403–411.

[4] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. “End-to-end object detection with transformers”. In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer. 2020, pp. 213–229.

[5] Y.-T. Chen, J. Shi, Z. Ye, C. Mertz, D. Ramanan, and S. Kong. “Multimodal object detection via probabilistic ensembling”. In: *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX*. Springer. 2022, pp. 139–158.

[6] S. A. Deevi and D. Mishra. “Expeditious Object Pose Estimation for Autonomous Robotic Grasping”. In: *International Conference on Computer Vision and Image Processing*. Springer. 2022, pp. 15–30.

[7] J. Fayyad, M. A. Jaradat, D. Gruyer, and H. Najjaran. “Deep Learning Sensor Fusion for Autonomous Vehicle Perception and Localization: A Review”. en. In: *Sensors* 20.15 (2020), p. 4220. ISSN: 1424-8220. DOI: 10.3390/s20154220.

[8] D. Feng, C. Haase-Schütz, L. Rosenbaum, H. Hertlein, C. Glaeser, F. Timm, W. Wiesbeck, and K. Dietmayer. “Deep Multi-modal Object Detection and

Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges". In: *IEEE Transactions on Intelligent Transportation Systems* 22.3 (2021), pp. 1341–1360. ISSN: 1524-9050, 1558-0016. DOI: 10.1109/TITS.2020.2972974.

[9] *Teledyne FLIR ADAS Dataset*. https://www.flir.com/oem/adas/adas-dataset-form/, Last accessed on 2023-10-27.

[10] H. Fu, S. Wang, P. Duan, C. Xiao, R. Dian, S. Li, and Z. Li. "LRAF-Net: Long-Range Attention Fusion Network for Visible–Infrared Object Detection". In: *IEEE Transactions on Neural Networks and Learning Systems* (2023).

[11] L. Gan, C. Lee, and S.-J. Chung. "Unsupervised rgb-to-thermal domain adaptation via multi-domain attention network". In: *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE. 2023, pp. 6014–6020.

[12] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. "Vision meets robotics: The kitti dataset". In: *The International Journal of Robotics Research* 32.11 (2013), pp. 1231–1237.

[13] A. Geiger, P. Lenz, and R. Urtasun. "Are we ready for autonomous driving? the kitti vision benchmark suite". In: *2012 IEEE conference on computer vision and pattern recognition*. IEEE. 2012, pp. 3354–3361.

[14] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, et al. "Searching for mobilenetv3". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 1314–1324.

[15] C. Lee, J. G. Frennert, L. Gan, M. Anderson, and S.-J. Chung. "Online Self-Supervised Thermal Water Segmentation for Aerial Vehicles". In: *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2023, pp. 7734–7741.

[16] M. Liang, B. Yang, S. Wang, and R. Urtasun. "Deep continuous fusion for multi-sensor 3d object detection". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 641–656.

[17] M. Liang, J. Hu, C. Bao, H. Feng, F. Deng, and T. L. Lam. "Explicit Attention-Enhanced Fusion for RGB-Thermal Perception Tasks". In: *IEEE Robotics and Automation Letters* (2023).

[18] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. "Microsoft COCO: Common Objects in Context". In: *Proceedings of the European Conference on Computer Vision*. Ed. by D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars. Cham: Springer International Publishing, 2014, pp. 740–755. ISBN: 978-3-319-10602-1.

[19] J. Liu, S. Zhang, S. Wang, and D. Metaxas. “Multispectral Deep Neural Networks for Pedestrian Detection”. en. In: *Procedings of the British Machine Vision Conference 2016*. York, UK: British Machine Vision Association, 2016, pp. 73.1–73.13. ISBN: 978-1-901725-59-9. DOI: 10.5244/C.30.73.

[20] J. Liu, X. Fan, Z. Huang, G. Wu, R. Liu, W. Zhong, and Z. Luo. “Target-aware Dual Adversarial Learning and a Multi-scenario Multi-Modality Benchmark to Fuse Infrared and Visible for Object Detection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2022, pp. 5802–5811.

[21] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. “Ssd: Single shot multibox detector”. In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer. 2016, pp. 21–37.

[22] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. “Swin transformer: Hierarchical vision transformer using shifted windows”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2021, pp. 10012–10022.

[23] M. B. Muhammad and M. Yeasin. “Eigen-cam: Class activation map using principal components”. In: *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2020, pp. 1–7.

[24] D. Patterson, J. Gonzalez, Q. Le, C. Liang, L.-M. Munguia, D. Rothchild, D. So, M. Texier, and J. Dean. “Carbon emissions and large neural network training”. In: *arXiv preprint arXiv:2104.10350* (2021).

[25] F. Qingyun, H. Dapeng, and W. Zhaokui. “Cross-modality fusion transformer for multispectral object detection”. In: *arXiv preprint arXiv:2111.00273* (2021).

[26] J. Redmon and A. Farhadi. “Yolov3: An incremental improvement”. In: *arXiv preprint arXiv:1804.02767* (2018).

[27] S. Ren, K. He, R. Girshick, and J. Sun. “Faster r-cnn: Towards real-time object detection with region proposal networks”. In: *Advances in neural information processing systems* 28 (2015).

[28] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, et al. “Scalability in perception for autonomous driving: Waymo open dataset”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2020, pp. 2446–2454.

[29] Y. Sun, B. Cao, P. Zhu, and Q. Hu. “Drone-based RGB-infrared cross-modality vehicle detection via uncertainty-aware learning”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 32.10 (2022), pp. 6700–6713.

[30] M. Tan and Q. Le. “Efficientnet: Rethinking model scaling for convolutional neural networks”. In: *International conference on machine learning*. PMLR. 2019, pp. 6105–6114.

[31] M. Tan, R. Pang, and Q. V. Le. “Efficientdet: Scalable and efficient object detection”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 10781–10790.

[32] J. Wagner, V. Fischer, M. Herman, and S. Behnke. “Multispectral Pedestrian Detection using Deep Fusion Convolutional Neural Networks”. In: *ESANN*. 2016.

[33] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu. “ECA-Net: Efficient channel attention for deep convolutional neural networks”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 11534–11542.

[34] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon. “CBAM: Convolutional block attention module”. In: *Proceedings of the European Conference on Computer Vision*. 2018, pp. 3–19.

[35] D. Xu, D. Anguelov, and A. Jain. “PointFusion: Deep Sensor Fusion for 3D Bounding Box Estimation”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 244–253. DOI: 10.1109/CVPR.2018.00033.

[36] H. Xu, J. Ma, J. Jiang, X. Guo, and H. Ling. “U2Fusion: A unified unsupervised image fusion network”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.1 (2020), pp. 502–518.

[37] Y. Yin, S. Huang, and X. Zhang. “Bm-nas: Bilevel multimodal neural architecture search”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 8. 2022, pp. 8901–8909.

[38] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell. “BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2020.

[39] H. Zhang, E. Fromont, S. Lefevre, and B. Avignon. “Multispectral fusion for object detection with cyclic fuse-and-refine blocks”. In: *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2020, pp. 276–280.

[40] H. Zhang, E. Fromont, S. Lefèvre, and B. Avignon. “Guided attentive feature fusion for multispectral pedestrian detection”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2021, pp. 72–80.

[41] Q.-L. Zhang and Y.-B. Yang. “Sa-net: Shuffle attention for deep convolutional neural networks”. In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2021, pp. 2235–2239.

[42] Y. Zhu, X. Sun, M. Wang, and H. Huang. “Multi-Modal Feature Pyramid Transformer for RGB-Infrared Object Detection”. In: *IEEE Transactions on Intelligent Transportation Systems* (2023).

[43] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye. “Object detection in 20 years: A survey”. In: *Proceedings of the IEEE* (2023).

Chapter 9

CONCLUSION

This thesis addressed critical challenges in learning-based perception for robotics. Although current computer vision models and approaches can serve as powerful visual front-ends for robots to visually comprehend their surrounding world, most common and popular off-the-shelf methods are based on supervised deep learning which ideally has access to abundant training data in order to perform well during deployment time. For robotic applications, however, this data landscape is not always ideal and bespoke solutions are often required. In this thesis, we confronted problems in robotic perception characterized by challenges such as data scarcity and the lack of apparent signals for supervised model training.

9.1 Visual Terrain Relative Navigation

In Chapter 2, we demonstrated the power of self-supervised approaches to improve existing visual terrain-relative navigation for high-altitude aerial vehicles in the presence of seasonal and other temporal variations. Despite the abundance of relevant high-resolution aerial imagery, the absence of obvious supervisory signals for the high-level navigation objective posed the primary challenge. To address this, we presented a CNN-based deep transform that can inject seasonal robustness into existing image registration-based VTRN pipelines.

Notably, we found that a deep transform optimized for area-based matching is highly effective against significant seasonal content variations. On the other hand, a deep transform optimized for feature-based matching is less robust but can reliably identify regions where registration is impractical and reduce false positive matches. Above all, we demonstrated how targeted integration of deep learning into classical visual perception systems can effectively handle edge cases that would otherwise compromise such systems.

In Chapter 3, we looked to reduce the storage demands required by image registration-driven VTRN methods by introducing a landmark-based approach for localization. Specifically, we focused on the problem of optimal landmark discovery and proposed a data-driven method, thus circumventing the need for human guidance. This enables scalability to larger geographic areas while removing human cognitive and visual biases, which could potentially lead to suboptimal outcomes in non-anthropomorphic regions. To accomplish this, we showed how reframing a subjective question—*what is a useful landmark for aerial navigation in this image?*—into a binary one enables us to formulate a simple, self-supervised contrastive loss function for a deep learning model whose network activations, after training, highlight salient image regions where such useful landmarks reside.

9.2 Thermal perception for aerial field robots

Next, we solved the problem of developing semantic perception algorithms for aerial robotics in littoral field settings where relevant dataset coverage is lacking. Our proposed strategies reflect the various states of the constantly, evolving data landscape of typical field robotic projects.

Chapter 4 marked the beginning of this project, a stage in which field roboticist have little to no data, and possibly just enough for model evaluation. Here, we developed a water segmentation algorithm to enable nighttime flights along coastlines and rivers, as well as to support downstream scientific studies such as bathymetry. Under tight data constraints, we developed an online, self-supervised algorithm that adapts to thermal data in real-time by taking advantage of visual water cues and inertial sensors. Furthermore, we devised a deployment methodology that permits online training and inference at 10 Hz. While our method is specific to water segmentation, we note that our overall online approach and deployment strategy offers a versatile framework applicable to other perception objectives.

By Chapter 5, our data landscape progressed such that we have enough diverse thermal data for model training but do not yet have annotations to enable supervised training. As such, we developed an unsupervised domain adaptation algorithm to train thermal perception models by harnessing large, annotated RGB datasets in conjunction with our unlabeled thermal data. Like prior works, we aimed to perform domain adaptation by achieving domain confusion. However, we proposed the use of domain-specific attention modules that prevent the forceful alignment of difficult-to-align, modality-specific features, while enabling the rest of the network to produce easily-generalizable, domain-invariant features. We demonstrated that our method outperforms other works in classification benchmarks before demonstrating its versatility by applying it towards semantic segmentation in aerial field settings.

In Chapter 6, we introduced the Caltech Aerial RGB-Thermal Dataset, compiled from collection efforts spanning the project's duration. We presented our robotic platform, our thermal camera calibration process, and the challenges of annotating thermal imagery for semantic segmentation. Along with the dataset, we provided new benchmarks for evaluating thermal and RGB-T semantic segmentation models, RGB-T image translation, and thermal motion tracking. We placed emphasis on real-time performance, and highlighted challenges due to geographic and temporal distribution shifts. In contrast to prior benchmarks, ours focus on natural field settings and provide unique challenges due to less structured surroundings and drastic lighting changes. Furthermore, we analyzed the effectiveness of visual foundation models in the thermal domain. We found that while most models struggle with the modality shift, the Segment Anything Model shows promise when paired with ground truth semantics.

In Chapter 7, we presented our method to automatically generate semantic segmentation labels for aerial thermal images captured from robotic platforms. Our method generates labels by warping land use land cover data from satellite imagery into the thermal camera frame, using knowledge of the local and global pose of the robot. We showed that this process can generate highly-precise segmentation labels from freely-available low-resolution land cover data by applying pre- and postprocessing refinement steps via dense CRFs and the Segment Anything model, respectively. This allows us keep annotation costs low by eliminating the need for costly, high-resolution land cover data.

9.3 RGB-thermal deep sensor fusion for autonomous driving

In Chapter 8, we turned our attention towards RGB-T deep sensor fusion perception algorithms for autonomous driving. This setting also faces a lack of data due to the need for custom RGB-T datasets that are synchronized and coregistered. To address this, we presented a data-efficient, deep sensor fusion method for object detection that can take full advantage of pretrained, single-modality networks. Our method makes fusion training much faster than existing end-to-end methods while surpassing or maintaining performance on various benchmarks. This moves us closer towards a modular, drag-and-drop paradigm that could enable faster real-world deployments after sensor re-configurations.