UNIVERSITÄT BONN

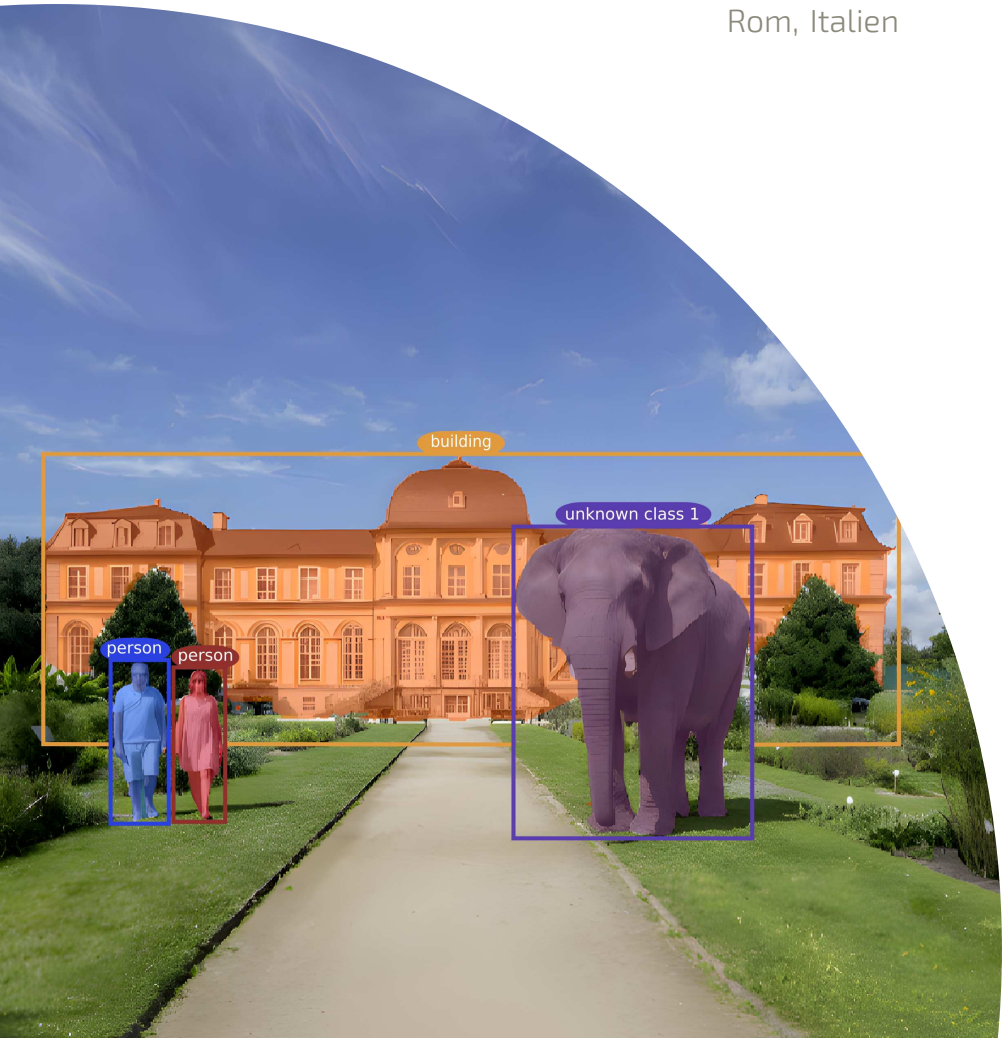
Dissertation
zur Erlangung des Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)
Agrar-, Ernährungs- und Ingenieurwissenschaftliche Fakultät
der Rheinischen Friedrich-Wilhelms-Universität Bonn
Institut für Geodäsie und Geoinformation

From Closed- to Open-World Panoptic Segmentation

von

Matteo Sodano

aus
Rom, Italien



Bonn 2026

Referent:
Prof. Dr. Cyrill Stachniss, University of Bonn, Germany

Korreferent:
Prof. Dr. Vikram S. Adve, University of Illinois Urbana-Champaign, USA

Tag der mündlichen Prüfung: 02.03.2026

Angefertigt mit Genehmigung der Agrar-, Ernährungs- und Ingenieurwissenschaftlichen Fakultät der Universität Bonn

The cover image was generated using generative artificial intelligence models.

Zusammenfassung

Autonome Robotertechnologien verändern die Industrie zunehmend, indem sie es intelligenten Maschinen ermöglichen, komplexe Aufgaben mit minimaler menschlicher Unterstützung auszuführen und so die Effizienz, Sicherheit und Skalierbarkeit zu steigern. Roboter können Aufgaben übernehmen, die für Menschen gefährlich, repetitiv oder einfach unpraktisch sind, und ebnen so den Weg für effizientere und zuverlässigere Arbeitsabläufe. Aus diesem Grund gewinnen Roboter in verschiedenen Anwendungsbereichen zunehmend an Bedeutung, von der Landwirtschaft und Fertigung bis hin zur Erkundung unstrukturierter Umgebungen und nicht zuletzt dem autonomen Fahren. Trotz der Vielfalt dieser Bereiche haben alle autonomen Systeme eine grundlegende Anforderung gemeinsam: Sie müssen ihre Umgebung wahrnehmen und verstehen können, bevor sie sich darin bewegen und mit ihr interagieren können. Beispielsweise müssen Unkrautbekämpfungsroboter Nutzpflanzen von Unkraut unterscheiden, um ihre Aufgaben präzise auszuführen, Kochroboter müssen die für ein Rezept erforderlichen Utensilien finden und autonome Fahrzeuge müssen andere Verkehrsteilnehmer erkennen, um eine sichere und zuverlässige Fahrt zu gewährleisten.

Die Perzeption ist daher ein wichtiger Bestandteil jedes autonom agierenden Systems. Roboter können Menschen nur dann unterstützen und effizient arbeiten, wenn sie in der Lage sind, ihre Umgebung zuverlässig zu interpretieren und zu verstehen. In den letzten Jahrzehnten haben wir enorme Fortschritte bei der Entwicklung von Perzeptionsalgorithmen erlebt, die vor allem durch Fortschritte im maschinellen Lernen und Deep Learning vorangetrieben wurden. Unter diesen haben sich Segmentierungsansätze für die Szeneninterpretation als zentrale Aufgabe der modernen Forschung zur Roboterperzeption herauskristallisiert, die es Systemen ermöglichen, jedem wahrgenommen Teil der Umgebung aussagekräftige semantische und instanzbezogene Bezeichnungen zuzuweisen. Semantische Informationen sind wichtig, um zu verstehen, was sich in der Szene befindet, und alle Kategorien zu erkennen, die in der Umgebung vorkommen. Instanzinformationen hingegen konzentrieren sich auf die Unterscheidung zwischen einzelnen Objekten. Diese Informationen sind wichtig für Roboter, die in der realen Welt agieren

und mit ihr interagieren müssen, da die Erkennung einzelner Objekte für alle Arten von nachgelagerten Aufgaben wie Navigation, Manipulation und mehr von entscheidender Bedeutung ist.

Die wichtigsten Beiträge dieser Arbeit sind Techniken für die Roboterperzeption, die das Verständnis von Szenen über mehrere Domänen, Sensoren und Modalitäten hinweg verbessern. Insbesondere konzentrieren wir uns auf die sogenannte panoptische Segmentierung, welche das oben beschriebene semantische Verständnis bezüglich semantischer Kategorien und Instanzen vereint. In dieser Arbeit befassen wir uns mit mehreren zentralen Herausforderungen für die panoptische Segmentierung. Zunächst gehen wir diese Aufgabe mit RGB-D Sensoren an und entwickeln einen Algorithmus, der die komplementären visuellen Informationen nutzt, um die Segmentierungsgüte zu verbessern, während er gleichzeitig robust gegenüber fehlenden RGB- oder Tiefeninformationen bleibt. Anschließend untersuchen wir, wie die zugrunde liegende Hierarchie von Objekten in der Szene genutzt werden kann, um die Segmentierungsgüte zu verbessern. Schließlich gehen wir über die klassische Einschränkung der panoptischen Segmentierung hinaus, die den meisten bestehenden Ansätzen gemeinsam ist, nämlich die sogenannte Annahme einer abgeschlossenen Welt. Diese Annahme beschränkt Perzeptionsansätze auf einen festen, vordefinierten Satz von Objektkategorien, was in realen Szenarien, in denen Roboter unweigerlich auf zuvor unbekannte Objekte stoßen, unrealistisch ist. Zunächst schlagen wir einen Datensatz für ein genaues und reproduzierbares Benchmarking von Segmentierungsmethoden für offene Welten vor. Anschließend entwickeln wir einen Algorithmus für die semantische Segmentierung offener Welten, mit dem wir zum Testzeitpunkt neue semantische Kategorien entdecken wollen, die während des Trainings nie aufgetreten sind. Schließlich widmen wir uns der Aufgabe der panoptischen Segmentierung offener Welten, mit dem Ziel, sowohl neue semantische Kategorien als auch neue Objektinstanzen zu entdecken.

Zusammenfassend lässt sich sagen, dass diese Arbeit mehrere neuartige Methoden und Datensätze für die panoptische Segmentierung in geschlossenen und offenen Welten vorschlägt und einen Beitrag zum aktuellen Stand der Technik in den Bereichen Roboterperzeption und Szenenverständnis leistet. Die vorgeschlagenen Methoden wurden anhand öffentlich zugänglicher Datensätze rigoros evaluiert und in Fachzeitschriften und auf Konferenzen veröffentlicht. Darüber hinaus wurden alle Implementierungen unserer entwickelten Ansätze als Open Source veröffentlicht, um die weitere Forschung und Entwicklung in diesem Bereich zu erleichtern.

Abstract

AUTONOMOUS robotic technologies are increasingly transforming industries by enabling intelligent machines to perform complex tasks with minimal human intervention, increasing efficiency, safety, and scalability. Robots can take over tasks that are dangerous, repetitive, or simply impractical for humans, paving the way for more efficient and reliable workflows. For this reason, robots are becoming more and more important in various applications, ranging from agriculture and manufacturing to exploration in unstructured environments and, last but not least, autonomous driving. Despite the diversity of these domains, all autonomous systems share a fundamental requirement: they must be able to perceive and understand their environment before navigating in and interacting with it. For example, weeding robots need to distinguish crops from weeds to perform their tasks accurately, cooking robots must locate the tools required for a recipe, and autonomous vehicles must recognize other traffic participants to ensure safe and reliable navigation on the road.

Perception is therefore a key building block of any autonomously acting system. Robots can only support humans and operate efficiently if they are able to reliably interpret and understand the environment in which they operate. Over the past decades, we have witnessed tremendous progress in the development of perception algorithms, largely driven by advances in machine learning and deep learning. Among these, segmentation approaches for scene interpretation have emerged as a central task of modern robotic perception research, allowing systems to assign meaningful semantic and instance-level labels to every part of the environment. Semantic information is important to understand what is in the scene and recognize all categories that appear in the environment. Instance information, in contrast, focuses on distinguishing among individual objects. These pieces of information are important for robots that act in the real world and need to interact with it, as recognizing an individual object is crucial for all kinds of downstream tasks, like navigation, manipulation, and more.

The main contributions of this thesis are techniques for robotic perception that enhances scene understanding across multiple domains, sensors, and modalities. In particular, we focus on panoptic segmentation. Panoptic segmentation

unifies the semantic and instance-level understanding described above, providing both information simultaneously. In this thesis, we address several key challenges for panoptic segmentation. First, we address this task using RGB-D sensors, developing an algorithm that exploits the complementary visual cues to enhance segmentation performance, while remaining robust to either missing RGB or depth input. Then, we investigate how the underlying hierarchy among objects in the scene can be exploited to improve segmentation performance. Finally, we move beyond the classic limitation of panoptic segmentation, shared by most existing state-of-the-art pipelines, namely the so-called closed-world assumption. This assumption constrains perception models to a fixed, predefined set of object categories, which is unrealistic in real-world scenarios where robots will inevitably encounter previously unseen objects. First, we propose a dataset for accurate and reproducible benchmarking of open-world segmentation methods. Then, we develop an algorithm for open-world semantic segmentation, where we aim to discover, at test time, novel semantic categories that never appeared during training. Finally, we tackle the task of open-world panoptic segmentation, aiming to discover both, novel semantic categories and novel object instances.

In summary, this thesis proposes several novel methods and datasets for closed-world and open-world panoptic segmentation, and contributes to the state of the art of robotic perception and scene understanding. The proposed methods have been rigorously evaluated on publicly available datasets and have been published in peer-reviewed journals and conferences. Furthermore, all software implementations are released as open-source to facilitate further research and development in the field.

Acknowledgements

LOOKING back on these past five years, it takes a moment to let everything settle in and truly realize that this PhD journey has reached its end. Like many things in life, a PhD is indeed a journey, and it demands motivation, resilience, stubbornness, and support. While some of these must be found within ourselves, support comes from the people around us: those we are privileged enough to choose, and those we are lucky enough to encounter along the way. Here, I want to thank all the people who made this journey not only possible, but meaningful, enriching, and unforgettable.

First and foremost, I want to express my heartfelt gratitude to my advisor, Cyrill Stachniss. I am deeply thankful for the opportunity he gave me to join his lab five years ago. Coming from a different academic background, I often found myself questioning my abilities and the path I had chosen. Cyrill guided me through this transition with patience and support, standing by me through both challenging and successful times. I am truly grateful for this, and for the honest, straightforward relationship we built over the years.

Second, I would like to thank my postdoc and, above all, my officemate, Jens Behley. I am deeply grateful not only for the constant support he has given me over the years, but also for the genuinely pleasant atmosphere we shared in the office. Our countless conversations, ranging from research to everyday life, from professional matters to personal stories, whether in the office or while traveling for conferences and work, are memories I truly cherish. As I leave the office behind, I do so firmly convinced that I ultimately won the long-standing battle over keeping the blinds open to let the sunlight in.

As I mentioned before, a PhD is a journey, and along the way, many people enter and leave our lives. I want to dedicate a few words to those who became part of mine during these years, and who, I am certain, will remain in it for many more to come.

To Elias and Federico, you were my very first friends here in Bonn, and you truly have no idea how important you have been throughout this entire journey. I will always cherish our time together: the highs and lows, the mutual support, the swimming sessions, and the countless coffee breaks that made the days brighter.

I also feel incredibly lucky and privileged to call many of the people I shared the office with not just colleagues, but friends. Louis, Rodrigo, Lucas, Gianmarco, Luca, and Meher, you made every day lighter and every challenge easier. The 24-year-old me would have never imagined that, five years later, he would be leaving Germany with so many friends spread across three continents. I am deeply grateful for the time we shared and look forward to meeting you again, wherever in the world our paths may cross.

Of course, just a few people do not make a whole lab. Over these five years, I have had the pleasure of crossing paths with so many amazing people I have learned from, shared laughs with, and truly enjoyed spending time alongside: Niklas Trekel, Benedikt Mersch, Saurabh Gupta, Yue Pan, Haofei Kuang, Starry Zhong, Linn Chong, Jan Weyler, Garvita Allabadi, Tiziano Guadagnino, Ignacio Vizzo, Rhiney Chen, Liren Jin, and Julius Rückin. Thank you all for creating such a welcoming and inspiring atmosphere in the lab. I would also like to extend my heartfelt thanks to Thomas Läbe, Birgit Klein, Kirsten Sadler, and Perrine Aguiar for their invaluable help and support, whether with technical challenges or bureaucratic hurdles. We are truly lucky to have you.

Bonn also brought into my life four remarkable women who taught me a great deal about life and about myself. Lidón, you have been a wonderful friend and the best swimming partner I could have wished for. Silvia, you cared for me like a younger brother and have always been someone I look up to and learn from. Nicky, despite our initial differences, you were the first to believe in my abilities and have always stood by me, something I will never forget. Irene, you helped me understand myself more deeply than anyone else ever has; the moments we shared will always hold a special place in my heart.

My PhD journey would not have been possible without my friends from Italy. I cannot name you all, but I want to especially mention Serena, Alessandro, and Tiziano, whose love and support I have felt unfailingly throughout these five years. No matter how much time passes between our meetings, we are always there for one another, and for that, I am endlessly grateful.

Lastly, I want to thank my parents and my brother for encouraging me every single day of my life. Their love and support are a true blessing, and I cannot express how much it means to have such a caring and steadfast family by my side.

Contents

Chapter 1

Introduction

Since the earliest myths and stories, humans have imagined artificial beings capable of acting on their own. This timeless dream of autonomous helpers supporting humans in daily tasks dates back as far as ancient Greece, where Hephaestus, the god of blacksmiths, was said to have built mechanical servants, among which there was Talos, the bronze giant who protected Crete [3]. In the Middle Ages, Leonardo da Vinci sketched a mechanical knight designed to mimic human motions and behaviors [39]. In modern times, science fiction has always been fascinated with autonomous machines either resembling humans, like in Fritz Lang's *Metropolis*, or supporting them, like in George Lucas's *Star Wars*. The dream of building autonomous machines has been a recurring theme across cultures and centuries. These visions reveal not only our fascination with technology but also our will to delegate tasks, enhance capabilities, and transform the way we live.

Over the past century, imagination has been gradually made reality. The industrial revolution introduced mechanical automata and programmable machines, transforming what was once only a mythological vision into something that was possible. "Shakey the Robot" [146] was the first successful and fully operational system that demonstrated that robots could combine sensing, reasoning, and action, and paved the way for modern robotics research. Gold standard algorithms such as A* [69] have been developed for Shakey. Today, robotics has moved far beyond laboratories: industrial manipulators, drones, autonomous vehicles, and household robots are increasingly present in our daily lives.

A central requirement for any autonomous system is the ability to perceive and interpret its surroundings. We as humans are often oblivious to the capability of our brain to sense, interpret, and understand the world around us. We have an almost innate capability to scan the surroundings and navigate the world. Autonomous systems, in contrast, do not possess this ability and need to be taught to interpret the surrounding environment in order to be made truly autonomous.

Without perception, robots would remain blind automata, incapable of adapting to the diversity and unpredictability of real-world environments. Thus, it is crucial that these systems are able to understand their surroundings in order to operate safely and robustly. Within the broad field of perception, scene interpretation and understanding play a pivotal role. Robots need to be able to not only know what is in the environment, but also understand what and where things are to interact with them.

Over the past decades, perception algorithms have evolved from using handcrafted features and rule-based systems [119, 197] to be much more adaptive approaches [72, 92, 179]. A decisive turning point came with the rise of machine learning and, more recently, deep learning, which revolutionized computer vision and perception. Neural networks have become the dominant paradigm, enabling machines to learn directly from raw sensory input and to generalize across complex, real-world scenarios [8, 206]. This shift has been fundamental in establishing robust perception pipelines, making perception-oriented tasks not only feasible but also highly accurate [31, 89].

Among these, segmentation approaches have emerged as a cornerstone of modern perception research. Semantic segmentation provides category-level understanding, instance segmentation distinguishes individual objects, and panoptic segmentation unifies both into a holistic representation of the scene [88]. This unified view is particularly important for robots that must interact with and reason about their surroundings, enabling them to both recognize what is present and determine how different elements of the environment relate to each other. Applications range from autonomous vehicles navigating dense urban traffic [30] to service robots assisting in human environments [40], where accurate scene understanding is critical for safe and efficient operation.

The task of panoptic segmentation [88] is defined independently of the specific sensing modality, making it applicable across a wide range of input data. Autonomous systems typically rely on cameras, which provide either color (RGB) or combined color and depth (RGB-D) information, as well as light detection and ranging (LiDAR) sensors, which deliver accurate 3D point clouds. In practice, panoptic segmentation can be applied to any of these modalities, supporting diverse environments and robotic platforms.

Semantic segmentation, and consequently panoptic segmentation, operates with a closed set of categories, meaning that the classes to be recognized are pre-defined during the training stage of the neural network [64]. This provides a clear and structured representation of the environment within the chosen taxonomy of classes, and ensures that systems can reliably parse their surroundings according to the available categories. Yet, under this formulation, any element that does not belong to the pre-defined set will be forced into one of those known

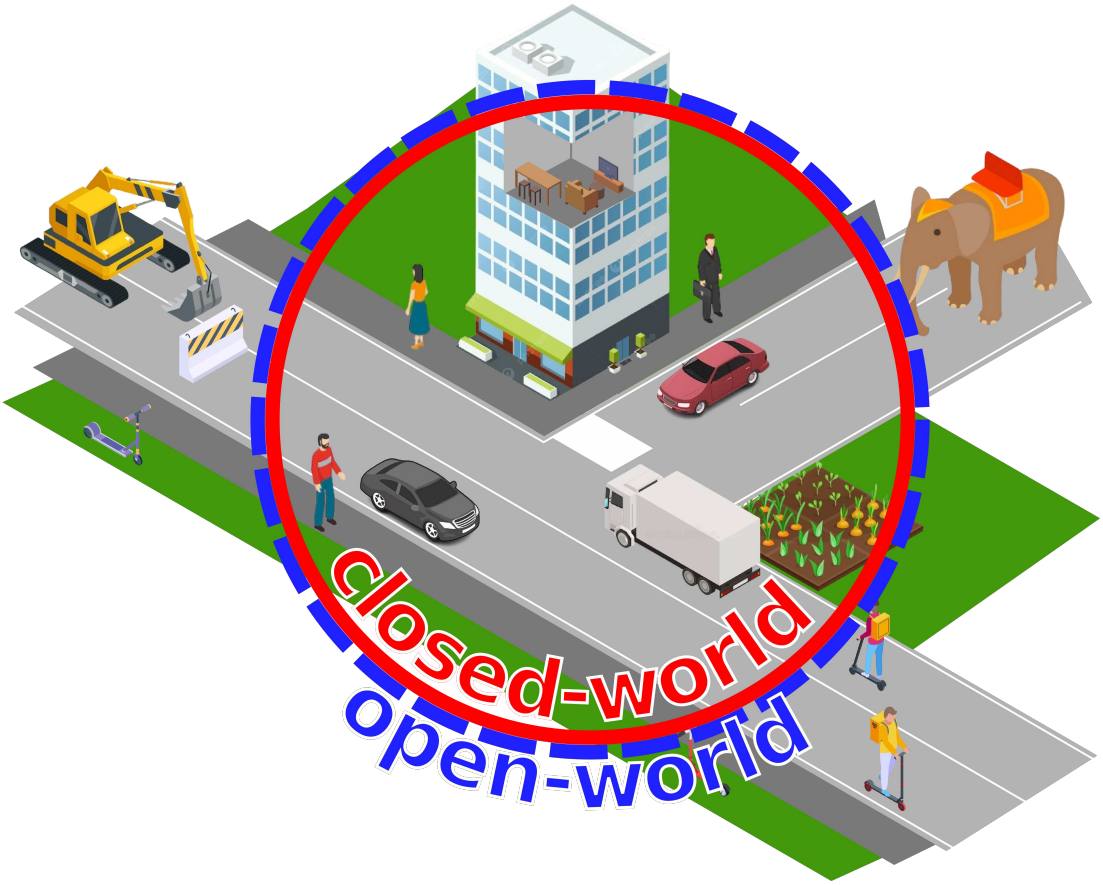


Figure 1.1: Illustration of the concepts of closed world and open world. In the closed-world setting, perception systems are limited to a predefined set of categories, typically those well represented in standard datasets. These include common traffic participants in autonomous driving, such as cars, trucks, and pedestrians, but also extend to other domains like indoor environments or agriculture. Beyond this "closed-world circle" lie objects that are absent from conventional training data: some may be new to society (e.g., electric scooters), others rare or domain-specific (e.g., excavators), or entirely out of distribution for the task at hand (e.g., an elephant on the road). Relaxing the closed-world assumption and embracing open-world scene understanding is essential for autonomous systems to operate safely and reliably in diverse real-world settings.

categories, as the network lacks the ability to recognize anything that does not appear in the training set. This property is commonly referred to as the *closed-world assumption* [162]. While effective within bounded domains, it fundamentally constrains the broader goal of holistic scene understanding, since achieving such a goal would require anticipating and labeling every possible category an autonomous agent might encounter to make it appear in the training dataset. Given the vast diversity of objects in the real world, such exhaustive annotation is not just prohibitively labor- and time-intensive, but practically infeasible.

In this thesis, we investigate panoptic segmentation from different perspec-

tives. The problem of panoptic segmentation is not coupled to any specific sensor input or domain of application, as robots can be equipped with a variety of sensors and be deployed in different settings, and they still rely on visual perception to accomplish their tasks. Thus, in this thesis, we target panoptic segmentation using a diverse set of sensors, and we apply it to different domains. Additionally, some of our algorithms push the boundaries of standard panoptic segmentation, while others aim to relax the aforementioned closed-world assumption to move towards an *open-world* scene interpretation. For this reason, we divide our thesis into two parts. Part I focuses on the standard formulation of panoptic segmentation under the closed-world assumption, which we will call *closed-world panoptic segmentation* (or simply panoptic segmentation) in the following. In Part II, we instead relax the closed-world assumption and explore *open-world segmentation*. This is crucial for deploying autonomous systems in the wild, as they, like us humans, will eventually encounter objects or situations they have never seen before and need to be able to operate safely and robustly, without causing harm to the surroundings. A schematic illustration of the concepts of closed world and open world is shown in Figure 1.1.

1.1 Main Contributions

The main contribution of this thesis is a set of novel approaches and datasets for panoptic segmentation, in both, closed-world and open-world settings.

We first introduce closed-world panoptic segmentation in Chapter 2 and discuss formal definitions and standard ways to tackle all closed-world segmentation tasks, building up from semantic and instance segmentation to panoptic segmentation. Those techniques are the foundation of our approaches, and a basic understanding of them is key to understanding our contributions.

In Chapter 3 we present an approach for RGB-D panoptic segmentation. Our goal is to investigate how to leverage different inputs, such as RGB images and depth maps, to improve segmentation performance. First, we introduce a new feature fusion module to merge RGB and depth cues in the convolutional neural network. We show how depth features are extremely helpful in enriching the information coming from the RGB. Second, we show that our approach is able to deal with missing input modalities, meaning that our neural network can segment also on RGB-only, depth-only, and full RGB-D input altogether, without the need to retrain. This capability is especially useful when a system is equipped with several sensors, such as RGB-D and RGB cameras, as it allows us to use a single model rather than having multiple sensor-specific ones. For this, we conduct experiments on two publicly available datasets for indoor navigation, where RGB-D data are widely used.

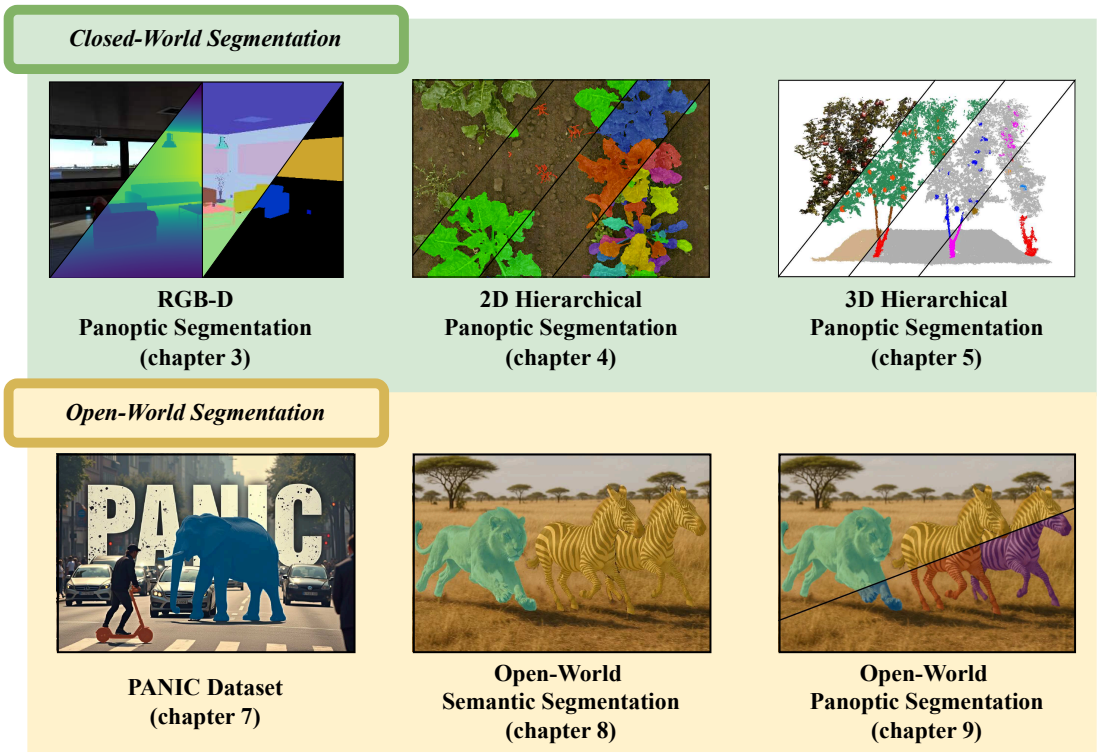


Figure 1.2: Overview of the thesis and all contributions it covers. In the top row, we show our three contributions to closed-world segmentation, namely an approach for RGB-D panoptic segmentation, an approach for 2D hierarchical panoptic segmentation, and one for 3D hierarchical panoptic segmentation. In the bottom row, we show our three contributions to open-world segmentation, namely a dataset for all open-world segmentation tasks, an approach for open-world semantic segmentation, and finally an approach for open-world panoptic segmentation.

The second contribution of this thesis, which we present in Chapter 4, is on 2D hierarchical panoptic segmentation. In several applications, it is important to go beyond the standard concept of semantic classes and object instances, and segment also more fine-grained components. Hierarchical panoptic segmentation tackles this problem, considering the case in which each object instance can be further segmented into individual, disjoint parts. This "nested" representation is what we aim to learn and exploit with the convolutional neural network we propose. For this task, we focus on the image domain in the agricultural context, where the semantic categories are typically soil, crop, and weed. Here, we aim to segment each individual crop and, for each one of them, each individual leaf. The intuition behind hierarchical panoptic segmentation is that these tasks are deeply interconnected and depend, to some extent, on each other: a reliable semantic segmentation simplifies extremely the plant instance segmentation; consequently, a strong plant instance segmentation reduces the complexity of leaf instance segmentation, as portions of different crops are, beyond any doubt, separate leaves.

The third contribution of this thesis, presented in Chapter 5, is on 3D hierarchical panoptic segmentation. Here, we extend the concepts and techniques of 2D hierarchical panoptic segmentation to the 3D case, and developed an approach to target 3D hierarchical panoptic segmentation. Additionally, we also recorded, labeled, and released a dataset of 3D point clouds of an apple orchard. We recorded data with a variety of sensors, including a terrestrial laser scanner, RGB-D, and RGB cameras, from which we extracted the point cloud resulting from bundle adjustment. The hierarchy in this case is defined differently from the crops of the previous chapter, as now each tree of the orchard is defined as a trunk and all the apples belonging to it, and each of these tree instances is composed of N instances, namely one trunk and $N-1$ fruits.

In Part II, we move beyond the closed-world assumption and investigate open-world segmentation. We start by providing an introduction in Chapter 6, and discuss previous approaches in open-world segmentation and precisely define and outline all open-world segmentation tasks of interest.

The fourth contribution of this thesis is a dataset that serves as a test set for all defined open-world segmentation tasks in the autonomous driving domain. We present this in Chapter 7. We recorded, labeled, and released this dataset, together with publicly available competitions, for the sake of reproducibility and benchmarking of approaches. Our dataset provides more than fifty new semantic classes that do not appear in the Cityscapes dataset, which is the most common dataset for segmentation in the autonomous driving scenario. Despite its importance, open-world segmentation has always been relatively underexplored with respect to its closed-world counterpart, mainly because of the lack of public benchmarks to evaluate approaches on. We believe that proposing such a dataset, especially when tailored to the autonomous driving application that benefits from tremendous traction these days, can help future research advance in open-world segmentation.

The fifth contribution, discussed in Chapter 8, is a method for open-world semantic segmentation, i.e., the task of discovering novel semantic categories at test time that have never appeared at training time. We achieve this by constructing a class descriptor for each known class during training, and updating the database of descriptors every time a novel class appears during testing. This allows us to discover a virtually unlimited number of novel categories, while still segmenting everything that is known, i.e., appears at training time. This is our first contribution that aims to relax the closed-world assumption and move towards open-world scene interpretation. However, it does not segment object instances, but rather considers only semantic classes.

Finally, in Chapter 9, we present our sixth and last contribution: an approach for open-world panoptic segmentation. This task adds the instance information

to the previously-discussed open-world semantic segmentation, allowing us to discover both, novel semantic categories and novel objects. With this, we achieve what we discussed at the very beginning of this chapter, that is, an algorithm that allows us to interpret and understand everything in the environment, without being limited by a pre-fixed taxonomy of categories.

To summarize, we propose several novel methods and datasets for closed-world and open-world panoptic segmentation to improve the vision capabilities of robotic systems operating in real-world environments. We investigated panoptic segmentation on different sensor data and different application domains, without focusing on one single modality or scenario, but rather exploring when and where the standard formulation of panoptic segmentation could be enriched and "upgraded". In Figure 1.2, we present a schematic overview that includes all contributions of this thesis. This thesis contributes to robotics and computer vision, exploiting techniques from both fields, including modern machine learning and deep learning approaches to achieve top-level performance for visual perception.

1.2 Publications

Parts of this thesis have been published in the following peer-reviewed conference and journal articles:

- M. Sodano, F. Magistri, T. Guadagnino, J. Behley, and C. Stachniss. Robust Double-Encoder Network for RGB-D Panoptic Segmentation. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2023. DOI: 10.1109/ICRA48891.2023.10160315.

- G. Roggiolani*, M. Sodano*, T. Guadagnino, F. Magistri, J. Behley, and C. Stachniss. Hierarchical Approach for Joint Semantic, Plant Instance, and Leaf Instance Segmentation in the Agricultural Domain. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2023. DOI: 10.1109/ICRA48891.2023.10160918.

- M. Sodano, F. Magistri, E. Marks, F. Hosn, A. Zurbayev, R. Marcuzzi, M. Malladi, J. Behley, and C. Stachniss. 3D Hierarchical Panoptic Segmentation in Real Orchard Environments Across Different Sensors. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2025. DOI: 10.1109/IROS60139.2025.11246899.

- M. Sodano, F. Magistri, L. Nunes, J. Behley, and C. Stachniss. Open-World Semantic Segmentation Including Class Similarity. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024. DOI: 10.1109/CVPR52733.2024.00307.

- M. Sodano, F. Magistri, J. Behley, and C. Stachniss. Open-World Panoptic Segmentation. *arXiv preprint*, arXiv:2412.12740, 2024 (under review). DOI: 10.48550/arXiv.2412.12740.

where the * symbol indicates equal contribution as first author.

Additionally, I have been involved in the following publications during my doctorate together with other researchers, but those are not directly part of this thesis:

- T. Guadagnino, X. Chen, M. Sodano, J. Behley, G. Grisetti, and C. Stachniss. Fast Sparse LiDAR Odometry Using Self-Supervised Feature Selection on Intensity Images. *IEEE Robotics and Automation Letters (RA-L)*, 7(3):7597–7604, 2022. DOI: 10.1109/LRA.2022.3184454.

- E. Marks, M. Sodano, F. Magistri, L. Wiesmann, D. Desai, R. Marcuzzi, J. Behley, and C. Stachniss. High precision leaf instance segmentation for phenotyping in point clouds obtained under real field conditions. *IEEE Robotics and Automation Letters (RA-L)*, 8(8):4791–4798, 2023. DOI: 10.1109/LRA.2023.3288383.

- N. Zimmerman, M. Sodano, E. Marks, J. Behley, and C. Stachniss. Constructing Metric-Semantic Maps using Floor Plan Priors for Long-Term Indoor Localization. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2023. DOI: 10.1109/IROS55552.2023.10341595.

- F. Magistri, R. Marcuzzi, E. Marks, M. Sodano, J. Behley, and C. Stachniss. Efficient and Accurate Transformer-Based 3D Shape Completion and Reconstruction of Fruits for Agricultural Robots. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2024. DOI: 10.1109/ICRA 57147.2024.10611717.

- J. Weyler, F. Magistri, E. Marks, Y.L. Chong, M. Sodano, G. Roggiolani, N. Chebrolu, C. Stachniss, and J. Behley. PhenoBench: A Large Dataset and Benchmarks for Semantic Image Interpretation in the Agricultural Domain. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 46(12):9583–9594, 2024. DOI: 10.1109/TPAMI.2024.3419548.

- E. Marks, L. Nunes, F. Magistri, M. Sodano, R. Marcuzzi, L. Zimmermann, J. Behley, and C. Stachniss. Tree Skeletonization from 3D Point Clouds by Denoising Diffusion. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2025.

- L. Lobefaro, M. Sodano, D. Fusaro, F. Magistri, M. Malladi, T. Guadagnino, A. Pretto, and C. Stachniss. Spatio-Temporal Consistent Semantic Mapping for Robotics Fruit Growth Monitoring. *IEEE Robotics and Automation Letters (RA-L)*, 10(9):9470–9477, 2025. DOI: 10.1109/LRA.2025.3594985.

- G. Bardak, M. Sodano, and M. Scholz. Integration of HD Maps and Point Clouds: An Efficient 3D Reconstruction Framework for Autonomous Driving Applications. *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 48:49–56, 2025. DOI: 10.5194/isprs-archives-XLVIII-4-W13-2025-49-2025.

- E. Marks, J. Bömer, F. Magistri, M. Sodano, A. Mahlein, J. Behley, and C. Stachniss. Autonomously estimating leaf parameters of individual plants in real agricultural environments. *Computers and Electronics in Agriculture*, 2025. (under review).

- L. Lobefaro, M. Sodano, L. Nunes, M. Malladi, L. Wiesmann, C. Stachniss. Annotation-Free Spatio-Temporal Consistent Instance Segmentation for Agricultural Robotics. *IEEE Robotics and Automation Letters (RA-L)*, 2026. (under review).

1.3 Open-Source Contribution

With the idea of facilitating reproducible and unbiased evaluation of new research ideas, in addition to the previously mentioned journal and conference publications, we made several open-source contributions including datasets and source code of our implementations:

- The code of our approach for RGB-D panoptic segmentation is available online at `https://github.com/PRBonn/PS-res-excite`;

- The code of our approach for hierarchical panoptic segmentation is available online at `https://github.com/PRBonn/HAPT`;

- The code of our approach for 3D hierarchical panoptic segmentation is available online at `https://github.com/PRBonn/hapt3D`;

- Our dataset for 3D hierarchical panoptic segmentation is available online at `https://www.ipb.uni-bonn.de/data/hops/`;

- The code of our approach for open-world semantic segmentation is available online at `https://github.com/PRBonn/ContMAV`;

- Our dataset for open-world segmentation tasks is available online at `https://www.ipb.uni-bonn.de/data/panic`.

Finally, I have contributed to several other open-source projects lead by colleagues highlighting the collaborative environment at the Photogrammetry and Robotics Lab of the University of Bonn.

Part I

Closed-World Panoptic Segmentation

Chapter 2

Introduction to Closed-World Segmentation

Perception is one of the fundamental building blocks for robot autonomy. Any system that is expected to operate without human supervision must be capable of perceiving, interpreting, and reasoning about its surroundings. For autonomous systems, perception is the gateway to safe and effective interaction with the world.

Within the field of robot perception, scene understanding has emerged as a central capability. The ability to analyze complex environments and extract meaningful information from raw sensor data enables autonomous agents to move beyond mere sensing and toward intelligent action. Scene understanding addresses two fundamental questions: *what* is present in the scene, and *where* it is located. Both of these questions must be answered reliably to achieve the holistic understanding necessary for any downstream task.

Panoptic segmentation has been proposed as a unified task to address these requirements [88]. It extends the capabilities of classical segmentation tasks by simultaneously providing information about the semantic category of each region in the scene and the individuality of the objects in it. For example, in an urban scenario, the model must not only recognize that certain pixels correspond to cars, but also separate each car into its own instance, while at the same time segmenting regions such as road, sidewalk, or sky. This unified view is critical because it allows understanding the difference between drivable and non-drivable regions, as well as distinguishing individual obstacles and traffic participants.

As mentioned in Chapter 1, the standard formulation of panoptic segmentation assumes a *closed world*, in which the set of categories to be recognized is fixed and fully specified during training. This closed-world assumption simplifies the task by providing a clear taxonomy of classes, allowing models to consistently parse scenes according to predefined categories. The closed-world

Figure 2.1: Overview of the contributions to closed-world panoptic segmentation, namely an approach for RGB-D panoptic segmentation, an approach for 2D hierarchical panoptic segmentation, and one for 3D hierarchical panoptic segmentation.

formulation, while restrictive, has provided a stable ground for developing and comparing methods, and continues to be a key reference point in the study of scene understanding.

In summary, closed-world panoptic segmentation provides a powerful and structured way to endow autonomous systems with holistic scene understanding. By combining semantic and instance-level information, it addresses the dual need of recognizing what is present in the environment and identifying each individual object. This capability forms the basis for many downstream tasks, making panoptic segmentation an essential component of modern perception pipelines. In this part of the thesis, we present a number of contributions to the state of the art of closed-world panoptic segmentation. We show them in Figure 2.1, which reports only the closed-world part of Figure 1.2.

2.1 Problem Definitions

To understand what panoptic segmentation entails and how it is formally defined, it is useful to introduce the problem incrementally. Panoptic segmentation combines the complementary goals of *semantic segmentation* and *instance segmentation* into a single, coherent formulation. Together, these tasks allow us to move beyond coarse classification toward a structured, fine-grained, and interpretable understanding of scenes.

In the following, we will primarily refer to RGB images and their pixels for clarity of exposition. However, the same concepts extend naturally to any type of sensory input. For example, in the case of LiDAR scans, points in the point cloud take the role of pixels, and the task of segmentation assigns semantic and instance labels to each of them.

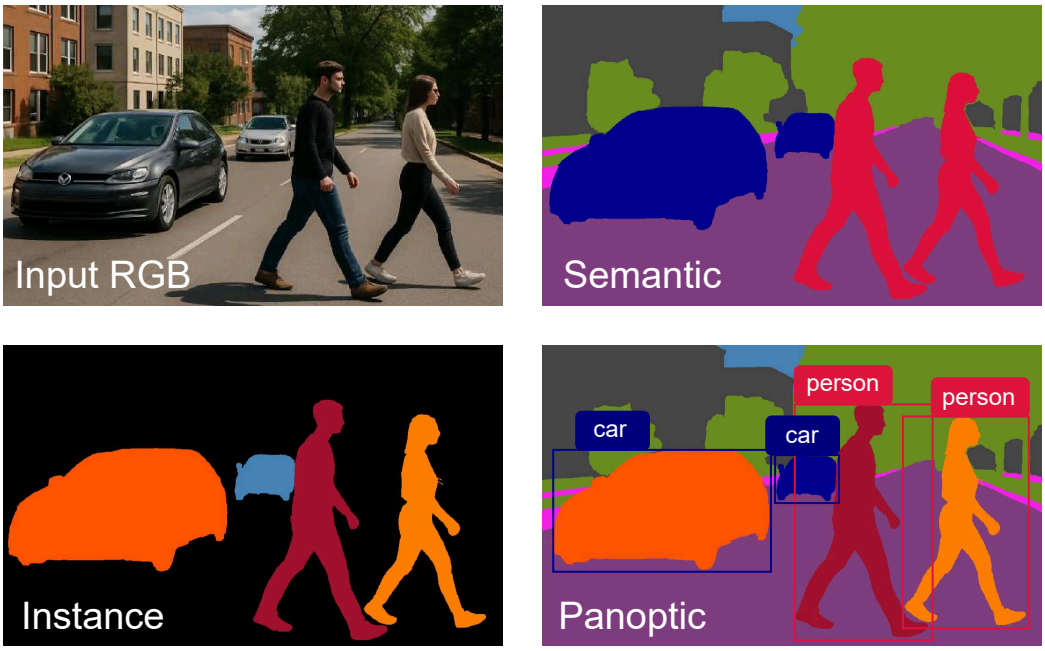


Figure 2.2: A visual breakdown of the three closed-world segmentation tasks. Given an input RGB image, semantic segmentation segments the scene into categories, like car (blue), person (red), building (gray), etc. Instance segmentation distinguishes among the different objects in the scene. Panoptic segmentation carries both, category and object information. RGB image is generated with GPT-4V [147].

2.1.1 Semantic Segmentation

Semantic segmentation addresses the task of scene categorization at the pixel level. The goal is to assign a semantic class, or category label, to every pixel in the image. For instance, pixels corresponding to road surfaces are assigned the label "road", while pixels corresponding to vehicles are assigned the label "car", "bus", etc. Importantly, semantic segmentation does not distinguish between different objects of the same class. Two different cars will both receive the label "car" and will be indistinguishable from each other in the segmentation mask.

Semantic segmentation is therefore a dense prediction task: every pixel in the input image is assigned a category. Conceptually, this extends the classical image-level classification problem down to the finest possible spatial granularity, turning each pixel into a classification target.

In the rest of this thesis, we interchangeably refer to the categories of the semantic segmentation task as classes, semantic classes, and semantic categories.

2.1.2 Instance Segmentation

Instance segmentation, in contrast, focuses on object individuation. The aim is to detect and delineate every distinct object in the scene, assigning a unique

identifier, also called instance ID, to the pixels belonging to each one. Pixels that belong to the same object instance receive the same ID, while pixels belonging to different objects are assigned different IDs.

Unlike semantic segmentation, instance segmentation is often class-agnostic. That is, it does not necessarily care about which category an object belongs to, but rather about separating objects as distinct entities. Moreover, it is not a dense task: certain areas of the image, such as sky or road surfaces, may not be assigned any instance ID, either because they do not correspond to discrete objects or because distinguishing instances there is not meaningful.

2.1.3 Panoptic Segmentation

Semantic and instance segmentation provide complementary views of the scene. Semantic segmentation captures *what* is present, while instance segmentation captures *how many* objects exist and *where* they are located, providing spatial separation between instances. Panoptic segmentation unifies them, assigning every pixel in the image a semantic class label and, when applicable, an instance ID. The resulting representation, often referred to as a panoptic ID, encapsulates both the semantic and instance information in a single, coherent output.

A key concept in panoptic segmentation is the distinction between *stuff* and *things*, first formalized by Adelson [2]. Stuff refers to amorphous, uncountable regions that are better described by texture or material than by object boundaries, like road, grass, or sky. Things, instead, are countable objects with discrete boundaries and clear individuality, such as cars, bicycles, or pedestrians. Semantic segmentation is typically used for stuff, while instance segmentation is applied to things. Panoptic segmentation reconciles the two perspectives, aiming to assign meaningful labels to both stuff regions and individual thing instances.

An intuitive illustration of the task breakdown into semantic, instance, and panoptic segmentation is shown in Figure 2.2. Together, these tasks represent a continuum of scene understanding, culminating in panoptic segmentation as the most holistic and complete formulation.

2.2 General Architectures

Among the many approaches developed to tackle this problem, convolutional neural networks (CNNs) have proven especially effective. Since the fundamentals of CNNs, their building blocks, and training procedures are already extensively covered in the literature, we do not review them here, but instead refer the reader to the textbook by Prince [156] for a comprehensive introduction.

2.2.1 Semantic Segmentation

Semantic segmentation is typically addressed by means of a CNN with an encoder-decoder architecture. The encoder progressively downsamples the input image while increasing the dimensionality of its feature maps, extracting high-level semantic representations at multiple scales. The decoder upsamples the feature map produced by the encoder back to the original resolution, producing an output with a feature dimension K, where K corresponds to the number of semantic categories used to train the network. Encoder and decoder are generally connected by means of skip connections. In their classical formulation introduced by Ronneberger *et al.* [168], they directly skip from early stages of the encoder to late stages of the decoder, and provide a direct gradient flow from the decoder to the encoder in the backpropagation, while also preserving low-level spatial information usually lost during downsampling in the forward pass.

A CNN for semantic segmentation is characterized by the last layer of its decoder, which has a number of neurons equal to K. The final output is a vector of scores for each pixel, where each score indicates, after normalization, the probability for that pixel to belong to a class k, $k \in \{1, ..., K\}$. It is important to highlight that the K semantic classes are fixed during training, and the network cannot assign labels outside of this predefined set. This restriction is known as *closed-world assumption*. Networks that operate under this assumption are usually referred to as *closed-world models*, and the pre-defined categories are called *known classes*. Consequently, any object not belonging to one of these known classes will be erroneously classified as one of them, limiting the network's ability to handle novel or unforeseen categories.

2.2.2 Instance Segmentation

Two major instance segmentation paradigms have been introduced in the literature: *top-down* methods and *bottom-up* methods. Top-down approaches predict instance proposals as object proxies, which are then filtered via non-maximum suppression and eventually refined [71, 90, 107, 190]. This process is likely to generate overlapping instance masks, and to resolve this ambiguity, several heuristics have been designed, based on the masks' predicted confidence scores [88], or pairwise relationship between categories [101] (e.g., ties should always be in front of person). Mask R-CNN [71] is the most widely used top-down framework, building on prior object detection networks [59, 163].

In contrast, bottom-up methods learn per-point embeddings and use them to cluster points so that they form a set of non-overlapping proposals [30, 53, 219]. These methods do not rely on heuristics to resolve conflicts coming from overlapping proposals, but instead on a post-processing module that clusters the

embeddings predicted at each pixel [5, 50, 131]. The concept of the embeddings is that of a descriptor: pixels belonging to the same instance must have the same descriptor, and pixels belonging to different instances must have different ones. The embedding can be a high-dimensional descriptor [135], as well as a 2D offset vector indicating a displacement for the pixel of interest, having all pixels of an instance spatially displaced to the same location in space [30, 205].

2.2.3 Panoptic Segmentation

Panoptic segmentation unifies semantic and instance segmentation, assigning both a semantic label and an instance identity to every pixel. CNN-based approaches for panoptic segmentation can also be divided into top-down [102, 214] and bottom-up [30,201] methods. These methods typically have a shared encoder, which feeds multiple decoders specialized for semantic and instance predictions. In recent years, a new wave of approaches based on vision transformers [32, 47, 100,225] arose, introducing the concept of object queries [24], enabling end-to-end mask prediction without separate post-processing steps [31, 200]. While conceptually elegant, these approaches often demand longer training times and require specifying a maximum number of object masks a priori.

In this thesis, we focus on CNN-based panoptic segmentation approaches. For the instance segmentation component, we employ bottom-up strategies across all tasks in both 2D and 3D, exploring the use of high-dimensional embeddings as well as 2D offset displacements. This choice allows us to maintain flexibility in handling multiple instances while leveraging robust feature representations learned by the shared encoder.

2.3 Metrics

The seminal paper introducing the task of panoptic segmentation [88] not only formalized the task itself, but also introduced a metric to properly evaluate it, called panoptic quality. The definition of panoptic quality relies on the intersection-over-union, the most common metric to evaluate semantic segmentation.

We evaluated all approaches present in this thesis using both, intersection over union and panoptic quality, to have a complete performance assessment in terms of semantic and panoptic quality. In the following, we will define and describe both metrics.

2.3.1 Mean Intersection Over Union

Intersection over union (IoU) [51] is a fundamental metric in computer vision to evaluate semantic segmentation. For each evaluated category, the IoU evaluates

(a) Ground truth (b) Prediction (c) IoU components

Figure 2.3: Components of the IoU computation reported in Equation (2.1) Given a ground truth mask (a) and a predicted mask (b), we show in (c) the true positives (TPs) as the predicted pixels that are also part of the ground truth, the false positives (FPs) as the predicted pixels that are not part of the ground truth, and the false negatives (FNs) as the ground truth pixels that are not predicted.

how accurately the predicted segment matches the ground truth segment. A visual example is shown in Figure 2.3. Given a ground truth label Y and a predicted label $\hat{Y}$, the IoU is defined as:

$$\text{IoU} = \frac{\hat{Y} \cap Y}{\hat{Y} \cup Y} = \frac{\text{FPs}}{\text{FPs} + \text{FPs} + \text{FNs}}. \tag{2.1}$$

The numerator considers the true positives (TPs), i.e., those pixels for which the prediction is the same as the ground truth. In other words, it indicates the shared area between prediction and ground truth. The denominator, in contrast, considers true positives, false positives (FPs), and false negatives (FNs), including also pixels that have been predicted as belonging to the evaluated category even if their ground truth is different, and pixels that have been predicted as belonging to another category even though their ground truth corresponds to the evaluated category. This ratio yields a score between 0 and 1, where 0 means there is no overlap between prediction and ground truth, while 1 means there is a perfect overlap with no false positives and false negatives.

IoU is a category-specific metric. Usually, the mean IoU (mIoU) is reported: the IoUs of all individual categories are averaged together in order to yield a single score that evaluates overall semantic segmentation performance.

2.3.2 Panoptic Quality

Panoptic quality (PQ) [88] is a recently introduced metric designed specifically for panoptic segmentation. The main motivation behind the introduction of PQ is that existing metrics focused either on things or stuff, but never on both. An exemplary image describing the components involved in the computation of the panoptic quality is shown in Figure 2.4. The computation of PQ happens in two stages: segment matching and the actual PQ computation.

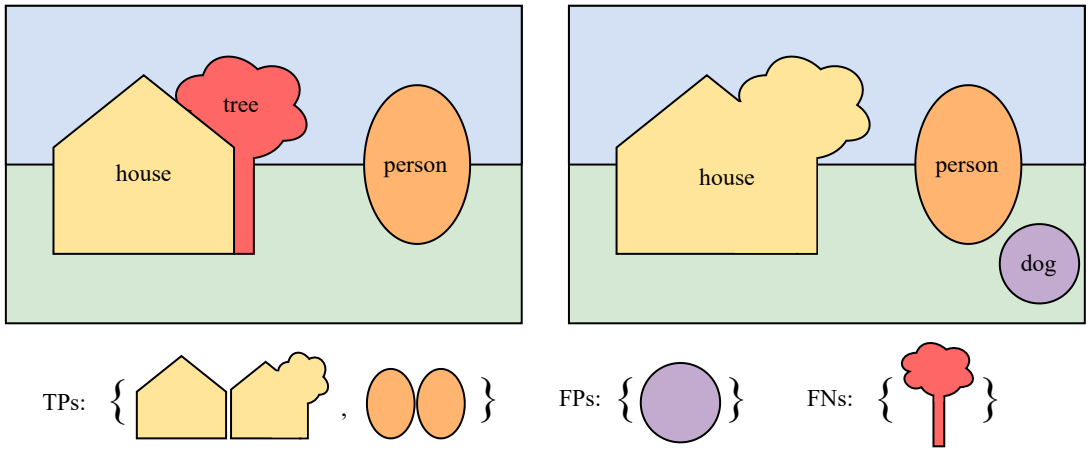


Figure 2.4: Components of the PQ computation reported in Equation (2.2) Given a ground truth mask (left) and a predicted mask (right), we show at the bottom the true positives (TPs) as the predicted segments with an overlap with the ground truth bigger than 50%, the false positives (FPs) as the predicted segments with no match in the ground truth, and the false negatives (FNs) as the ground truth segments with no predicted segment associated.

Given a predicted and a ground truth panoptic segmentation of an image, a predicted segment is matched to a ground truth segment if and only if they have the same semantic class and the IoU between them is strictly greater than 0.5. This ensures a unique matching: each ground truth segment is matched with at most one predicted segment. The resulting matches represent the true positives (TPs). After the matching phase, for each semantic class, there will be predicted segments with no associated ground truth, i.e., the false positives, and ground truth segments with no associated prediction, i.e., the false negatives. Notice that all this is completely independent of the concept of things or stuff: for things classes, there are several segments to match (i.e., all objects belonging to that category), while for stuff classes, there is only one. Finally, for each semantic category, PQ is computed as:

$$\mathrm{PQ} = \frac{\sum_{(\hat{s},\, s)\in \mathrm{TPs}} \mathrm{IoU}(\hat{s},\, s)}{|\mathrm{TPs}| + 0.5|\mathrm{FPs}| + 0.5|\mathrm{FNs}|}, \tag{2.2}$$

where $\hat{s}$ indicates a predicted segment and s a ground truth segment. Furthermore, by multiplying and dividing the PQ by the size of the true positive set, we obtain

$$\mathrm{PQ} = \underbrace{\frac{\sum_{(\hat{s},\, s)\in \mathrm{TPs}} \mathrm{IoU}(\hat{s},\, s)}{|\mathrm{TPs}|}}_{\text{segmentation quality (SQ)}} \times \underbrace{\frac{|\mathrm{TPs}|}{|\mathrm{TPs}| + 0.5|\mathrm{FPs}| + 0.5|\mathrm{FNs}|}}_{\text{recognition quality (RQ)}}. \tag{2.3}$$

Written in this way, the recognition quality is the commonly-used F1-score, while the segmentation quality yields the average mIoU of matched segments.

Similarly to IoU, the values for PQ range from 0, indicating no true positive matches, to 1, indicating all perfect matches and no false positives nor false negatives. For assessing general panoptic segmentation performance, panoptic quality is also usually averaged, and a cumulative score encapsulating the performance on all categories is reported.

Chapter 3

RGB-D Panoptic Segmentation

In Chapter 2, we described the task of panoptic segmentation, and the standard ways to address it. We also described its importance, as the ability to recognize objects and obtain a semantic interpretation of the environment is one of the key capabilities of truly autonomous systems.

In this chapter, we target panoptic segmentation using RGB-D sensors. In recent years, RGB-D imagery has proven relevant and efficient across a wide range of applications. These include autonomous navigation [14, 85, 140], augmented reality [33, 151], and indoor scene reconstruction [22, 236]. Indoor environments present numerous challenges for this task. Occlusions, complex and wide-ranging contextual relations among objects, illumination changes, as well as variations in appearance, scale, pose, and viewpoint, make indoor perception extremely hard. For instance, occlusions can lead to incomplete object boundaries, making it difficult for segmentation models to distinguish between closely packed or overlapping objects, such as chairs arranged around a table. Likewise, scale variation, common in indoor settings due to proximity differences, can impact the accuracy of semantic labeling, especially for small or distant objects. Depth provides geometric information that can help alleviate some of these challenges, providing, for example, illumination-independent features that can be vital for indoor scene understanding. Depth cues can enhance geometric reasoning by making it easier to distinguish between foreground and background instances, resolve boundary ambiguities, and improve segmentation of textureless or visually similar objects. For panoptic segmentation, where both semantic categories and instance identities must be predicted, incorporating depth allows for more robust spatial separation and improved structural understanding.

A central challenge in developing an algorithm for RGB-D panoptic segmentation is that it needs to be able to process both, RGB and depth, at the same time. Differently from the standard formulation of panoptic segmentation we discussed in Chapter 2, where a convolutional neural network takes as input a single sensor

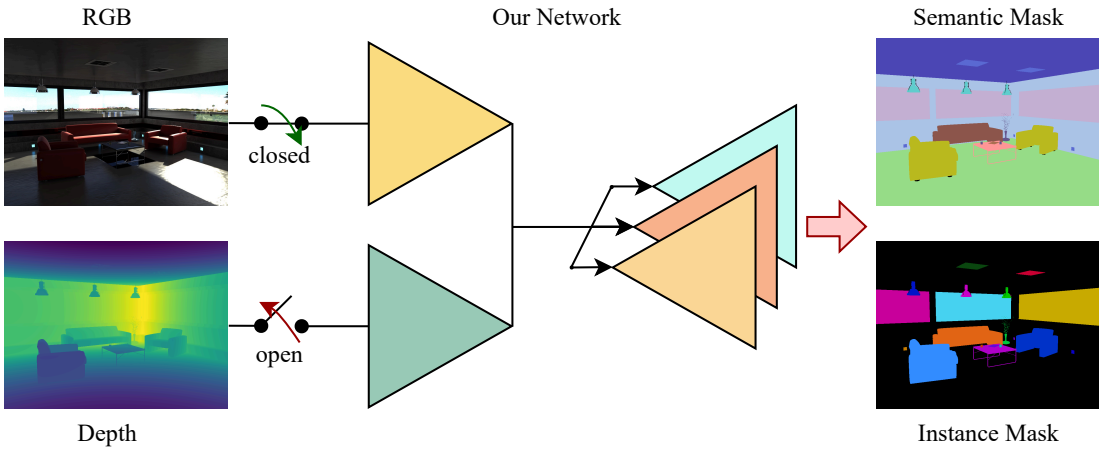


Figure 3.1: Our network is designed to take both an RGB image and a depth image as input, but it operates even when one of the modalities is unavailable. As shown by the switch depicted after the input images, each input stream may either reach the network or be omitted entirely: in the example, the RGB input reaches the network as its switch is closed, while the depth input does not since its switch is open. When present, the RGB and depth cues are processed by separate encoders. The model outputs both, a semantic segmentation mask and an instance segmentation mask.

reading (e.g., an RGB image), in this case we need to simultaneously process both sensor information, and merge their resulting features in a meaningful way before we get to the decoder phase of the CNN.

Additionally, we address the problem of being robust to missing cues, i.e., when either the RGB or the depth image is missing. This is a practical issue, as in real-world robotic systems, sensors are often subject to environmental limitations or hardware failures. For instance, RGB cameras may fail to provide meaningful input under low-light or overexposed conditions, while depth sensors can suffer from reflection artifacts, interference in outdoor environments, and reduced accuracy or complete signal loss at long range. Thus, a single model able to handle various input modalities, whether they are RGB-only, depth-only, or both simultaneously, is helpful in practical applications. Furthermore, it also simplifies development, as it removes the need to train and maintain several models for each individual application, and deployment, as one single unified model is enough for operations, reducing the computational workload on the robot. To this end, we investigate how an encoder-decoder architecture with two encoders for the RGB and depth cues can provide compelling results in indoor scenes. Previous efforts showed how double-encoder architectures are effective in processing RGB-D data [152, 177], but they target only semantic segmentation.

The main contribution of this chapter is a novel approach for RGB-D panoptic segmentation based on a double-encoder architecture. We propose a novel

feature merging strategy, called ResidualExcite, for merging the features of the two encoders at different levels to enrich the feature extraction. In addition, our double-encoder structure allows training and inference with various input modalities, like RGB-D, RGB-only, and depth-only, without the need to re-train the model (see Figure 3.1). We show that (i) our fusion mechanism performs better with respect to other state-of-the-art fusion modules, and (ii) our architecture allows training and inference on RGB-D, RGB-only, and depth-only data without the need for a dedicated model for each modality. We report extensive experiments on two indoor RGB-D datasets: ScanNet [40] and HyperSim [164].

3.1 Related Work

Panoptic segmentation [88] is extremely common for RGB data, with both, top-down [71, 87] and bottom-up [30, 53, 165] approaches being developed in the last years. We focus on bottom-up approaches as they have the advantage of not predicting overlapping segments, since they operate directly at pixel level. Panoptic segmentation is also common for LiDAR data, both in the form of range images [132] and point clouds [54]. However, when considering RGB-D data, semantic segmentation [41, 158] and instance segmentation [49, 76] are common, while panoptic segmentation has received less attention so far [210]. The most common ways of elaborating RGB-D data rely on 3D representations via truncated signed distance functions [76] or voxel grids [67]. Few works go in the direction of using RGB-D images [37] directly. In our approach, we target panoptic segmentation directly on RGB-D frames, instead of relying on an intermediate representation.

Double-encoder architectures are the most successful way for processing 2D representations of RGB-D frames. They allow processing RGB and depth cues separately with individual encoders and rely on feature fusion for combining the outputs of the encoders [153, 177]. An alternative to the direct exploitation of RGB and depth, proposed by Gupta *et al.* [65], consists of a pre-processing of the depth to encode it with three channels for each pixel, describing horizontal disparity, height above ground, and angle between the pixel's surface normal and the gravity direction. The core idea of all these works, however, is that RGB and depth are processed separately, and fusion happens only at a later point in the network, after the encoding part. This process is called *late fusion*, and consists of merging the two separate feature representations right before the decoders. Hazirbas *et al.* [70], however, show that feature merging at different feature resolutions, called early-mid fusion, can enhance performance.

A parallel research direction explored the possibility of leveraging depth not simply as an additional input channel, but as a factor that actively shapes the operations of the network. Originally proposed by Wang *et al.* [202], the key

idea is to modify standard CNN operations by incorporating depth information directly into their computation. In practice, this has been applied to convolutional layers [212,213], where the receptive field or kernel weights are adapted according to depth values, and to pooling operations [211], where the aggregation of features is guided by depth similarity rather than purely spatial proximity. In this way, depth serves as a structural prior that influences how features are combined and propagated through the network.

We adopt the first approach and design a double-encoder architecture to process RGB-D frames. Instead of committing to purely early or late fusion, our network performs multi-resolution merging at every downsampling stage of the encoder, thereby strengthening the exchange of information between the RGB and depth branches.

Different merging strategies for features of data streams are available. Summation [70] and concatenation [113] are the earliest strategies, which have the limit of considering all features without weighing them according to their effective usefulness. Newest efforts go in the direction of Squeeze-and-Excitation modules [177] and gated fusion [215], which are two different channel-attention mechanisms that aim to increase the focus on features that are more relevant. Other works exploit correlations between modalities to recalibrate feature maps based on the most informative features [180, 193]. Recent research has further investigated cross-modal interactions and feature recalibration. For instance, the Multimodal Transfer Module [83] recalibrates individual modalities by first generating a joint representation from both streams and then applying gating signals to each modality separately, enabling mutual influence. Another example is the Attentional Feature Fusion [42], which employs multi-scale channel attention to fuse features from different layers and streams more effectively, especially when feature scales or semantics differ. In the RGB-D domain specifically, RFBNet [44] introduces residual fusion blocks that interconnect separate RGB and depth encoders through gated residual units. This structure extracts modality-specific features and jointly-learned complementary ones, leading to state-of-the-art performance in segmentation tasks. Feature merging is also common for LiDAR, where it is relevant to merge information from point clouds, intensity images, and range images [180]. In our work, we build on top of channel-attention mechanisms. We propose a new merging mechanism called ResidualExcite, inspired by Squeeze-and-Excitation and residual networks [72], that aims to measure the importance of features at a more fine-grained scale.

Additionally, we leverage the double-encoder structure to have a single model capable of training and inferring on different modalities (RGB-D, RGB-only, depth-only). Multi-modal models have been investigated in the past, but mostly exploiting multiple "expert models" whose outputs are fused in a single predic-

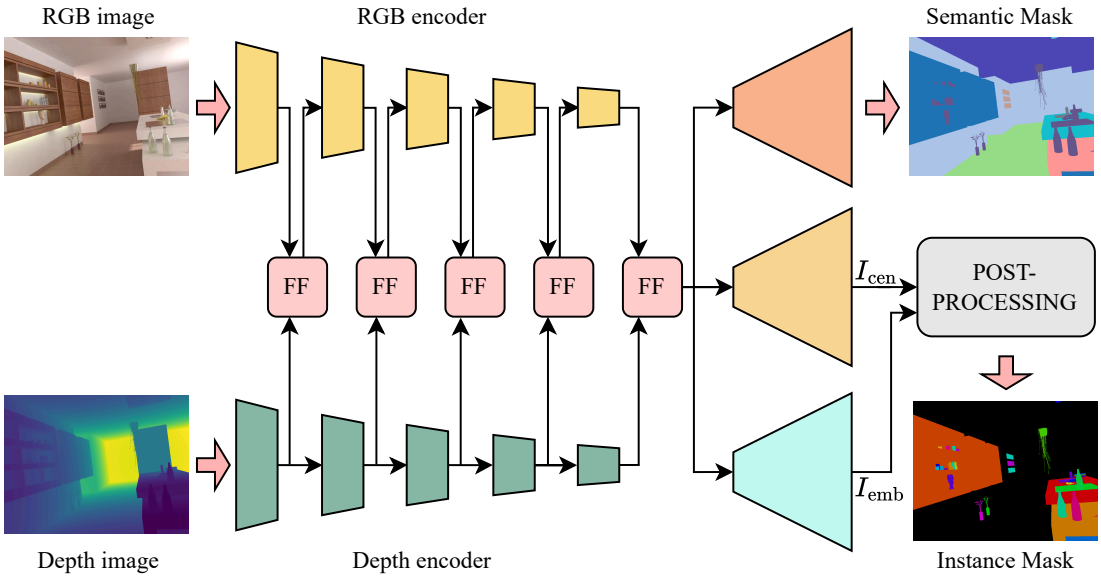


Figure 3.2: Our double-encoder network for RGB-D panoptic segmentation. RGB and depth images are separately processed, and their features are merged at different output strides by the feature fusion modules (FF).

tion, as in the work by Blum *et al.* [16]. After the publication of the article on which this chapter is based [187], the development of architectures robust to missing input modalities gained traction and showed promising results [231, 232], even showing how multimodal learning has the potential to surpass unimodal performance [228, 233].

3.2 Our Approach to RGB-D Panoptic Segmentation

Our panoptic segmentation network is an encoder-decoder architecture that operates on RGB-D images and processes RGB and depth by means of two different encoders. We merge encoder features at different resolutions, and send them to three decoders that restore the backbone features to the original image size. The first decoder targets semantic segmentation. The second decoder predicts the location of object centers in the form of a probability heatmap. The third decoder predicts an embedding vector for each pixel of the image. A post-processing module aggregates information coming from the last two decoders to obtain instance segmentation in a bottom-up fashion. Figure 3.2 shows our proposed network architecture. The next sections explain the individual parts of our method.

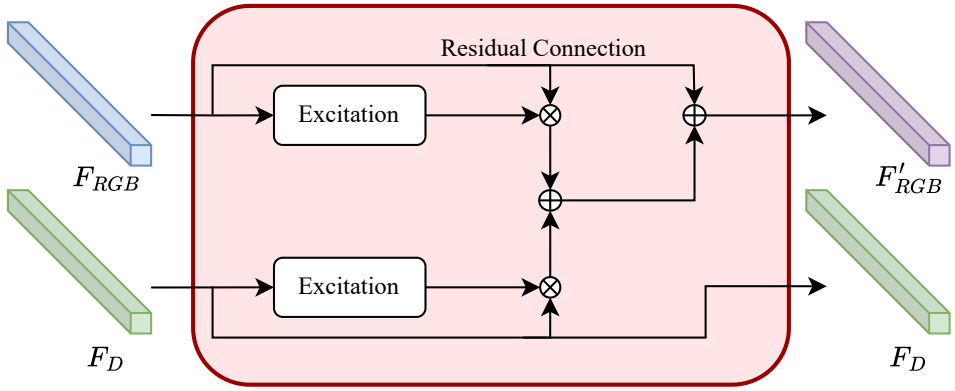


Figure 3.3: Detail of the ResidualExcite module. It elaborates the feature maps and produces a novel one that encodes information from both RGB and depth. Symbols $\bigoplus$ and $\bigotimes$ stand for elementwise addition and multiplication, respectively.

3.2.1 Encoders

Our panoptic segmentation network is based on two ResNet34 encoders [72], which are fed with the RGB image $I_{\text{rgb}} \in \mathbb{R}^{H \times W}$ and the depth image $I_{\text{depth}} \in \mathbb{R}^{H \times W}$, respectively. In both encoders, the basic ResNet block is replaced by the Non-Bottleneck-1D block [166], which allows a more lightweight architecture than the vanilla ResNet, since all 3×3 convolutions are replaced by a sequence of 3×1 and 1×3 convolutions with a ReLU in between, while increasing segmentation performance [177]. We merge features from the two encoders at different resolutions and project them into the RGB encoder. We provide more details about our merging strategy in Section 3.2.2. After the last merging, the resulting feature is processed by an adaptive pyramid pooling module [229], which has the role of increasing the receptive field of the network. From the RGB encoder, we extract features at different resolutions and use them in the decoders by means of skip connections [168], which have the role of transporting low-level spatial information coming from the encoder, usually lost during downsampling, to the later stages of the decoder, while also providing a direct gradient flow to the encoder.

3.2.2 Feature Fusion

We perform feature fusion in the encoders at different resolutions. We merge features from the two encoders at every downsampling step and use them in the RGB encoder only. The depth encoder processes depth features only to avoid processing the same features with both encoders.

We propose a novel way of merging features, inspired by the Squeeze-and-Excitation module [77]. This module produces a channel descriptor (squeezing operation) and assigns to each channel a modulation weight that is applied to the

feature map (excitation). Our goal is to obtain a global modulation weight rather than a channel-wise weight, as we believe that a more fine-grained reweighing of features is crucial for effective segmentation results. Thus, we remove the squeezing operation, and we add a residual connection for a better flow of information. This module, called ResidualExcite (see Figure 3.3), is given by

$$\boldsymbol{X}_{\text{rgb}} = \boldsymbol{X}_{\text{rgb}} + \lambda \left(E(\boldsymbol{X}_{\text{rgb}}) \odot \boldsymbol{X}_{\text{rgb}} + E(\boldsymbol{X}_{\text{depth}}) \odot \boldsymbol{X}_{\text{depth}} \right), \tag{3.1}$$

where $\boldsymbol{X}_i \in \mathbb{R}^{C_d \times H_d \times W_d}$, $i \in \{\text{rgb}, \text{depth}\}$ is the feature coming from the respective branch, $E(\boldsymbol{X}_i) \in \mathbb{R}^{C_d \times H_d \times W_d}$ is the excitation module, which is a sequence of 1×1 convolutions followed by a sigmoid activation function, λ is a non-trained parameter for weighing the excitation module over the residual connection, the subscript d refers to the dimension of the features at the specific output stride in which the merging happens, and $\odot$ indicates the Hadamard product (element-wise multiplication). The output of the excitation procedure $E(\boldsymbol{X}_i)$ is a tensor with the same spatial dimensions as the input tensor, and scores between 0 and 1. Basically, this mechanism generates an element-wise weight for all entries of the considered feature map $\boldsymbol{X}_i$. Then, $E(\boldsymbol{X}_i)$ reweights the feature map $\boldsymbol{X}_i$ by means of an element-wise multiplication. The RGB and the depth features are both individually excited and then summed, so that each of them can be used separately in case the other cue is missing. Finally, a residual connection adds $\boldsymbol{X}_{\text{rgb}}$ again.

3.2.3 Decoders

The decoders are composed of three SwiftNet-like modules [148], where we incorporate Non-Bottleneck-1D blocks, and we extend the feature channel to 512 in the first module, and then we reduce it as the resolution increases. Finally, two upsampling modules based on nearest-neighbor and depthwise convolutions, that are less computationally expensive than transposed convolutions [177], restore the original resolution. Our model is composed of three decoders for semantic segmentation, center prediction, and embedding prediction.

In the following, we call $\Omega = \{(1, 1), \ldots, (H, W)\}$ the set of pixels in the image, and p the individual pixel tuple, for which we drop the bold notation. We will adopt this notation throughout the thesis, unless explicitly specified.

Semantic Segmentation. The semantic segmentation decoder has an output feature dimension equal to the number of semantic classes K, followed by a softmax activation function. We call the output of this decoder $I_{\text{sem}} \in \mathbb{R}^{H \times W \times K}$. This decoder is trained with the standard cross-entropy loss $\mathcal{L}_{\text{sem}}$:

$$\mathcal{L}_{\text{sem}} = -\frac{1}{|\Omega|} \sum_{p \in \Omega} \omega_k \, \boldsymbol{t}_p^{\top} \, \log \left(\sigma(\boldsymbol{f}_p) \right), \tag{3.2}$$

where ω_k is a class-wise weight computed via the inverse frequency of each class in the dataset, $\boldsymbol{t}_p \in \mathbb{R}^K$ is the one-hot encoded ground truth annotation at pixel location p, $\sigma(\cdot)$ denotes the softmax operation, and $\boldsymbol{f}_p$ denotes the pre-softmax feature predicted at pixel p.

Center Prediction. The center prediction decoder has an output feature dimension of 1, followed by a sigmoid activation function to predict pixelwise probabilities of being a center. The output of this decoder is named $I_{\text{cen}} \in \mathbb{R}^{H \times W}$. We optimize it with a binary focal loss [135]:

$$\begin{aligned} \mathcal{L}_{\text{cen}} &= \frac{1}{|\Omega|} \sum_{p \in \Omega} \text{BFL}(\hat{c}_p,\, c_p) \\ \text{BFL}(\hat{c}_p,\, c_p) &= \begin{cases} -\alpha\,(1-\hat{c}_p)^{\tau}\,\log(\hat{c}_p) & \text{, if } c_p = 1 \\ -(1-\alpha)\,\hat{c}_p^{\,\tau}\,\log(1-\hat{c}_p) & \text{, if } c_p = 0 \end{cases} \end{aligned}, \tag{3.3}$$

where $c_p = \{0,\, 1\}$ is the binary ground truth variable indicating whether pixel p is a center or not, $\hat{c}_p$ is the center prediction at pixel p, α and τ are design parameters and are fixed in all experiments to 0.1 and 2, respectively.

Embedding Prediction. The third decoder of the network we propose predicts a D_{emb}-dimensional embedding vector for each pixel in the image. We name the output of this decoder $I_{\text{emb}} \in \mathbb{R}^{H \times W \times D_{\text{emb}}}$. We optimize this decoder with a composed hinge loss, which is composed of three separate terms. The first term $\mathcal{L}_{\text{att}}$ attracts embedding vectors of pixels belonging to the same instance, the second term $\mathcal{L}_{\text{rep}}$ repels embedding vectors of pixels belonging to different instances, and the third term $\mathcal{L}_{\text{reg}}$ is a regularization term that aims to keep the norm of the embeddings as small as possible, avoiding exploding entries:

$$\mathcal{L}_{\text{emb}} = \beta_1\,\mathcal{L}_{\text{att}} + \beta_2\,\mathcal{L}_{\text{rep}} + \beta_3\,\mathcal{L}_{\text{reg}}, \tag{3.4}$$

$$\mathcal{L}_{\text{att}} = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{P_n} \sum_{p=1}^{P_n} \left[\, \left\| \hat{\boldsymbol{e}}_n - \hat{\boldsymbol{e}}_p^{\,n} \right\| - \delta_{\text{a}} \right]^{+}, \tag{3.5}$$

$$\mathcal{L}_{\text{rep}} = \frac{1}{N(N-1)} \sum_{\substack{n_1=1 \\ n_1 \neq n_2}}^{N} \sum_{n_2=1}^{N-1} \left[\delta_{\text{r}} - \left\| \hat{\boldsymbol{e}}_{n_1} - \hat{\boldsymbol{e}}_{n_2} \right\| \right]^{+}, \tag{3.6}$$

$$\mathcal{L}_{\text{reg}} = \frac{1}{N} \sum_{n=1}^{N} \left\| \hat{\boldsymbol{e}}_n \right\|, \tag{3.7}$$

where N is the number of instances in the image, $\hat{\boldsymbol{e}}_n \in \mathbb{R}^{D_{\text{emb}}}$ is the unbounded logit predicted by the decoder at the center of instance n, $\hat{\boldsymbol{e}}_p^{\,n} \in \mathbb{R}^{D_{\text{emb}}}$ is the unbounded logit predicted by the decoder at any pixel p that belongs to instance n, P_n is the number of pixels of the k-th instance, $[\cdot]^+$ corresponds to $\max(0, \cdot)$, and δ_{a} and δ_{r} are thresholds for attracting and repelling the embeddings, respectively.

To speed up computations, we compute $\mathcal{L}_{\text{att}}$ only between pixels belonging to an instance and their corresponding center, and $\mathcal{L}_{\text{rep}}$ only among centers of different instances. Similarly, we regularize only the vectors of the centers.

We optimize the network with a panoptic loss that is a weighted sum of the previously-defined terms:

$$\mathcal{L}_{\text{pan}} = w_1 \, \mathcal{L}_{\text{sem}} + w_2 \, \mathcal{L}_{\text{cen}} + w_3 \, \mathcal{L}_{\text{emb}}. \tag{3.8}$$

3.2.4 Post-Processing

Our post-processing module computes the instance mask based on the output of the three decoders. Since the center prediction decoder usually outputs blobs around the desired center, we perform a non-maximum suppression operation in order to reduce each blob to a single pixel, filtered by the semantic prediction to ensure consistency.

In particular, centers are first filtered by the semantic prediction I_{sem} to avoid having centers belonging to stuff classes, which do not have any instances. Then, pixels that have a probability of being a center lower than a threshold δ_{cen} are discarded. Next, for each blob, we extract the pixel with the highest probability of being a center. A blob $\mathcal{B}$ is defined as the set of pixels belonging to the same semantic class and having a similar embedding vector. Referring to Ω_c as the set of pixels that are predicted as centers in I_{cen}, i.e., $\Omega_c = \{p \mid I_{\text{cen}}(p) \geq \delta_{\text{cen}}\}$, a blob is defined as

$$\mathcal{B} = \left\{p \in \Omega_c \mid I_{\text{sem}}(p) = k \, \wedge \, \forall p_i, \, p_j \in \mathcal{B}, \left\|\hat{\boldsymbol{e}}_{p_i} - \hat{\boldsymbol{e}}_{p_j}\right\| < \delta_{\text{emb}}\right\}, \tag{3.9}$$

where k is a specific semantic class, δ_{emb} is a threshold for aggregating embedding vectors, and p_i, p_j are generic pixels.

After the center extraction, we perform an agglomerative clustering operation to group pixels to centers according to the Euclidean distance in the embedding space and semantic class. For each center, we compute its distance in the embedding space from all pixels of the same semantic class. This operation is less computationally intensive than the similarity matrix between all pixels of the image, and motivates the use of object centers. Finally, we assign the pixel to a center if the distance among them in the embedding space is below a threshold θ. The use of the semantic segmentation prediction enforces consistency and avoids grouping pixels belonging to different semantic classes in the same object.

3.2.5 Dealing with Missing Inputs

Since we process RGB and depth with two separate encoders, we can feed the network with partial information, i.e., without either RGB or depth, and freeze

the part corresponding to the missing data. This can also be done at training time, with a switching mechanism that freezes gradients if no input is provided to one branch. In this way, the frozen encoder does not contribute to the forward and backward pass, and the network can train with complete RGB-D, RGB-only, or depth-only images. Furthermore, the network is able to infer on different data without the need for re-training. Additionally, the feature fusion module is explicitly designed in such a way as not to be affected by missing modalities.

We train the full model with a probability of dropping data (RGB or depth), equal to p_{drop}. This means that the network can train either with the full RGB-D data or not. If data is dropped, then no input is sent to the corresponding encoder, which we freeze. Additionally, we use an adaptive sampling mechanism to choose what needs to be dropped: in particular, if one cue has been dropped more times than the other, its probability of being dropped in the next iteration is reduced. This helps to have a more balanced dropping mechanism and alleviates the problem of always dropping the same modality.

3.3 Experimental Evaluation

We present our experiments to show the capabilities of our method and compare it with other fusion methods common in the literature. Furthermore, we show the performance of models trained with partial data.

3.3.1 Experimental Setup

Datasets and metrics. We test our method on the validation sets of two datasets: ScanNet [40] and HyperSim [164]. ScanNet is composed of 2.5M real-world images organized in 1,513 scenes. HyperSim is a photorealistic synthetic dataset of indoor scenes, and it is composed of 77.4K images organised in 461 scenes. HyperSim annotations follow the definition given by Kirillov *et al.* [87], for which stuff classes (wall, floor) do not have instances. On the contrary, ScanNet reports instance annotations also for walls and floors. We filter them out and do not consider these classes in the instance segmentation.

For the center prediction, we pre-process the instance masks of both datasets to extract a center ground truth that is inside the object mask. We consider this to be more effective than computing the center of the associated bounding box, which can fall outside the object mask and the segmentation mask, for example, in the case of an isolated concave object.

As discussed in Section 2.3, we evaluate our method by means of the panoptic quality (PQ) [87]. We also report the mean intersection over union (mIoU) [51] over all classes to evaluate the performance of the semantic segmentation head.

Table 3.1: Performance of the different panoptic segmentation methods on ScanNet and HyperSim. EsaNet-add and EsaNet-SE refer to the two variants of EsaNet for feature fusion, based on addition or Squeeze-and-Excitation. Best results in bold. All metrics are reported as percentages (%).

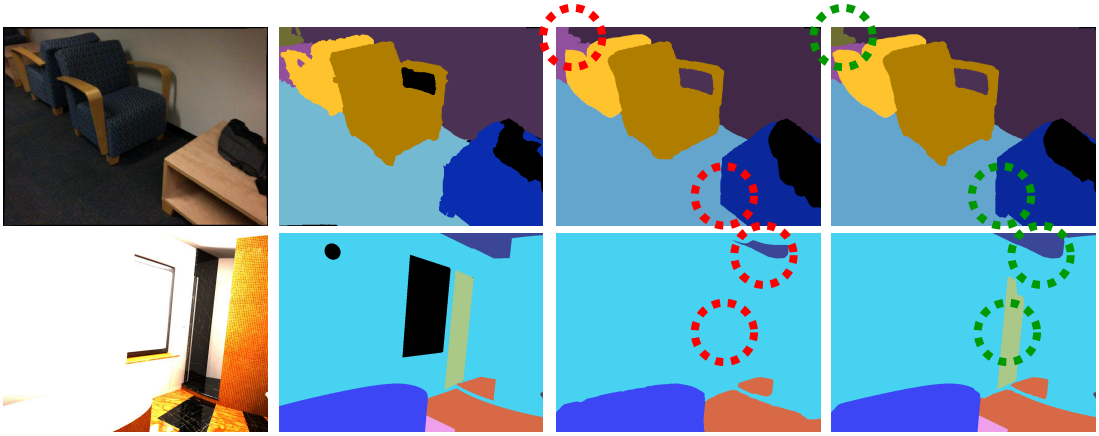
Approach	ScanNet		HyperSim	
	mIoU	PQ	mIoU	PQ
Panoptic DeepLab	43.1	30.1	40.5	26.1
4D Panoptic DeepLab	45.5	31.4	41.2	28.6
EsaNet-add [177]	51.8	35.7	50.7	32.2
EsaNet-SE [177]	54.0	37.1	54.1	35.9
CBAM [209]	58.1	39.1	54.2	37.0
ResidualExcite (ours)	**59.0**	**40.9**	**55.1**	**38.7**

Training details and parameters. In all experiments, except when explicitly specified, we use the one-cycle learning rate policy [181] with an initial learning rate of 0.004. We perform random scale, crop, and flip data augmentation, and optimize with AdamW [115], for 200 epochs. The batch size is set to 32. Additionally, we set $D_{\text{emb}} = 32$ as embedding dimension, $\delta_a = 0.1$, $\delta_r = 1$, $\delta_{\text{emb}} = 0.5$, $\delta_{\text{cen}} = 0.5$, $\theta = 0.5$, and $\lambda = 1.5$. Loss weights are set to $w_1 = 1$, $w_2 = 0.1$, $w_3 = 10$, $\beta_1 = 1$, $\beta_2 = 1$, $\beta_3 = 0.001$.

3.3.2 Panoptic Segmentation on RGB-D Images

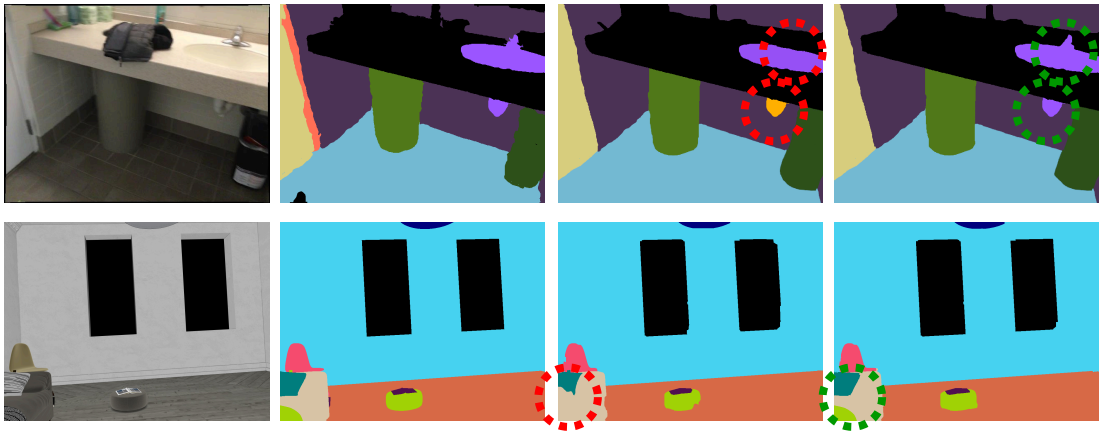
The first set of experiments evaluates the performance of our proposed method and offers comparisons to other architectures common in the literature. We base our work on ESANet [177], which is a double-encoder network for RGB-D semantic segmentation on images. To use it as a baseline for panoptic segmentation, we expand ESANet with the decoders for the center prediction and embedding prediction. Notice that ESANet leverages Squeeze-and-Excitation as a fusion strategy, but reports in the paper also fusion by addition, which simply sums up features coming from the two encoders and projects them into the RGB encoder. Here, we use both variants. Additionally, we use another fusion module as baseline, CBAM [209], which infers attention maps along two separate dimensions, channel, and spatial. Furthermore, we also compare against a single-encoder architecture that processes the RGB-D image as a four-channel input signal. For that, we adapted Panoptic DeepLab [30] to process images with four channels, and we fed the model with a 4D tensor that is the concatenation of the RGB and the depth images.

We compare our approach to such methods since we focus on image-like data,



(a) Input RGB (b) Ground truth (c) Panoptic Deeplab (d) Ours

Figure 3.4: Qualitative results of different panoptic segmentation approaches on ScanNet (first row) and HyperSim (second row).



(a) Input RGB (b) Ground truth (c) Sq. and Excitation (d) Ours

Figure 3.5: Qualitative results of different feature fusion modules on ScanNet (first row) and HyperSim (second row).

without relying on 3D representations such as truncated signed distance fields or point clouds. Results are reported in Table 3.1, qualitative results are shown in Figure 3.4 and Figure 3.5. Our reimplementation of Panoptic DeepLab shows inferior performance when compared to ESANet and our approach. We also report numbers from the vanilla implementation of Panoptic DeepLab that does not make use of the depth. Interestingly, the performance of the standard Panoptic DeepLab is close to the that of the RGB-D re-implementation, which simply processes an input with four channels rather than three. This suggests that processing depth as an additional input channel does not add much information, while a separate processing via a second encoder is more effective for such a task. Our ResidualExcite module helps segmentation performance and outperforms other attention-based merging strategies such as CBAM and Squeeze-and-Excitation.

Table 3.2: Performance of different semantic segmentation models on the ScanNet dataset. ResidualExcite-semantic corresponds to the variant of our approach that drops the two decoders for center and embedding prediction. Best result in bold. All metrics are reported as percentages (%).

Approach	mIoU
AdapNet++ [193]	54.6
FuseNet [70]	56.7
SSMA [193]	66.1
ResidualExcite (ours)	59.0
ResidualExcite-semantic (ours)	**69.8**

Fusion by addition shows inferior performance, which is an expected result as it processes all features without weighing them according to their effective usefulness. This experiment indicates that our more fine-grained weighing mechanism, which has an effect on each single entry of the encoder feature rather than each channel, enhances the performance on the downstream task.

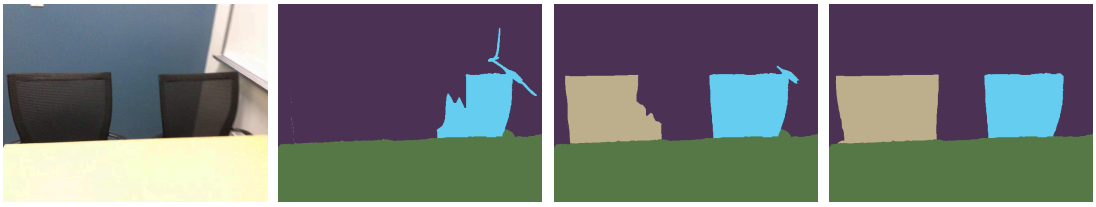
To empirically validate our architecture design, we compare it with some state-of-the-art models from the ScanNet benchmark for semantic segmentation [70, 193]. We use both our full model and its task-specific reduction, in which the decoders for center and embedding prediction are cut out to address semantic segmentation only. This is useful because the learning procedure is now fully focused on semantic segmentation only, and the encoder is not affected by the other decoders via backpropagation. Table 3.2 shows that even if our full model has weaker performance, our task-specific model outperforms the baselines. We use as baselines only methods that rely on a single frame, and exclude those that aggregate multiple views, as they cannot be directly compared to our approach.

3.3.3 Robustness to Missing Inputs

The second set of experiments shows that our approach can train and infer on partial data, such that the network learns to deal with missing RGB- or depth-frames. We test different values for p_{drop}: 0.25, 0.5, and 0.75. This means that the network will drop either the RGB frame or the depth frame according to the specified probability. If dropping happens, we choose which cue to drop according to the adaptive sampling mechanism mentioned in Section 3.2.5. This strategy gives a better performance than random sampling, which is therefore not reported here. Table 3.3 shows performance for inference on RGB-D, RGB-only, and depth-only data. All models produce inferior segmentation results to the model that does not drop any frame, i.e., our full model discussed above, when doing inference on full RGB-D frames. However, its performance drops

Table 3.3: Performance of the model when dropping either RGB or depth with different probability. Best results in bold. All metrics are reported as percentages (%).

p_{drop}	**RGB-D**		**RGB-only**		**Depth-only**	
	PQ	mIoU	PQ	mIoU	PQ	mIoU
0	**40.9**	**59.0**	9.1	21.1	12.5	24.6
0.25	31.1	44.6	20.1	34.8	21.2	35.1
0.5	30.7	42.9	25.6	39.2	26.7	38.9
0.75	26.8	40.1	**27.5**	**39.6**	**28.2**	**39.5**



(a) Input RGB (b) $p_{\text{drop}} = 0.25$ (c) $p_{\text{drop}} = 0.5$ (d) $p_{\text{drop}} = 0.75$

Figure 3.6: Results when doing inference on RGB-only after training with missing inputs. The bigger p_{drop}, the better the performance.

substantially when doing inference on partial data, as the network was never trained with missing cues. Additionally, we notice how dropping frames more often makes the model better for doing inference on partial data, since the network was trained more with missing cues. On the contrary, low values of p_{drop} bring poor performance when handling partial data, because the network was mainly trained with both, RGB and depth. The model trained with $p_{\text{drop}} = 0.5$ is the best compromise to achieve satisfactory results both on RGB-D, RGB-only, and depth-only, even without reaching the performance of the RGB-D model. Qualitative results are shown in Figure 3.6.

All experiments described in Section 3.3.3 are done with a batch size of 4 and an initial learning rate of 0.001. Due to the missing inputs, the training procedure is less stable and thus requires a smaller learning rate. We use ResidualExcite for merging; experiments are performed on ScanNet only.

3.3.4 Ablation Studies

In this last section, we provide ablations to show the improvements provided by the fusion strategy. We perform all ablations on the ScanNet dataset only.

First, we analyze the ResidualExcite and investigate the effect of the residual connection. Without it, the excitation module (ExciteOnly) still provides an entry-wise reweighing of the feature. Experiments show that this is already

Table 3.4: The first two lines refer to RGB- and depth-only. Then, we show double-encoder networks with different kinds of feature fusion. We use ✓ to indicate which branch processes fused features. Best results in bold. All metrics are reported as percentages (%).

RGB	**Depth**	**Fusion**	mIoU	PQ
✓		None	38.9	25.6
	✓	None	41.0	28.9
✓	✓	Addition	51.8	35.6
✓	✓	Squeeze-and-excitation	54.0	37.1
✓	✓	ExciteOnly	55.8	38.7
✓	✓	ResidualExcite	56.7	38.8
✓	✓	ResidualExcite	**59.0**	**40.9**

enough to ensure superior performance with respect to other baselines, but the residual connection gives further improvements, see Table 3.4. Additionally, in our case, fusing in the RGB encoder is more effective than fusing in the depth encoder.

In the same table, we compare the performance of the full model reductions in which a single encoder is used. We test panoptic segmentation on RGB-only and depth-only data. The results are clearly inferior to the double-encoder models. Interestingly, depth-only gives better results than RGB-only. This is probably due to the fact that some scenes have challenging lighting conditions, and some objects are hard to recognize in the RGB image. Such information is not lost in the depth image. Also, this suggests that geometric cues may be more relevant than color information when it comes to object recognition for segmentation.

3.4 Conclusion

In this chapter, we presented a novel approach for panoptic segmentation on RGB-D images of indoor environments. Our approach separately processes RGB and depth images, while merging their features at every downsampling stage of the encoders by means of a novel feature fusion mechanism. Our method enables training and inference when cues are missing using the same model and without the need for retraining. We implemented and evaluated our approach on different datasets and provided comparisons with other existing feature fusion modules. The experiments suggest that our more fine-grained reweighing of features is crucial for effective segmentation results. Furthermore, our multimodal model, trained with missing input cues, can infer on RGB-only, depth-only, and RGB-D data at test time.

This chapter showed how, by modifying the standard formulation of panoptic segmentation presented in Chapter 2, our approach is now able to deal with multiple sensor inputs at the same time. For this, we extended the standard panoptic segmentation architecture from the point of view of the input to accommodate multiple modalities. In the next chapter, we will investigate how to modify the panoptic segmentation task when multi-layered semantic or instance information is necessary. In particular, we will see how to integrate nested instances in the problem formulation. In this formulation, we aim to extend panoptic segmentation from the point of view of the output rather than the input, and thus we will go back to the original setting in which only one sensor input is available. Furthermore, we will move from the indoor domain to the agricultural setting, where the concept of nested instances is crucial for several tasks, from phenotyping to yield estimation.

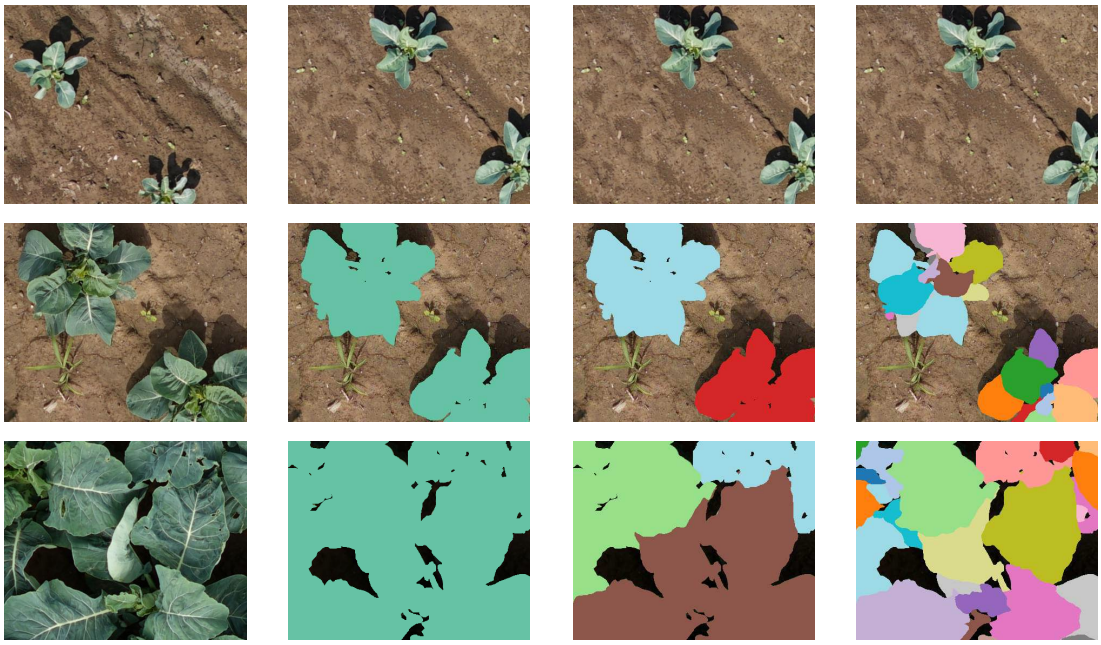
Chapter 4

Hierarchical Panoptic Segmentation

PANOPTIC segmentation, in its standard formulation presented in Chapter 2, aims to provide a category descriptor and an instance ID for each pixel in the image. However, there are some cases for which this information is not sufficient. Problems where nested object instances appear, such as human body part segmentation [105], cannot be easily solved with standard panoptic segmentation, since each pixel requires information about both, the individual instance of class "person" and the individual instance of the respective body part. To address such needs, *hierarchical* panoptic segmentation has emerged as a natural extension of the standard panoptic segmentation paradigm, to capture multi-level semantic and instance relationships [99].

While in the previous chapter we extended the standard panoptic segmentation formulation from the point of view of the input, as we wanted to effectively process the different sensor modalities together, in this chapter, we aim to extend it from the output point of view. Specifically, we address hierarchical panoptic segmentation to capture nested instance representations.

Plant phenotyping is a well-suited test ground for hierarchical panoptic segmentation. Its importance is paramount, as sustainable crop farming is fundamental to fulfilling the demand for food, fuel, and fiber while reducing its environmental impact. Plant phenotyping aims to accurately identify plants' growth stages and appearance, often to optimize management in the fields or support plant breeders with variety-specific information [189]. The first step in phenotyping is the perception of crops and weeds, which can be automated by robots. Additionally, information about the plant's growth and phenotypic traits can be exploited in automated intervention procedures and decision-making. One of the popular phenotypic traits is the number of leaves each plant has, which is one key aspect of assessing the growth stage and the need for fertilization [95], as well as



(a) Input RGB (b) Semantics (c) Plant instances (d) Leaf instances

Figure 4.1: Our approach takes as input RGB images from real fields and provides semantic, plant instance, and leaf instance segmentation. Different instances are shown with different colors.

for other delicate tasks such as precision intervention [20], where it is relevant to recognize finer structures.

In this chapter, we propose a solution to *hierarchical* panoptic segmentation in a crop field. Given an RGB image recorded by a UAV, our goal is to segment the scene at three different levels: crops at the semantic level, individual plants at the first instance level, and their leaves at the second instance level. An intuitive representation is shown in Figure 4.1. In the agricultural domain, vision-based approaches mostly target crop-weed classification [129, 134], plant [26], or leaf segmentation [204, 207] separately. These approaches ignore the underlying hierarchical relationship between them.

The main contribution of this chapter is a new approach for hierarchical panoptic segmentation that relies on convolutional neural networks using RGB data. For each pixel in the input, we predict the semantic class and, if it is a crop, to which plant and leaf instance it belongs. We solve the three tasks jointly, exploiting the underlying task hierarchy by means of a novel design of skip connections. In particular, semantic segmentation can support plant instance segmentation, which can further help leaf instance segmentation. We additionally propose an automatic post-processing strategy to aggregate the network's outputs and produce the instance mask. Thanks to the structure of our network and the post-processing, our approach yields a pixel-wise semantic, plant instance,

and leaf instance segmentation of the image data at the frame rate of a typical camera.

Our experiments suggest that (i) our approach can jointly perform semantic, plant instance, and leaf instance segmentation, i.e., hierarchical panoptic segmentation, on real-world data; (ii) our novel scheme for the skip connections better exploits the hierarchical connections between the tasks; and (iii) our improved post-processing achieves superior performance with respect to common state-of-the-art methods, while yielding end-to-end inference in real-time.

This work has been performed in equal contribution with Gianmarco Roggiolani. My main contribution regards the design of the architecture for hierarchical panoptic segmentation, and the proposed skip connections scheme, which are discussed in Section 4.2.1 and Section 4.2.2. His contribution lies in the novel post-processing operation, discussed in Section 4.2.3 and reported in this thesis for the sake of completeness. Experiments have been performed jointly.

4.1 Related Work

Over the last years, we have seen significant progress in the application of vision-based methods for semantic and instance segmentation in real agricultural settings. Deep learning architectures in the agricultural domain usually target only one specific task, while we address jointly semantic, plant instance, and leaf instance segmentation. This one-shot approach is more efficient, does not require individual networks for each task to run in parallel on the robot, and provides consistent results for the three tasks.

In the agricultural domain, CNNs are the gold standard to provide a pixel-wise classification of the input image. Lottes *et al.* [118] use as input the sequential images recorded by agricultural robots to exploit the spatial arrangement of the fields. McCool *et al.* [130], instead, focus on models that can achieve high accuracy and are lightweight to run easily on robotic platforms. This is due to the fact that low memory consumption is crucial for real-world applications, together with fast inference time, as also addressed by Milioto *et al.* [134]. They add the near-infrared (NIR) images as input next to the RGB images and compute multiple vegetation indices as pre-processing to support the training. NIR images are exploited by Bosilj *et al.* [19] as well, but their work focuses on how well the semantic segmentation performance transfers between different crop datasets, to reduce the amount of time and labels needed to train a network on new species. In contrast, Jeon *et al.* [82] use two architectures in parallel to learn different features that are exchanged during training. Its final result is an ensemble of the outputs. We do not focus on semantic segmentation only, and we experimentally show that performance benefits from sharing information between tasks.

Regarding instance segmentation for agriculture, most methods aim to detect and segment individual plants or leaves. Milioto *et al.* [133] propose a two-stage approach that first detects single plants and then feeds each one into a CNN classifier to distinguish whether it is a crop or a weed. Building on this idea of analyzing individual plant components, joint approaches for stem detection have also been explored [117]. Beyond stems, several methods have been proposed to distinguish individual leaves, highlighting the importance of fine-grained plant structure analysis in agricultural perception. For instance, Morris [139] exploits the differences in texture between leaf boundaries and interiors, segmenting them through a pyramid CNN. In contrast, Romera-Parades *et al.* [167] focus on the spatial arrangement of leaves, employing convolutional long short-term memory units to sequentially count them. Together, these approaches illustrate the diversity of strategies used to capture different structural elements of plants, ranging from whole-plant detection to detailed leaf-level analysis. One well-known approach for instance segmentation is Mask R-CNN [71]. It has a two-step procedure where the first step is object detection, and the second produces pixel-wise masks. Though the network is a general-purpose one, Champ *et al.* [26] investigated its performance for agriculture. The above-mentioned methods only detect plant or leaf instances, but not both jointly, thus limiting the information that we can extract from a single network. Weyler *et al.* [204] perform plant and leaf segmentation at the same time; they use a bottom-up approach where each plant can be seen as the union of the leaves. In contrast, our approach jointly segments plants and leaves in a top-down fashion, thus without predicting overlapping instance segments, as explained in Chapter 2.

Most of the methods that extract plant traits rely on data acquired in a laboratory, such as the CVPPP Leaf Segmentation Challenge [174] where each image presents only one plant. In this simplified setting, the approach by Kulikov *et al.* [93] detects leaves with a two-stage method which first predicts target embeddings with a CNN, and then clusters them. Another common way to deal with leaf counting in literature is by predicting salient points, as in the work from Itzhaky *et al.* [81]. They use a CNN to generate a heatmap of leaf keypoints that is fed to a non-linear regression model to predict the number of leaves per plant. Shi *et al.* [178] operate in a similar setting, performing semantic and instance segmentation using multiple images of single plants. They combine the predictions of the different viewpoints to 3D point clouds and refine the segmentation of leaves, stems, and nodes. In contrast, the approach presented by Weyler *et al.* [207] works under real field conditions, detecting the bounding box of single plants and per-plant leaf keypoints. This method, however, only provides coarse keypoints that are not suitable for determining leaf size and shape. In their follow-up work [204], the authors present a model to predict a pixel-wise plant and leaf in-

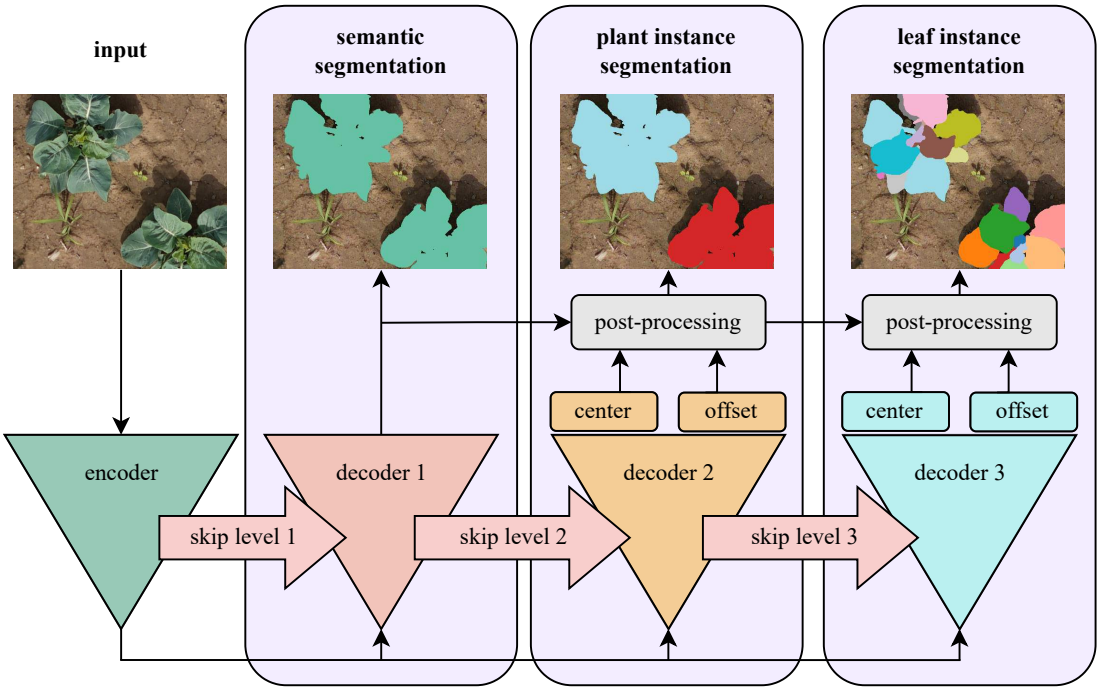


Figure 4.2: Overview of our architecture. The encoder takes an RGB image as input, processes it, and sends the resulting to the decoders. Skip connections are present in a hierarchical fashion after each downsampling and upsampling block.

stance segmentation, allowing for the extraction of relevant plant and leaf traits. The main novelty is the introduction of a learned covariance matrix for each instance center, which is crucial for post-processing and final instance detection. Our approach, in contrast, does not use covariance matrices but relies solely on the predictions of offsets and centers for both plants and leaves. Additionally, unlike the model from Weyler *et al.* [204], our approach has a dedicated decoder for semantic segmentation, which allows us to predict other classes than crops, such as weeds.

4.2 Our Approach to Hierarchical Panoptic Segmentation

The goal of our approach is to perform hierarchical panoptic segmentation, i.e., simultaneously solving semantic, plant instance, and leaf instance segmentation. We achieve this by means of an encoder-decoder CNN architecture with multiple decoders, which takes RGB images $I \in \mathbb{R}^{H \times W}$ as input. The decoders address the three tasks of semantic, plant instance, and leaf instance segmentation, as illustrated in Figure 4.2. Additionally, we employ encoder-decoder and decoder-decoder hierarchical skip connections for information flow.

4.2.1 Architecture

We use an ERFNet [166] encoder architecture and three ERFNet-based decoders, which allow us to have a lightweight network well-suited for real-time tasks. The semantic segmentation decoder has only one non-bottleneck-1D block after the deconvolutions, while the instance segmentation decoders have two, as defined in the original paper. Both encoder and decoders use the Gaussian error linear unit [74] activation function, as suggested in the work by Liu *et al.* [111].

The semantic segmentation decoder has a single output head with output dimension equal to the number of semantic classes and a softmax activation function. Training is performed with the Lovasz-Softmax loss [12], denoted as $\mathcal{L}_{\text{sem}}$. This loss is a convex surrogate of the IoU, designed to directly optimize segmentation quality by weighting prediction errors according to their effect on the IoU. For a detailed derivation and the complete mathematical formulation, we refer the reader to the original paper.

The instance segmentation decoders have two heads each, for centers and offsets prediction. The center prediction heads have an output dimension equal to 1 and a sigmoid activation function to predict pixel-wise probabilities of being a center. We define the center of an object as the internal pixel closest to its median point. We optimize it with a binary focal loss [135] $\mathcal{L}^i_{\text{cen}},\ i \in \{p,\, l\}$, where p and l denote plants and leaves:

$$\begin{aligned} \mathcal{L}^i_{\text{cen}} &= \frac{1}{|\Omega|} \sum_{p \in \Omega} \text{BFL}(\hat{c}_p,\, c_p), \\ \text{BFL}(\hat{c}_p,\, c_p) &= \begin{cases} -\alpha\,(1-\hat{c}_p)^{\tau}\,\log(\hat{c}_p) & \text{, if } c_p = 1 \\ -(1-\alpha)\,\hat{c}_p^{\,\tau}\,\log(1-\hat{c}_p) & \text{, if } c_p = 0 \end{cases}, \end{aligned} \tag{4.1}$$

where $c_p = \{0,\, 1\}$ is the binary ground truth variable indicating whether pixel p is a center or not, $\hat{c}_p$ is the center prediction at pixel p, α and τ are design parameters and are fixed in all experiments to 0.1 and 2, respectively.

The offset prediction heads output a 2D vector per pixel that points from the pixel location to the corresponding object center. These offsets are optimized with an L_1 regression loss $\mathcal{L}^i_{\text{off}},\ i \in \{p,\, l\}$, where p and l denote plants and leaves:

$$\mathcal{L}^i_{\text{off}} = \frac{1}{N} \sum_{p=1}^{N} \left\| \hat{\mathbf{o}}_p - \mathbf{o}_p \right\|_1, \tag{4.2}$$

where N is the number of pixels, $\hat{\mathbf{o}}_p \in \mathbb{R}^2$ is the predicted offset vector at pixel p, and $\mathbf{o}_p \in \mathbb{R}^2$ is the ground-truth offset pointing from pixel p to its object center.

Thus, the final loss function $\mathcal{L}$ is given by

$$\mathcal{L} = w_1\,\mathcal{L}_{\text{sem}} + w_2\,\mathcal{L}^p_{\text{cen}} + w_3\,\mathcal{L}^l_{\text{cen}} + w_4\,\mathcal{L}^p_{\text{off}} + w_5\,\mathcal{L}^l_{\text{off}}, \tag{4.3}$$

where w_i are scalar weights for the different terms.

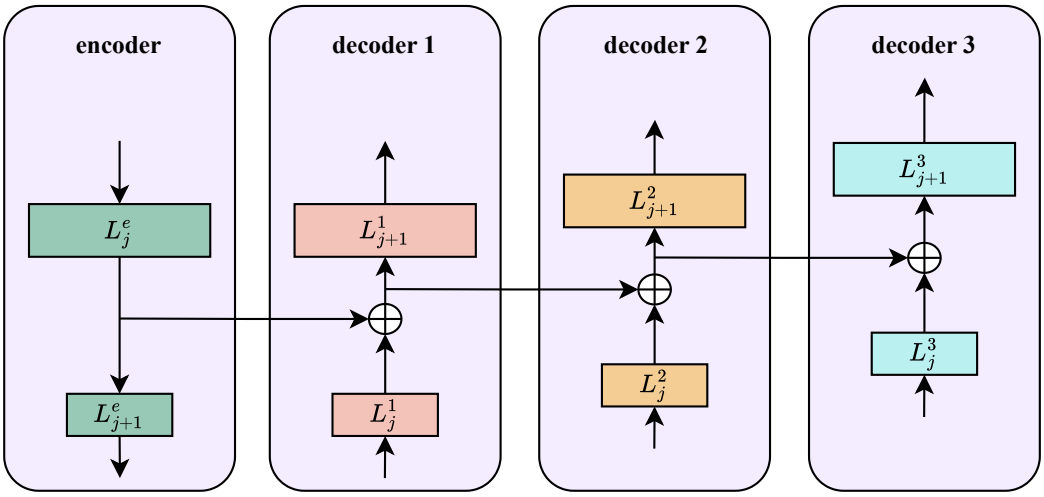


Figure 4.3: Visual breakdown of the hierarchical skip connection scheme. Only one downsampling stage for the encoder and one upsampling stage for all decoders are shown for simplicity. L_j indicates the j-th layer of the corresponding branch. The superscripts indicate either the encoder (denoted with e) or a decoder (denoted with 1 , 2, or 3), following the nomenclature introduced in Figure 4.2. Each decoder takes a skip connection composed of the features extracted by the encoder summed to the ones retrieved by all previous decoders, if any.

4.2.2 Skip Connections

Low-level spatial information can be lost during the downsampling stage of the encoder. Skip connections tackle this issue by skipping one or more layers and providing a direct gradient flow from late to early stages. This is fundamental in several architectures, as it ensures feature reusability and solves the degradation problem of deep models [149]. The common usage of skip connections in segmentation models is inspired by the UNet model of Ronneberger *et al.* [168], where the higher-resolution feature maps of the encoder are concatenated with the feature maps of the decoder. While nowadays the concatenation is usually replaced by summation, in order to have a symmetrical network structure, the concept of transferring information from encoder to decoder remains the most common design choice of modern segmentation architectures [87, 102, 187, 204].

In this work, we suggest a new scheme for skip connections that takes into account the relations between the different tasks we address. We propose to connect directly different decoders, rather than the encoder and decoders only, to improve segmentation performance. A visual breakdown of this novel scheme is shown in Figure 4.3. Our skip connections propagate feature maps after the first two downsampling operations of the encoder to the decoders, after the corresponding upsamplings, as shown in the figure. In particular:

1. For the semantic segmentation task, we keep the skip connections from the

encoder to the decoder, since spatial information coming from the features extracted at higher resolutions helps the decoder.

2. Plant instance segmentation aims to distinguish each pixel classified as crop in a specific instance, so the information coming from the semantic task is more helpful than the features coming from the encoder only. Thus, we sum the contribution coming from the semantic segmentation decoder to focus on the relevant regions of interest.

3. The objective of leaf instance segmentation is to discriminate individual leaves in each plant. To achieve this, knowing the position of each distinct plant is more helpful than just knowing where crops are or feature maps from earlier stages. Thus, analogously to before, we augment the skip connections with the contribution from the plant instance decoder.

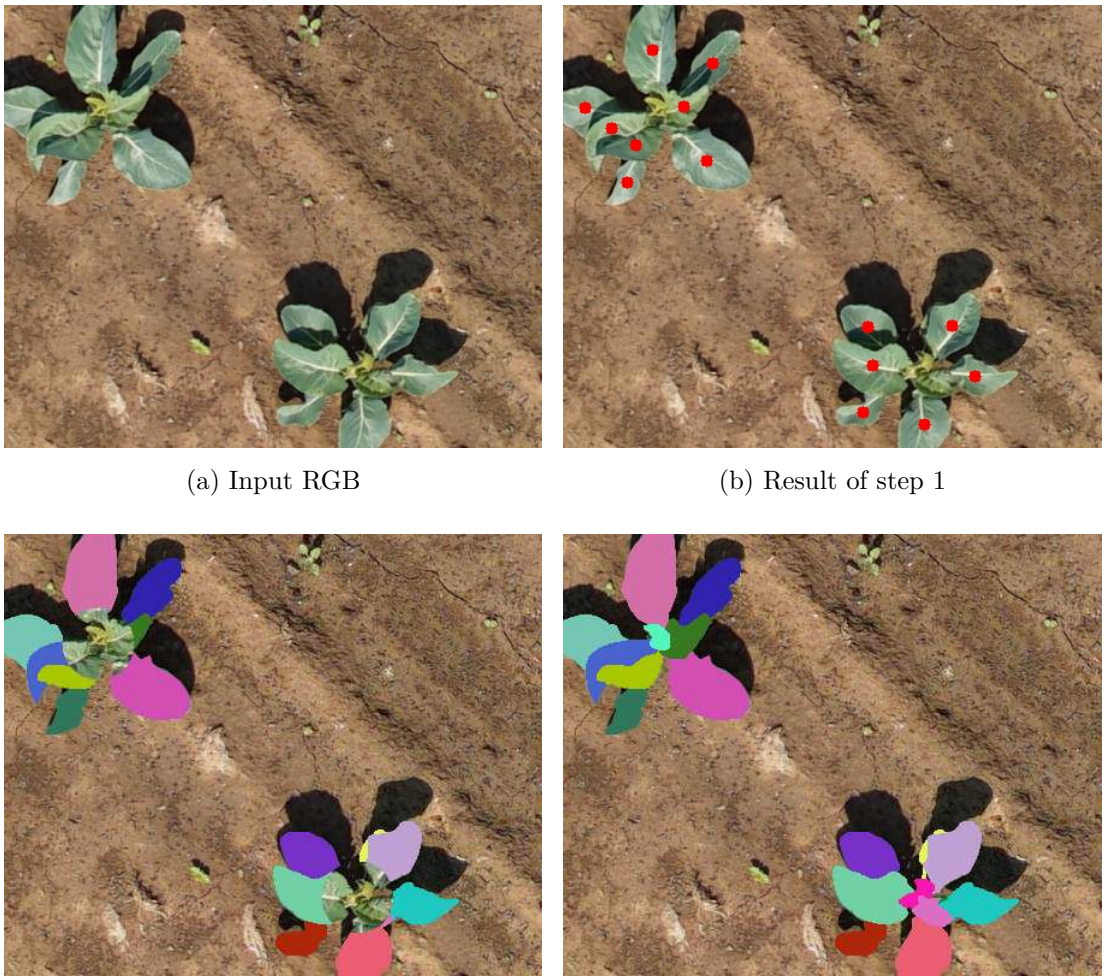
To better illustrate the information flow in our hierarchical panoptic segmentation architecture, consider the process as assembling segmentation in layers. The encoder provides a basic high-resolution spatial guide. The semantic decoder adds category labels to it. The plant instance segmentation decoder builds upon this enriched map, identifying individual crops as landmarks. Finally, the leaf instance segmentation decoder refines this by adding finer distinctions, the individual leaves, using all previously gathered information. At each stage, skip connections act like reference notes, ensuring that each level benefits from both low-level detail and high-level context, progressively enriching the representation.

This newly-proposed skip connection scheme directly exploits the underlying hierarchy between the tasks, designed to realize a meaningful transfer of features from one branch to the other. Extensive experiments reported in Section 4.3.2 suggest that these task skip connections lead to superior performance.

4.2.3 Post-Processing

Section 4.2.3 introduces a novel post-processing operation and is the latest contribution of this chapter. It has been developed by Gianmarco Roggiolani, and we report it here for the sake of completeness.

Our automatic post-processing consists of three steps, detailed in Figure 4.4 (a). The first step is inspired by Panoptic DeepLab [30], and has the objective to extract a single center for each object. Specifically, we take the center prediction coming from the decoder and filter it with the predicted semantic mask to discard any center that does not belong to the class of interest. Since the center prediction head usually outputs blobs around the desired center, we perform a non-maximum suppression operation in order to reduce each blob to a single pixel, as shown in Figure 4.4 (b).



(a) Input RGB

(b) Result of step 1

(c) Result of step 2

(d) Result of step 3

Figure 4.4: Visual workflow of our three-steps post-processing: (a) the input image, (b) the predicted leaf centers after non-maximum suppression in step 1, (c) leaf instance segmentation result after step 2, (d) final leaf instance segmentation after step 3.

Afterwards, we need to assign each pixel to its center, which defines the individual instance. However, the offsets could point to regions of space close to more than one center. In the second step, we assign only those pixels whose offsets point to a single center. To this end, we build an image of coordinates, where each pixel p is a tuple of values (h, w), with $h \in \{1, \ldots, H\}, w \in \{1, \ldots, W\}$. We compute the Euclidean distance between this image and every ground truth object center pixel c, producing for each center a distance map $\boldsymbol{D}^c$. Then, we compute a predicted distance map $\hat{\boldsymbol{D}}$ from the offsets. When the offsets point close to a center pixel c, we expect the predicted distance map to be similar to $\boldsymbol{D}^c$. Thus, defining a distance threshold τ, one pixel p is assigned to the instance with center pixel c if

$$\left|\left|\boldsymbol{D}_p^c - \hat{\boldsymbol{D}}_p\right|\right| \leq \tau, \tag{4.4}$$

holds for that instance only. The instance mask we have at this point is displayed in Figure 4.4 (c).

The third step takes care of the pixels that were not assigned to any instance. This can happen if their offset points too far from every extracted center or are close to more than one. In this case, we use a voting mechanism. We compute the instance label that occurs the most between the N closest neighbors and assign it to the current pixel. The outcome of the automatic post-processing can be seen in Figure 4.4 (d). To enforce consistency between the masks, we filter all post-processing results with the semantic segmentation masks.

4.3 Experimental Evaluation

We present our experiments to show the capabilities of our method for joint semantic, plant instance, and leaf instance segmentation of RGB data. The results of our experiments show that: (i) our approach can jointly perform semantic, plant instance, and leaf instance segmentation, i.e., hierarchical panoptic segmentation, on real-world data, (ii) our novel scheme for the skip connections better exploits the hierarchical connections between the tasks; and (iii) our improved post-processing achieves superior performance with respect to common state-of-the-art methods.

4.3.1 Experimental Setup

Datasets and metrics. Originally, we tested our method on two RGB datasets: a sugar beets dataset introduced by Weyler *et al.* [204], denoted as SugarBeets, and GrowliFlower by Kierdorf *et al.* [84]. SugarBeets is composed of 1,316 images with size 512 px × 1024 px. The images are recorded with an UAV equipped with a PhaseOne iXM-100 camera mounted in nadir view. GrowliFlower is a dataset of cauliflower images taken in agricultural fields. It is composed of 2,198 images with size 368 px × 448 px. The images are recorded with an UAV equipped with a Sony A7 rIII RGB camera and a MicaSense 5CH for multispectral image data. Both datasets provide an official data split that we adopt. Here, for completeness, we also report results on the PhenoBench dataset [206], which was not yet published when we originally developed this work. PhenoBench is composed of 2,872 images of sugar beets recorded at different growth stages with a PhaseOne iXM-100 camera mounted on a DJI M600 drone, with size 1024 px × 1024 px. Additionally, PhenoBench is the only dataset among the ones reported in this chapter that explicitly provides ground truth semantic annotations for weeds, which we include in our evaluation.

For quantitative evaluation, we employ the metrics introduced in Section 2.3

Table 4.1: Hierarchical panoptic segmentation results on the test set of the SugarBeets dataset. P and L stand for plant and leaf instance segmentation, respectively. Best results in bold. All metrics are reported as percentages (%), except Params and FPS.

Model	P	L	IoU	PQ_P	PQ_L	Params	FPS
Mask R-CNN [71]	✓		46.2	47.8	-	43.9M	13.5
Panoptic DeepLab S [30]	✓		75.4	69.4	-	7.7M	93.5
Panoptic DeepLab M [30]	✓		75.5	69.8	-	55.3M	4.7
Panoptic DeepLab L [30]	✓		76.4	71.1	-	69.6M	48.4
Mask R-CNN [71]		✓	64.9	-	53.6	43.9M	13.4
Panoptic DeepLab S [30]		✓	75.4	-	50.8	7.7M	93.7
Panoptic DeepLab M [30]		✓	76.7	-	54.4	55.3M	49.1
Panoptic DeepLab L [30]		✓	76.3	-	52.9	69.6M	48.5
Weyler [204]	✓	✓	75.3	72.3	63.1	2.25M	0.14
Ours	✓	✓	**79.3**	**76.2**	**63.5**	2.4M	26.3

and, for evaluating semantic segmentation performance, we compute the intersection over union [51] of the "crop" class. For PhenoBench, we also report the IoU of the "weed" class, which we call "IoU_W". For the plant and leaf instance segmentation, we evaluate our method by means of the panoptic quality [88], which we call PQ_P for the plants and PQ_L for the leaves.

Training details and parameters. In all experiments, we use AdamW [115] without weight decay with an initial learning rate of $5 \cdot 10^{-4}$ for the encoder and the semantic decoder and $8 \cdot 10^{-4}$ for the instance decoders, and train for 500 epochs. We initialize our network with the Xavier initialization [60]. The batch size is set to 1. We resize images from the SugarBeets dataset to $256\,\text{px} \times 512\,\text{px}$, and from PhenoBench to $512\,\text{px} \times 512\,\text{px}$, to keep the aspect ratio. No resize is applied to the GrowliFlower dataset. Additionally, we set $w_1 = 1$, $w_2 = w_3 = 0.1$, and $w_4 = w_5 = 50$ in Equation (4.3) to balance the magnitude of the individual loss functions, while in the post-processing we use number of neighbors $N = 5$, grouping threshold $\tau = 6$ for the plant instance segmentation and 2 for the leaf instance segmentation. We tuned all hyperparameters on the validation sets.

4.3.2 Hierarchical Panoptic Segmentation

The first experiment evaluates the performance of our approach, and its outcomes show that it can effectively address hierarchical panoptic segmentation, i.e., jointly provide pixel-wise semantic, plant instance, and leaf instance segmentation. Table 4.1 and Table 4.2 show the IoU for the crops and the panoptic

Table 4.2: Hierarchical panoptic segmentation results on the test set of the GrowliFlower dataset. P and L stand for plant and leaf instance segmentation, respectively. Best results in bold. All metrics are reported as percentages (%), except Params and FPS.

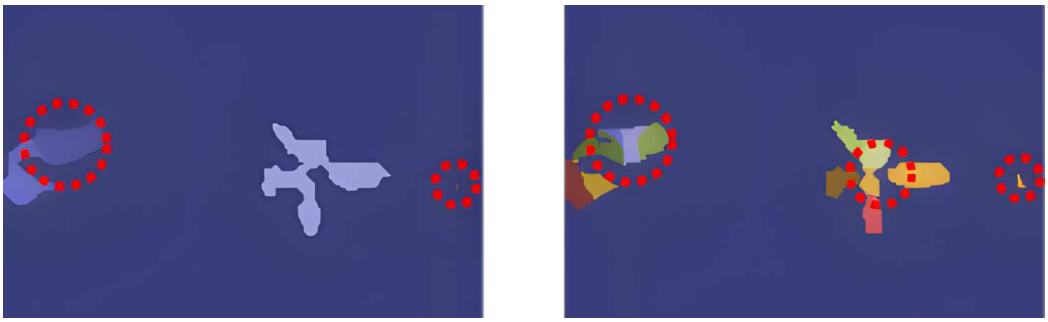
Model	P	L	IoU	PQ_P	PQ_L	Params	FPS
Mask R-CNN [71]	✓		25.4	27.9	-	43.9M	9.6
Panoptic DeepLab S [30]	✓		83.1	69.9	-	7.7M	43.4
Panoptic DeepLab M [30]	✓		82.0	68.0	-	55.3M	47.6
Panoptic DeepLab L [30]	✓		82.7	69.4	-	69.6M	23.8
Mask R-CNN [71]		✓	53.8	-	41.0	43.9M	16.2
Panoptic DeepLab S [30]		✓	**84.4**	-	58.8	7.7M	76.5
Panoptic DeepLab M [30]		✓	80.2	-	43.4	55.3M	41.6
Panoptic DeepLab L [30]		✓	82.8	-	50.1	69.6M	30.3
Weyler [204]	✓	✓	65.8	67.8	69.4	2.25M	0.25
Ours	✓	✓	80.2	**89.2**	**71.0**	2.4M	20.7

quality for both plant and leaf instances for SugarBeets and GrowliFlower, respectively. We also report the number of parameters of the networks and the end-to-end frame rate of each method at inference time (FPS). We compare our method against Mask R-CNN [71], which is a common approach in the agricultural domain, and Panoptic DeepLab [30], which is a state-of-the-art model for panoptic segmentation. We use three variants of Panoptic Deeplab with different backbones: a small model that uses MobileNetV2 [172], called Panoptic DeepLab S in the tables, a medium-size model with ResNet50 [72], called Panoptic DeepLab M, and a big-size model with Xception65 [34], named Panoptic DeepLab L.

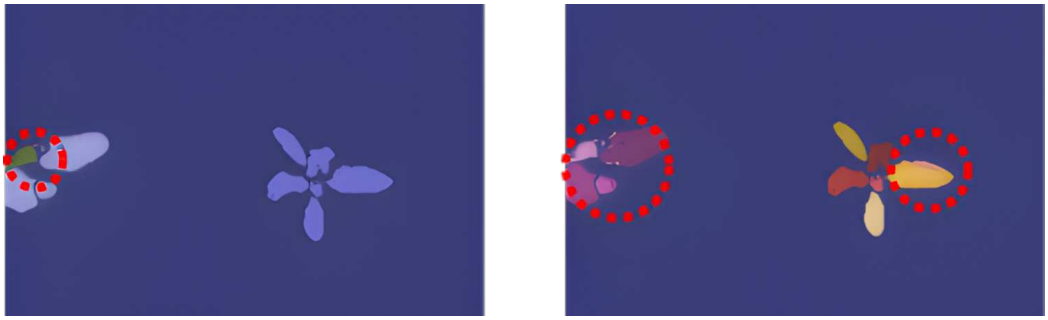
All these baselines, however, can only address one instance segmentation task at a time and, thus, they need to be trained for either plants-only or leaves-only. We also compare with the work from Weyler *et al.* [204] denoted as Weyler, which addresses both, plant and leaf instance segmentation.

Interestingly, the models that tackle all tasks are also the smallest ones in terms of number of parameters, which makes it more suitable to run on resource-constrained robotic systems. The semantic segmentation decoder, which is our first output and filters the following predictions, is the reason behind the extra parameters compared to Weyler. Our approach is suitable for real-time operations with a frame rate that exceeds 20 Hz. All baselines have worse segmentation performance on all the tasks. Additionally, most of them need two models to perform all tasks that we tackle with our network, which also means that the same

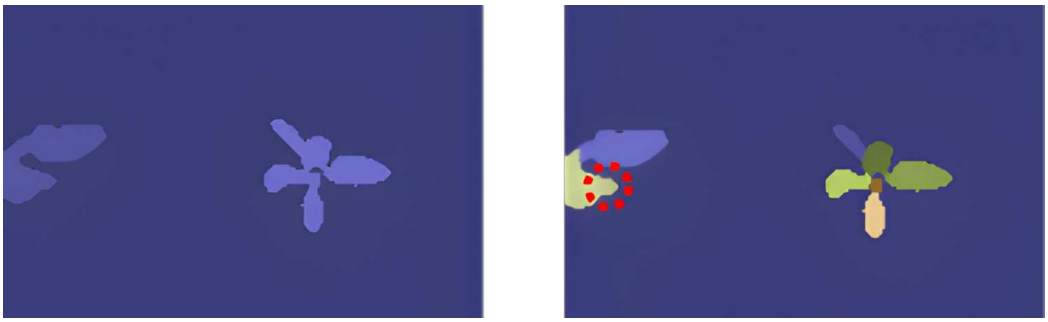
(a) Input RGB and ground truth leaf instance segmentation



(b) Results from Panoptic DeepLab [30]



(c) Results from Weyler *et al.* [204]



(d) Our results

Figure 4.5: Qualitative results on the SugarBeets dataset for plant (left) and leaf (right) instance segmentation. We show some segmentation errors in the red circles.

Table 4.3: Hierarchical panoptic segmentation results on the test set of the PhenoBench dataset. P and L stand for plant and leaf instance segmentation, respectively. Best results in bold. All metrics are reported as percentages (%).

Model	P	L	IoU	IoU_W	PQ_P	PQ_L
Weyler [204]	✓	✓	-	-	38.4	42.6
Ours	✓	✓	85.1	61.1	**54.6**	**46.8**

RGB image needs to pass through two models that do not share parameters and two post-processing operations that do not ensure consistency of the results. The only baseline that addresses both, the plant and leaf instances with one network is Weyler *et al.* [204], which is not suitable for real-time operations due to its relatively low framerate of 0.14 Hz on SugarBeets and 0.25 Hz on GrowliFlower.

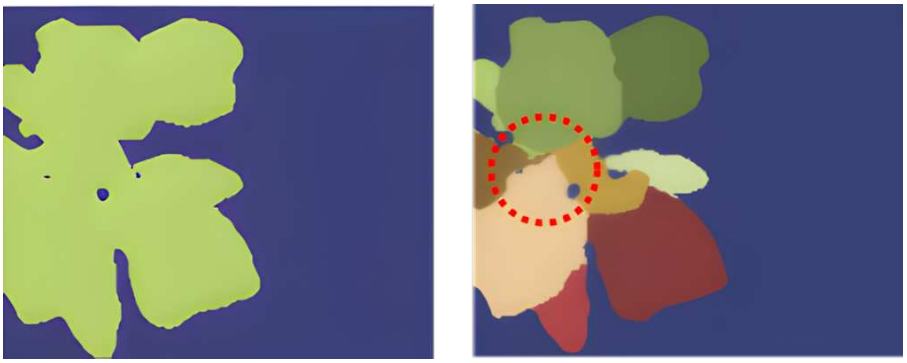
In sum, our model with specifically-designed skip connections and novel automatic post-processing operations outperforms state-of-the-art architectures on all tasks on the SugarBeets dataset. On GrowliFlower, our model substantially outperforms all baselines on instance segmentation tasks. Additionally, our model is able to run at the frame rate of common RGB cameras. Qualitative results are shown in Figure 4.5 for SugarBeets, and in Figure 4.6 for GrowliFlower.

To further evaluate our approach, we report results on PhenoBench [206] in Table 4.3. PhenoBench is more challenging than SugarBeets and GrowliFlower, as it contains very different growth stages as well as highly different levels of weeds in the field. Here, we compare only with the approach from Weyler *et al.* [204] as it is the only one that targets both segmentation tasks simultaneously. However, this baseline is a bottom-up approach that first predicts leaves, which are then associated with a plant, and does not explicitly target semantic segmentation. While the other datasets we mentioned do not contain weeds, and so we could infer semantic segmentation performance by grouping all the pixels that belong to instances into a single semantic class "crop", this would not be meaningful here anymore. Thus, we do not report semantic segmentation performance for this baseline. Our results on PhenoBench are aligned with the ones we achieved on the other datasets in terms of semantic segmentation, while being inferior in terms of panoptic quality, despite still being superior to the baseline. This suggests that our hierarchical approach is still effective, also on a more challenging dataset.

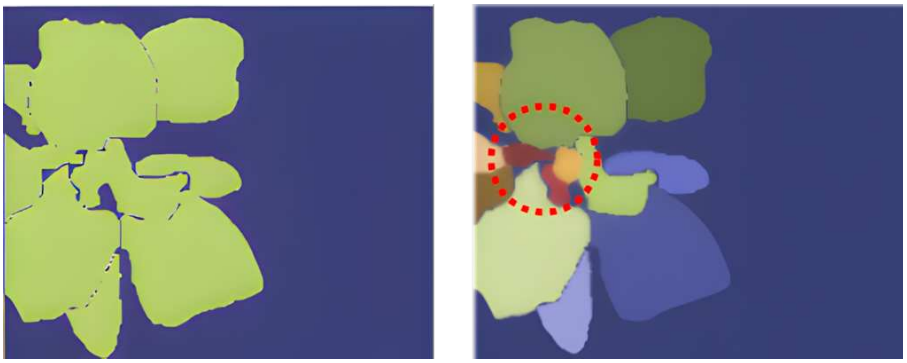
4.3.3 Ablation Studies

In this section, we provide ablations to show the improvements provided by the skip connections scheme and the post-processing operations. We perform all ablations on the validation set of the SugarBeets dataset.

(a) Input RGB and ground truth leaf instance segmentation



(b) Results from Panoptic DeepLab [30]



(c) Results from Weyler *et al.* [204]

(d) Our results

Figure 4.6: Qualitative results on the GrowliFlower dataset for plant (left) and leaf (right) instance segmentation. We show some segmentation errors in the red circles.

Table 4.4: Comparison between different skip connections and different post-processings. PD stands for Panoptic DeepLab. Best results in bold. All metrics are reported as percentages (%).

	Skip Connections	Attached	Post-proc	IoU	PQ_P	PQ_L
A	None		Ours	84.4	75.5	65.1
B	Encoder	✓	Ours	83.3	79.4	65.6
C	Task + Encoder		Ours	83.2	78.9	65.6
D	Task	✓	Ours	84.4	81.1	66.0
E	Task + Encoder	✓	PD [30]	**84.5**	79.0	47.2
F	Task + Encoder	✓	Ours	**84.5**	**81.7**	**67.8**

Table 4.5: Comparison between our skip connections and the UNet-like [168] encoder-decoder scheme for the standard panoptic segmentation task (i.e., semantic segmentation and a single instance segmentation). Best results in bold. All metrics are reported as percentages (%).

	Skip Connections	Attached	Post-proc	IoU	PQ_P
G	Encoder	✓	Ours	85.1	81.0
H	Task + Encoder	✓	Ours	**85.8**	**82.6**

We thoroughly evaluate the design of our novel skip connection scheme by comparing it to other ways to connect them, as well as to the standard UNet [168] scheme, and show the results in Table 4.4. In particular, we use the exact same network as the one we propose with no skip connections (A); typical encoder-decoder skip connections [168] (B); hierarchical skip connections with no gradient flow (C); skip connections without summing the contribution from the encoder (D). When we do not use any skip connection, the panoptic qualities are noticeably lower, because the corresponding decoders have no help from previous features. In the case of encoder-decoder skip connections, the panoptic qualities are better since the decoders get features from the encoder that has to compromise between all tasks, harming the semantic segmentation. Interestingly, we notice no improvement from the hierarchical skip connections with no gradient flow, where skip connections are detached and thus they do not participate in the backward pass, since the feature flow does not play any role in the optimization, leading to suboptimal performance. On the other hand, hierarchical skip connections without the encoder contribution substantially improve performance with respect to the skip connections from the encoder only. This suggests that decoder features are more relevant than restoring features from the encoder when it comes to tasks that present an underlying hierarchical structure. In the same

table, we also show the performance of our best model with the post-processing from Panoptic DeepLab (E). Clearly, a different post-processing does not play any role in the semantic segmentation, and our post-processing substantially improves instance segmentation, especially when clustering leaves (F).

Our last ablation study focuses on the standard panoptic segmentation problem: semantic and single instance segmentation. We keep the plant instance decoder in order to maintain the hierarchy of our approach. As we can see in Table 4.5, we evaluate our skip connections scheme against the commonly-used encoder-decoder strategy coming from UNet [168] (G). The experiment confirms that our skip connection scheme (H) exploits the hierarchy between the tasks better, leading to superior segmentation performance.

4.4 Conclusion

In this chapter, we introduced a novel approach for hierarchical panoptic segmentation of RGB images in the agricultural domain. Our approach enables fine-grained and structured scene understanding by jointly modeling multiple levels of semantic and instance segmentation in a coherent hierarchy. We explicitly leverage the inherent hierarchical relationship among the individual segmentation tasks through a specifically designed skip-connection scheme, which enables information to flow not only across layers of the network, but also across tasks in a way that mirrors their semantic dependencies. We carefully designed and validated the hierarchical structure of our CNN, paying particular attention to the way skip connections are arranged. In our experiments, we demonstrate that our proposed design better captures the structured nature of the problem, leading to improved performance across multiple metrics. To further substantiate the benefits of our approach, we benchmarked it against several common baselines in the agricultural vision literature. Remarkably, our model achieved superior results even when compared to approaches that produce only one instance segmentation task at a time. This highlights that jointly modeling task hierarchy not only reduces redundancy but also provides complementary information that benefits all segmentation subtasks simultaneously.

All in all, our experiments provide strong evidence that exploiting task hierarchy is a powerful strategy for boosting segmentation performance in complex agricultural scenes. The success of this approach, however, is not confined to RGB imagery. In the next chapter, we extend these ideas to the domain of 3D point clouds, demonstrating that the advantages of hierarchical reasoning generalize beyond image data and hold across sensing modalities.

Chapter 5

3D Hierarchical Panoptic Segmentation

As discussed in the previous chapter, the task of hierarchical panoptic segmentation adds a layer of complexity with respect to standard panoptic segmentation, and is of crucial importance when nested instances appear, such as segmenting all individual crops in a field, and then further segmenting the individual leaves of each crop plant. We addressed this problem by designing a CNN with multiple decoders, one for each segmentation subtask, where the skip connections play a fundamental role. In fact, rather than just allowing information flow from the encoder to the decoders, the decoders also exchange information among them, exploiting the underlying hierarchy among the various nested segmentation tasks. This intuition, which we experimentally validated in Chapter 4, is not, however, especially tailored to 2D data and, thus, 2D convolutions. In this chapter, we apply it to the 3D domain and 3D convolutions. The standard 3D panoptic segmentation task is not different from its 2D formulation we discussed in Chapter 2, except for the format of the input data, which is now a 3D point cloud rather than a 2D image. Consequently, all considerations made still apply.

Plant phenotyping has been a good application area for 2D hierarchical segmentation, primarily because crops are commonly monitored from above using drones, resulting in 2D imagery. Consequently, numerous methods have been developed for 2D phenotyping [117, 134, 207] and, thus, many datasets supporting this line of research have been released [68, 84, 206]. However, 3D datasets of real agricultural environments are not common, and are usually not suited for evaluating 3D hierarchical segmentation due either to their extremely limited size [48], or to the fact that they isolate instances, basically removing the need for a first-level instance segmentation [128, 176].

In this chapter, we look at another important task in agriculture, namely

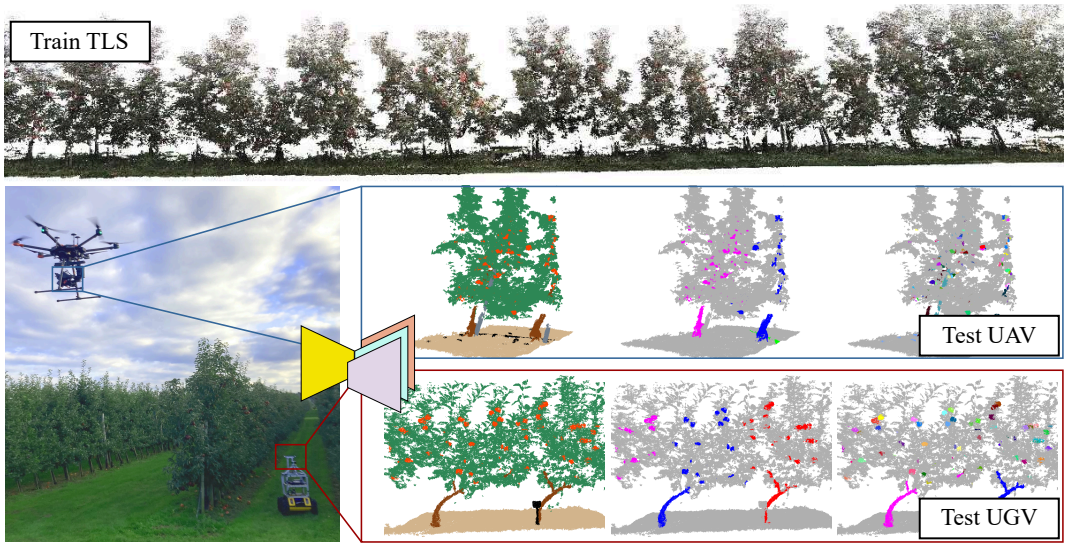


Figure 5.1: View of the apple orchard (top row) recorded with a terrestrial laser scanner (TLS), and used for training our network for hierarchical panoptic segmentation. The test set is composed of different splits corresponding to different sensors and robots. The two examples shown in the image are a UAV equipped with a PhaseOne iXM-100 camera, and a UGV equipped with a RealSense d435i camera.

yield estimation. An accurate yield estimate can help farmers make management decisions to increase crop production and optimize key factors such as harvest time and fertilizer use [98, 194, 216], as well as enable precision interventions [15]. Robots can automate or support many of these interventions, but to do so reliably and robustly, they need to understand their surroundings through the interpretation of sensor data. Orchards provide a particularly relevant testbed in this regard: they are commonly scanned with 3D sensors to capture their intricate structure, encompassing foliage, branches, and fruits. In this regard, tasks such as segmenting, counting, and localizing fruits in orchards are crucial in horticulture, as they provide essential information for monitoring and managing crop production, as well as supporting extremely labor-intensive processes such as fruit picking [21]. Additionally, unlike arable crops observed from overhead imagery, fruit trees require capturing their complex 3D structure, including foliage, branches, and fruit, making them a practical testbed for 3D perception tasks. Thus, we address the problem of 3D hierarchical panoptic segmentation in a real apple orchard.

The main contributions of this chapter are twofold. First, we propose an approach based on a convolutional neural network for 3D hierarchical panoptic segmentation, which is able to address multiple instance segmentation tasks at once while exploiting their underlying hierarchical relationship by means of a novel skip connections scheme. Second, to validate our approach, we release a

3D point cloud dataset of real apple orchards, called HOPS, short for hierarchical orchard panoptic segmentation, recorded with various sensors, such as terrestrial laser scanners, RGB-D cameras, and RGB cameras with subsequent bundle adjustment. We recorded data of the apple orchard at different growth stages across several crop rows in the span of two years. Afterwards, to enable 3D hierarchical panoptic segmentation, we labeled the resulting point clouds to obtain semantic, tree instance, and fruit and trunk instance annotations. Specifically, in the tree instance segmentation, each trunk has a label that is shared with all fruits belonging to it. Thus, in this instance-level task, fruits are not individually labeled but are rather assigned to a trunk. This is important because it gives an idea of the yield of each tree. The next instance segmentation level aims to separate the trunk from its fruits and segment the apples individually, providing a different instance for each individual trunk and fruit. We show a portion of the dataset and an exemplary image of the approach we propose in Figure 5.1.

Our experiments suggest that (i) our approach can jointly perform 3D hierarchical panoptic segmentation on real-world 3D data, while being able to generalize to data acquired with different sensors; and (ii) our skip connection scheme allows the CNN to exploit the underlying hierarchical relationship between the individual segmentation tasks, leading to better segmentation performance.

5.1 Related Work

Semantic scene interpretation is often required for realizing robotic automation in agriculture, since it identifies task-relevant objects in the scene that enables applications such as monitoring [116, 150, 182] and intervention [4, 96, 123].

While often images are used for panoptic segmentation in the agricultural domain [207], 3D point clouds generated by RGB-D cameras, LiDARs, or photogrammetric structure-from-motion techniques received increasing research interest [235] due to the capability to extract precise geometric information required for high-precision fruit grasping [96, 123], plant segmentation [137], and fine-grained trait estimation [127]. These applications benefit from 3D data's robustness to occlusions, scale variations, and illumination changes, challenges that are difficult to overcome with 2D imagery alone. 3D panoptic segmentation has been widely investigated in the domain of autonomous driving [132, 180, 234], where most approaches use dedicated branches for semantic and instance segmentation using features of a shared encoder. However, agricultural environments pose different challenges compared to urban scenes: plant structures are non-rigid and usually overlapping. Additionally, in plant phenotyping, we can exploit the hierarchical structure of plants that can be decomposed into individual parts [63], such as plant, leaf, and even more fine-grained leaf structure, to

train segmentation models. Prior work, as the hierarchical panoptic segmentation discussed in Chapter 4, explored this hierarchy using 2D imagery, employing decoder-level skip connections to propagate features across levels of the hierarchy for improved bottom-up instance prediction [165], which is the foundation of our approach to 3D panoptic segmentation.

In contrast to prior approaches for 3D panoptic segmentation, we also explicitly consider the hierarchical structure of the prediction task by means of our novel skip connection scheme, which allows us to predict a hierarchical semantic scene interpretation consisting of tree instances as well as fruit instances that can be used for plant monitoring and yield estimation in the agricultural domain.

While most agricultural datasets for semantic scene interpretation provide only RGB images [120], 3D datasets have recently become available. Chaudhury *et al.* [28] proposed a synthetic dataset generated based on plant models with the addition of noise. While being able to generate large amounts of data, the gap between simulation and reality poses a challenge for the deployment of approaches developed on this data to real fields. The Pheno4D dataset [176] provides organ-level annotation of point clouds of real tomato and maize plants at different growth stages acquired with a high-precision laser scanner. Similarly, the TomatoWUR dataset [195] contains high-resolution point clouds of tomato plants, labeled with semantic and instance annotations. However, these datasets were acquired in a lab setting under controlled conditions, which limits their applicability to real agricultural fields. Dutagaci *et al.* [48] provide a dataset of x-ray imaging of rose plants, but it contains an extremely limited number of samples. BUP20 [182], instead, provides an RGB-D dataset for instance segmentation of sweet peppers in a glasshouse [183]. For 3D plant phenotyping in real agricultural fields, BonnBeetClouds [128] provides structure-from-motion point clouds of sugar beet plants with plant and leaf instance annotations. The Crops3D dataset [235] provides organ-level annotations of different crop varieties for point cloud data acquired with a terrestrial laser scanner (TLS) in the field and single plant point clouds using structure-from-motion of RGB images or using structured light cameras. These datasets are all characterized by a relatively limited number of samples, or are not well-suited for 3D hierarchical panoptic segmentation due to the fact that they isolate instances, removing the need for multi-level instance segmentation.

In contrast to existing agricultural datasets for semantic scene interpretation, we provide a domain-specific agricultural dataset of apple orchards recorded with different sensors. We provide annotations for 3D hierarchical panoptic segmentation that include semantic segmentation, tree instance segmentation, i.e., a trunk and all of the apples belonging to it, and standard instance segmentation, i.e., individual trunks and apples. We provide high-resolution point

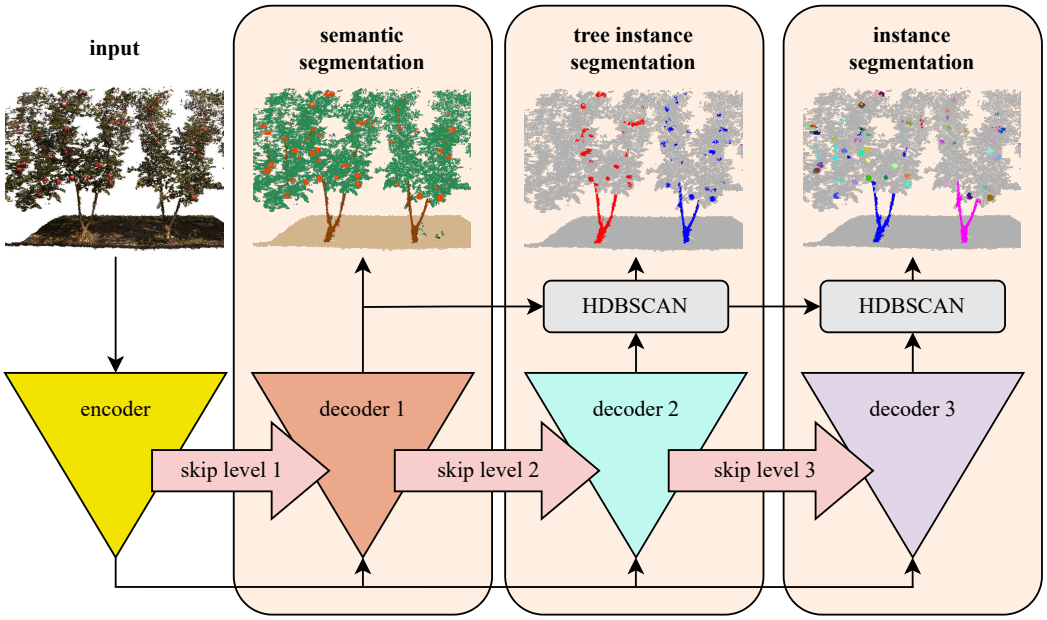


Figure 5.2: The encoder takes the colored point cloud as input, and the resulting features are processed by the decoders. We use hierarchical skip connections after each downsampling and upsampling block of the encoder and decoders. We use HDBSCAN to obtain instances.

clouds recorded with a TLS, but also structure-from-motion point cloud data acquired with robotic platforms equipped with a high-resolution RGB camera or a consumer-grade RGB-D sensor commonly used in robotics.

5.2 Our Approach to 3D Hierarchical Panoptic Segmentation

We propose an approach for hierarchical panoptic segmentation, i.e., the task of simultaneously performing semantic segmentation and multiple instance segmentation tasks with underlying hierarchical relationships. The network we use is an encoder-decoder architecture, and the decoders address semantic segmentation and two levels of instance segmentation, as illustrated in Figure 5.2. The first instance segmentation aims to identify "tree" instances, where a tree is defined by a trunk and all fruits belonging to it. The second instance segmentation looks for all individual instances, i.e., both trunks and fruits, in the point cloud.

5.2.1 Network Architecture

For our approach to 3D panoptic segmentation, we follow best practices and use a MinkUNet-based neural network [35]. This kind of neural network is inspired

by the UNet [168] design, where encoder and decoder are connected by skip connections. To process 3D data, MinkUnet networks use sparse 3D convolutions to process the input data. Specifically, we use the MinkUNet14A model, which allows us to have a lightweight network. In fact, MinkUNet14A is a modification of the standard MinkUNet14 model and has fewer feature channels per layer, allowing faster computation. We keep the original structure of the network, and replicate the original decoder two additional times. This allows us to have three identical decoders addressing the three segmentation tasks we aim to tackle, which is a convenient solution for exploiting the underlying hierarchy among them.

The first decoder targets semantic segmentation, and has a single output head with depth equal to the number of semantic classes that appear in the dataset. It is optimized using the standard weighted cross-entropy loss:

$$\mathcal{L}_{\text{sem}} = -\frac{1}{|\mathcal{S}|} \sum_{\boldsymbol{p} \in \mathcal{S}} \omega_k \, \boldsymbol{t}_p^{\top} \, \log(\sigma(\boldsymbol{f}_p)), \tag{5.1}$$

where $|\mathcal{S}|$ is the number of points in the point cloud, $\boldsymbol{p}$ indicates the individual point, ω_k is a class-wise weight computed via the inverse frequency of each class in the dataset, where $k \in \{1, \ldots, K\}$ indicates the semantic class, $\boldsymbol{t}_p \in \mathbb{R}^K$ is the one-hot encoded ground truth annotation at point location $\boldsymbol{p}$, $\sigma(\cdot)$ denotes the softmax operation, and $\boldsymbol{f}_p$ denotes the pre-softmax feature predicted at point $\boldsymbol{p}$.

The second decoder targets tree instance segmentation, i.e., the task of finding tree instances where a tree is defined as a trunk and all apples belonging to it. This decoder also has a single output head for offset prediction. Thus, each point predicts a 3D displacement towards a location in the 3D space that facilitates instance separation via clustering. We predict an offset $\boldsymbol{o}_p \in \mathbb{R}^3$ for each point $\boldsymbol{p}$, so that $\boldsymbol{e}_p = \boldsymbol{p} + \boldsymbol{o}_p$ is the 3D location where the point is displaced. To obtain 3D points belonging to the same instance displaced to the same 3D location in space, and points belonging to different instances displaced to different locations, we adapt the Lovász Hinge loss [144] to 3D as:

$$\mathcal{L}_{\text{tree}} = \frac{1}{|\mathcal{C}|} \sum_{j=1}^{|\mathcal{C}|} \text{Lovasz}(\boldsymbol{F}_{\mathcal{C}_j}, \boldsymbol{G}_{\mathcal{C}_j}), \tag{5.2}$$

where $\mathcal{C}$ is the set of object instances, $\boldsymbol{G}_{\mathcal{C}_j} \in \{0, 1\}^{|\mathcal{S}|}$ denotes the binary ground truth mask of the j-th instance, and $\boldsymbol{F}_{\mathcal{C}_j}$ is a soft-mask obtained by the offset prediction. The soft-mask $\boldsymbol{F}_{\mathcal{C}_j}$ for instance $\mathcal{C}_j$ is obtained from the offset prediction: each point in the point cloud gets a score that depends on how far from the instance centroid $\boldsymbol{c}_j$ its offset points to. The score is formalized as

$$f_{\mathcal{C}_j} = \exp\left(- \frac{||\boldsymbol{e}_p - \boldsymbol{c}_j||^2}{2\eta^2} \right), \tag{5.3}$$

where $\boldsymbol{e}_p$ indicates the 3D location pointed by the predicted offset at point $\boldsymbol{p}$, and η is a hyperparameter that defines an isotropic clustering region around the centroid.

Finally, the third decoder targets standard instance segmentation, where we aim to individually segment each trunk and fruit in the scene. Similarly to the previous decoder, here also we have a single output head for 3D offset prediction, and optimize with the Lovász Hinge loss $\mathcal{L}_{\text{ins}}$, as in Equation (5.2).

The final loss function $\mathcal{L}$ is given by a weighted sum of the individual losses:

$$\mathcal{L} = w_1 \mathcal{L}_{\text{sem}} + w_2 \mathcal{L}_{\text{tree}} + w_3 \mathcal{L}_{\text{ins}}. \tag{5.4}$$

5.2.2 Skip Connections

Skip connections are crucial to ensure feature reusability and address the gradient degradation problem of CNNs [168]. They skip one or more layers, providing a direct gradient flow from the late stages of the decoder to the early stages of the encoder, which is usually hindered by the downsampling operation of the encoder. We aim to extend our previous work on hierarchical panoptic segmentation on RGB images [165] to the 3D point cloud domain, and thus we propose to adopt a hierarchical skip connection scheme to exploit the underlying hierarchy of segmentation tasks. Similarly to Chapter 4, we propose to connect different decoders directly, as shown in Figure 5.2, instead of encoder and decoders only as in the widespread standard UNet-inspired model [168]. A visual breakdown of the scheme we use is shown in the previous chapter, in Figure 4.3. In particular:

1. For semantic segmentation, we keep the skip connection from the encoder to the decoder. The spatial information contained in the high-resolution feature maps of the encoder helps the decoder for segmentation.

2. For tree instance segmentation, we fuse the high-resolution maps coming from the encoder with the feature coming from the same level of the semantic decoder. In this way, the tree instance segmentation decoder will process an enriched feature map, containing also information about semantic segmentation.

3. For instance segmentation, we fuse the high-resolution maps coming from the encoder with the feature coming from the same level of the tree instance segmentation decoder. Notice that, in this way, the skip features include semantic segmentation features as well, which come from the tree instance segmentation branch.

To provide a more intuitive idea of the contribution brought by such skip connection scheme, we reconsider the example we discussed in the previous chapter, which treated the hierarchical panoptic segmentation task as a process that

assembles segmentation in layers. The encoder provides a basic high-resolution spatial guide, and the semantic decoder adds labels to it. The tree instance segmentation decoder leverages this enriched representation, identifying individual trees as landmarks (at this point, in the previous chapter, we obtained individual crops as landmarks). Finally, the instance segmentation decoder refines this by adding finer distinctions using all previously gathered information. In this case, these finer details consist of individual fruits and trunks, in contrast to the crop leaves of the previous chapter. At each level, skip connections integrate detailed and contextual features, leading to a more comprehensive and enriched representation.

This skip connection scheme allows us to exploit the hierarchy among tasks, enriching the features propagated to each decoder. In Section 5.4.3, we provide extensive experiments on different skip connection schemes to show that this design choice yields superior performance.

5.2.3 Post-Processing

As mentioned, the instance segmentation decoders predict an offset vector for each point. The goal is to have offsets of points belonging to the same instance indicating the same 3D location in space, and offsets of points belonging to different instances indicating different 3D locations. To obtain the final instance masks, we cluster the offsets with HDBSCAN [131]. Additionally, we use the semantic prediction to enforce consistency among instances in the third decoder, for example, to avoid the case in which two points with two different semantic predictions end up in the same instance. This cannot be applied to the second decoder, as there instances are composed of one trunk and all apples attached to it, so a single instance actually includes multiple semantic classes.

5.3 Our Dataset for Hierarchical Panoptic Segmentation

Our dataset, called HOPS, is composed of point clouds collected with three different sensors at Campus Klein-Altendorf near Bonn, Germany. First, we collected point clouds using a TLS placed at multiple locations in the orchard row. Our training and validation set entirely contains point clouds collected with this sensor. Furthermore, we use a few TLS point clouds in the test set, collected in a different year and orchard location compared to the training and validation set. Second, we use point clouds obtained with a bundle adjustment procedure [191] using as input images collected with a UAV equipped with a PhaseOne iXM-100 camera. We flew three missions on the same orchard with a camera angle of 45°,

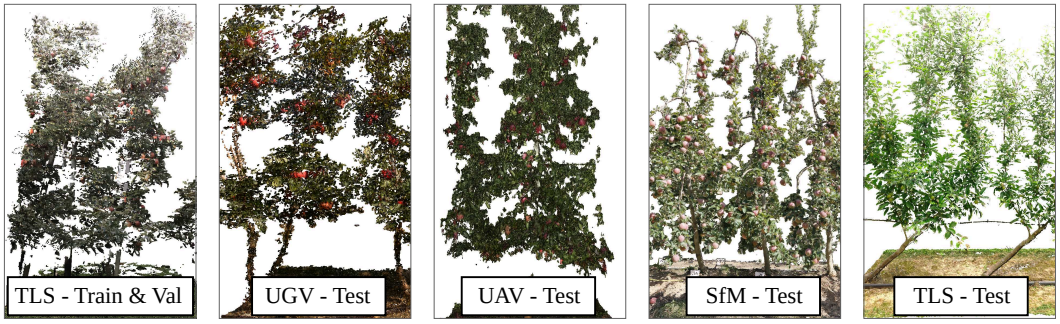


Figure 5.3: An example of the colored point clouds of HOPS. The point clouds from the train and validation sets are obtained using a TLS. We have four test sets recorded with different sensors. The one called "SfM" contains point clouds from the Fuji-SfM dataset [56, 57].

90°, and 135° from the ground plane at a height of approximately 20 m. This setup allows us to obtain good coverage of the trees, including the lower part of the canopy. The photogrammetric point clouds obtained in this way from the UAV are only included in the test set. Third, we use a RealSense D435i mounted on a mobile robot driving through the orchard rows to collect another part of our test set. Again, to obtain photogrammetric point clouds, we use a bundle adjustment and 3D reconstruction pipeline. Furthermore, we label a few point clouds from the Fuji-SfM dataset [56,57] to obtain a fourth test set from a different setup. This dataset is collected in Agramunt in Catalunya, Spain, with a Canon EOS 60D DSLR camera, followed by a bundle adjustment procedure to generate point clouds. In contrast to existing datasets in the agricultural domain [68, 157, 206], we specifically designed HOPS to have a remarkable domain shift between train and test sets, to more closely resemble the challenges encountered in real-world deployments. This intentional separation encourages the development and evaluation of models under realistic conditions, where distributional discrepancies are common. For this reason, we only included high-quality point clouds recorded with the TLS in the training set, while the test set includes a variety of sensors.

We report statistics about the different splits of our dataset in Table 5.1 and we show exemplary point clouds in Figure 5.3 to visualize the difference between training and test point clouds. To summarize, our dataset consists of a training set and a validation set obtained with a TLS and four different test set obtained with different sensors (TLS, PhaseOne iXM-100, RealSense d435i, and EOS 60D) in different locations.

For our dataset, we define 5 semantic classes, namely *ground*, *trunk*, *canopy*, *apple*, and *pole*. To formalize the panoptic segmentation task, we define the set of "things" classes, namely, trunk and apple, and "stuff" classes namely, ground, canopy, and pole. Additionally, we define the *tree* thing class, which puts together

Table 5.1: Dataset statistics. We report the number of samples per split, the average number of points, fruits, and trunks per sample, and the average number of fruits per tree.

	Train	**Val**	**Test**			
	TLS	TLS	UGV	UAV	SfM	TLS
Number of samples	90	18	12	22	6	27
Avg. points per sample	0.8M	1M	0.6M	1.5M	1.8M	1M
Avg. fruits per sample	90.2	99.2	71.8	99.1	232.3	39.9
Avg. trunks per sample	3.1	3.2	2.9	1.9	2.7	2.9
Avg. fruits per tree	19.2	19.3	17.8	31.7	62.1	9.8

apples and trunk belonging to the same tree.

To label the data, we split the aggregated point cloud of each orchard row into individual tiles with an average of 1 million points each. We manually label each tile in three stages using the online tool provided by segments.ai. One annotator labels each tile for semantic and two-level instance segmentation, ensuring consistency between instance levels. A second annotator then verifies label quality. Unlike image annotation, point cloud labeling requires frequent viewpoint changes, making it more time-consuming. The average effort consists of 4 h per tile, totaling 525 h for labeling and 175 h for verification.

5.4 Experimental Evaluation

We present our experiments to show the capabilities of our method for hierarchical panoptic segmentation on real-world 3D point clouds. The results of our experiments also show that: (i) our approach can jointly perform semantic, tree instance, and standard instance segmentation on real-world 3D data acquired with different sensors; and (ii) our skip connection scheme allows the CNN to exploit the underlying hierarchical relationship between the individual segmentation tasks, leading to better segmentation performance.

5.4.1 Experimental Setup

Metrics. Following the explanation of Section 2.3, for semantic segmentation, we compute the intersection over union [51] on all five classes of our dataset, which we discussed in Section 5.3, and report the mean IoU in the following tables. For the tree instance segmentation, we compute the single-class panoptic quality over the "tree" class, which we report as PQ_T. For the standard instance

Table 5.2: Semantic segmentation results on the four different test sets of HOPS, and the average score across all sets and all tasks. Best results in bold. All metrics are reported as percentages (%).

Approach	mIoU				Average
	TLS	UAV	UGV	SfM	
MaskPLS [125]	23.3	20.0	18.7	14.7	23.3
ForestPS [124]	52.0	64.3	40.2	47.0	50.9
Ours	**57.7**	**65.2**	**42.6**	**47.7**	**53.3**

Table 5.3: Panoptic segmentation results on the four different test sets of HOPS, and the average score across all sets and all tasks. Best results in bold. All metrics are reported as percentages (%).

Approach	PQ				PQ_T				mPQ
	TLS	UAV	UGV	SfM	TLS	UAV	UGV	SfM	
MaskPLS [125]	36.0	36.0	35.1	41.4	-	-	-	-	42.3
	-	-	-	-	48.9	48.4	**47.1**	**45.3**	
ForestPS [124]	49.4	**62.8**	**37.7**	**46.2**	-	-	-	-	27.5
	-	-	-	-	21.5	2.1	0	0	
Ours	**49.5**	51.9	35.2	34.5	**55.4**	**48.9**	42.1	39.1	**44.6**

segmentation, we compute the panoptic quality over all classes and report the mean panoptic quality as PQ in the tables. We also report the overall mean panoptic quality (mPQ) as the average between PQ and PQ_T.

Training details and parameters. In all experiments, we use AdamW [115] with weight decay of 0.99 with an initial learning rate of $5 \cdot 10^{-3}$. We set the weights of the loss function to $w_1 = w_2 = w_3 = 1$. We train our approach for 500 epochs. We use batch size equal to 1. We use voxel downsampling to 3 mm. Due to practical constraints, the training set is limited to TLS-acquired data. However, to mitigate potential overfitting to TLS-specific characteristics and improve generalization to non-TLS domains, we apply extensive data augmentation techniques including scale, rotation around all axes, shear, color jittering, and elastic deformation [25]. We tuned all hyperparameters on the validation set.

All baselines can only address one instance segmentation task at a time and, thus, need to be trained twice to solve both instance-level tasks. Our method addresses both instance segmentation tasks with a single training run.

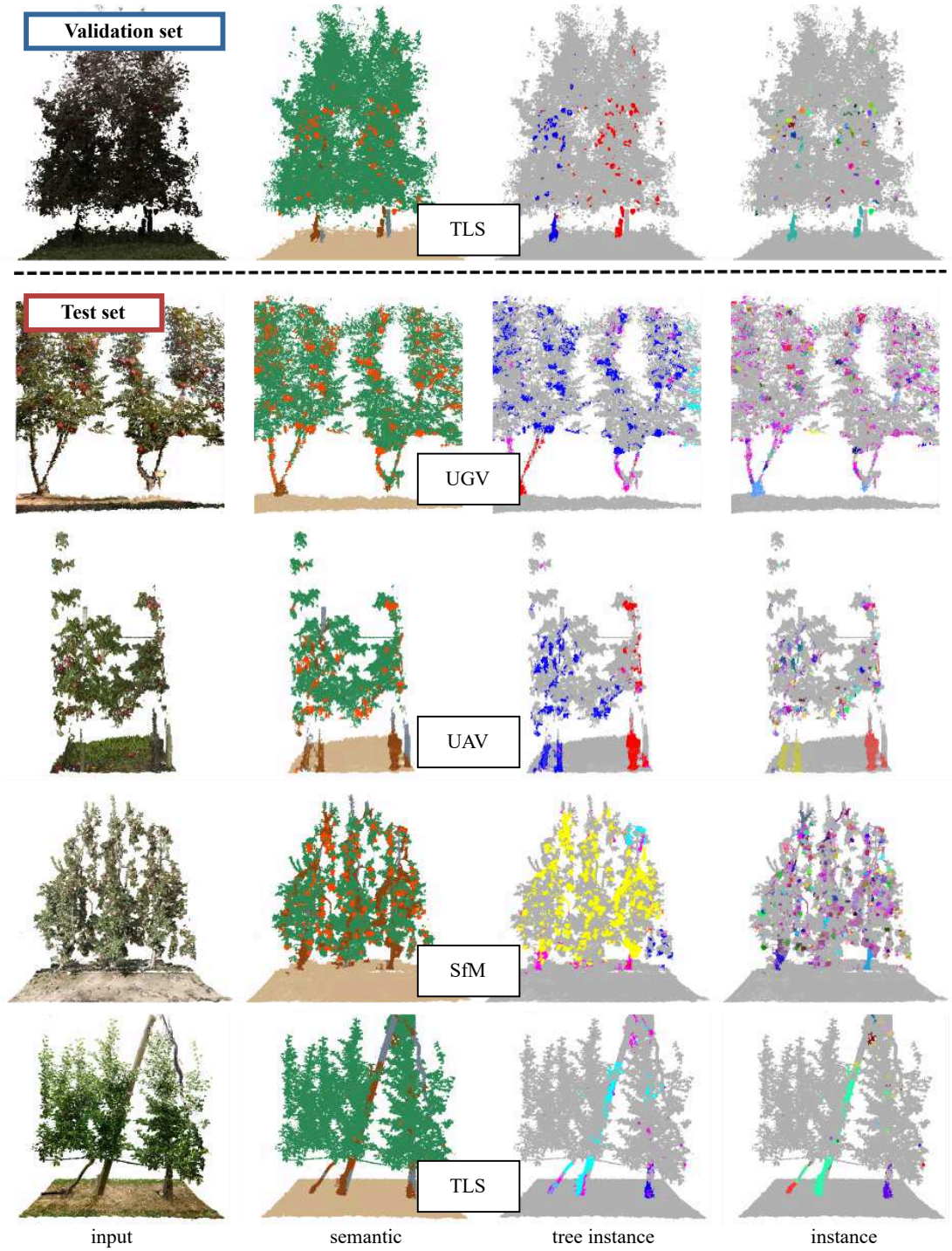


Figure 5.4: Qualitative results of our approach for hierarchical panoptic segmentation. The first row shows validation set results, the other rows show test set result. We report the input point cloud with all three segmentation predictions, and one result for each sensor. In the semantic prediction, different colors indicate different categories. In the instance predictions, different colors indicate different instances.

Table 5.4: Results on the validation set. We also report the number of parameters of each network, and the average iteration time on a GPU NVIDIA RTX A5000. Best results in bold. All metrics are reported as percentages (%), except Params and runtime.

Approach	mIoU	PQ	PQ_T	mPQ	Param	It. Time [s]
MaskPLS [125]	28.2 -	42.8 -	- 48.9	45.6	1.8M	4.0
ForestPS [124]	58.9 -	**62.2** -	- 2.3	32.3	1.5M	1.8
Ours	**70.9**	60.8	**52.1**	**56.5**	19.4M	2.0

5.4.2 Hierarchical Panoptic Segmentation

The first experiment evaluates the performance of our approach, and its outcomes demonstrate that our approach can jointly perform semantic, tree instance, and standard instance segmentation on real-world 3D data. For this experiment, we compare our approach to existing baselines for panoptic segmentation: MaskPLS [125] is a transformer-based approach that extends Mask2Former [31] to 3D point clouds, while ForestPS [124] is a convolutional neural network approach based on MinkUNet [35], originally designed for tree segmentation in a forest environment. The baselines are trained for only one of the two instance segmentation tasks at a time, as they do not support hierarchical multi-level instance segmentation. The test set results are presented in two separate tables: Table 5.2 reports the performance on 3D semantic segmentation, while Table 5.3 summarizes the results for 3D hierarchical panoptic segmentation. Although both sets of results are obtained from the same experimental run, they are reported separately for clarity and to accommodate space constraints. Furthermore, we report results on the validation set in Table 5.4. Our approach outperforms the baselines on both splits in terms of mIoU and on most PQ_T, while ForestPS achieves the best PQ on most splits. However, our approach performs better than the baselines on mPQ. While our PQ is inferior to ForestPS, our approach also yields good performance on tree instance segmentation, on which the dedicated training of ForestPS fails completely, achieving results below 3% on four out of five splits.

We show qualitative results in Figure 5.4. As explained in Section 5.3, only the validation set is recorded in the same time period and with the same sensor as the training set. The test sets, in contrast, are recorded either 2 years later, also using a TLS, or with different sensors (UGV and UAV), or belong to entirely different datasets (SfM). This motivates the performance gap between the validation and

Table 5.5: Ablation study on the validation set. Comparison between different skip connections schemes. Best results in bold. All metrics are reported as percentages (%).

	Skip Connections	mIoU	PQ	PQ_T	mPQ
A	None	55.8	42.8	47.7	45.3
B	Decoder	64.4	53.4	51.3	52.4
C	Encoder	69.8	58.8	50.9	54.9
D	Encoder+Decoder	**70.9**	**60.8**	**52.1**	**56.5**

the test set. We believe that this aspect pushes researchers to build models that yield good generalization performance and can be used in different conditions. In Table 5.2 and Table 5.3, we also report average scores for mIoU and PQ across all four splits and both instance segmentation tasks, to show how different approaches deal with both tasks at the same time.

5.4.3 Ablation Studies

In the ablation study, we aim to validate that the proposed skip connection scheme allows the CNN to exploit the underlying hierarchical relationship between the segmentation tasks, leading to better segmentation performance. We perform this experiment on the validation set. We compared to other skip connection schemes, and report results in Table 5.5. The simplest approach does not use skip connections, denoted as A in Table 5.5, which leads to poor performance. This is expected, since the decoders do not get any high-level information from features coming from earlier stages, which harms gradient flow. The second approach uses skip connections only from decoder to decoder, excluding the contributions from the encoder, denoted as B in Table 5.5. This approach, despite exploiting the hierarchy among tasks with decoder-based skip connections, does not yield good performance. The third approach is the standard UNet-like skip connections from the encoder to all decoders, denoted as C in Table 5.5. In this case, all decoders obtain the same information from the encoder. This proves more effective than the decoder-only skip connections. This is probably due to the fact that restoring high-level information is important, and using decoder-only skip connections does not help gradient propagation. Thus, our method using skip connections from the encoder and from decoder to decoder, denoted as D in Table 5.5, performs better than all others, as on one hand it benefits from the high-level features coming from the encoder, and on the other hand it exploits the underlying hierarchy among segmentation tasks. Interestingly, it also provides better results on semantic segmentation, despite there is no difference in the skip connection scheme for the semantic decoder, as the hierarchy affects only the instance decoders. This

suggests that having information coming from the instance decoders indirectly affecting the encoder and the semantic decoder via backpropagation is useful for semantic segmentation performance.

5.5 Conclusion

In this chapter, we presented a novel approach for hierarchical 3D panoptic segmentation on point cloud data. Our approach enables comprehensive and structured 3D scene understanding by jointly reasoning about multiple segmentation tasks in point clouds. By means of a novel skip connections scheme, our method exploits the underlying hierarchy among different segmentation tasks, and is able to yield semantic segmentation, tree instance segmentation, where a tree is defined as a trunk and all the apples belonging to it, and standard instance segmentation at the same time. Thanks to the proposed skip connection scheme in our architecture, our approach achieves state-of-the-art results, surpassing or being comparable to existing task-specific baselines, despite that they can deal with only one instance segmentation task at a time. Additionally, we introduce a novel point cloud dataset of real apple orchards, called HOPS, labeled for hierarchical panoptic segmentation. Our dataset includes data recorded with different sensors over the course of two years. Our dataset is specifically designed to exhibit a pronounced domain shift between train and test sets, in order to better reflect the challenges encountered in real-world deployments. We believe that this intentional separation will encourage researchers to actively address the generalization limitations of CNNs, which are known to overfit to sensors, modalities, and environmental conditions. Our dataset aims to foster progress toward more resilient and broadly applicable perception systems.

This is the last chapter of the first part of this thesis, and it completes our discussions on how to go beyond the standard formulation of panoptic segmentation. In the next part, we face a major challenge. We aim to move beyond the closed-world assumption discussed in Section 2.2, which has been implicitly adopted by all the methods discussed so far, and move towards open-world panoptic segmentation.

Part II

Open-World Panoptic Segmentation

Chapter 6

Introduction to Open-World Segmentation

Scene interpretation is a fundamental capability for autonomous vision systems operating in real-world scenarios. Over the past years, we have witnessed remarkable progress in this area, with significant advances in semantic segmentation [114, 136], instance segmentation [71, 127], and panoptic segmentation [87, 165, 187]. In Part I of this thesis, we presented several approaches for panoptic segmentation, demonstrating their effectiveness in structured environments and benchmarking them on well-established datasets. These methods are able to deliver accurate results, but they share a critical limitation: they all operate under the so-called *closed-world* assumption.

As discussed in Chapter 2, the closed-world assumption means that the set of categories encountered during inference is identical to the set of categories available during training. In other words, the world is assumed to be perfectly described by the training data, and every object type that might appear at test time has already been labeled and seen before. This assumption is convenient from a machine learning perspective, as it simplifies both model design and evaluation, as models can be optimized to discriminate only among a fixed, predefined set of classes. It is also consistent with the way most large-scale vision benchmarks are constructed: datasets such as Cityscapes [36] or COCO [106] define a finite set of categories and evaluate models exclusively on their ability to recognize those. However, this assumption is fundamentally unrealistic in unconstrained real-world scenarios. The visual world is inherently open-ended, and new objects, artifacts, or categories emerge continuously; it is unrealistic to aim to represent them all in the training set.

This mismatch between the closed-world assumption and the open-ended nature of reality poses a major limitation for deploying current segmentation algorithms in safety-critical applications. A central challenge for learning-based

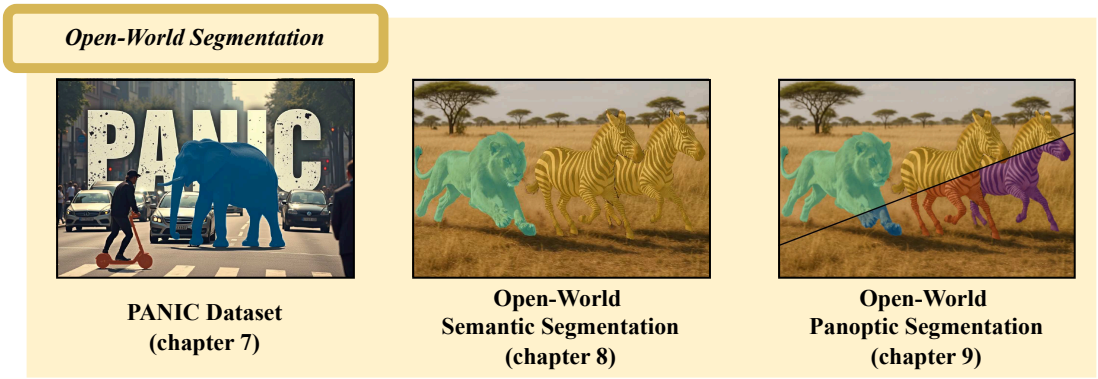


Figure 6.1: Overview of the contributions to open-world panoptic segmentation, namely a dataset for all open-world segmentation tasks, an approach for open-world semantic segmentation, and an approach for open-world panoptic segmentation.

perception systems is therefore the ability to generalize beyond the training distribution and robustly handle novel categories at test time. Consider, for instance, an autonomous vehicle navigating a busy urban environment: while it may be trained on large-scale datasets containing cars, bicycles, pedestrians, etc., it will inevitably encounter unfamiliar objects such as new types of personal mobility devices, construction machinery, delivery robots, or even unusual obstacles like furniture or debris left on the road. From the perspective of a closed-world model, these objects do not belong to any known class and are thus generally misclassified, potentially leading to catastrophic decisions downstream.

Failing to recognize and properly account for the unknown can severely compromise both the safety and robustness of autonomous systems. For example, if a delivery robot on the roadside is mistaken for a pedestrian, the vehicle may behave overly cautiously, leading to inefficient navigation. Conversely, if an animal or a different kind of vehicle, such as an electric scooter, is ignored because it does not match any known class, the vehicle might collide with it, causing damage or harm. These scenarios illustrate why it is not sufficient for autonomous systems to merely excel on closed-world benchmarks, but they must also be able to reason about what lies beyond the scope of their training data.

For this reason, autonomous vision systems must move beyond the closed-world paradigm and embrace the *open-world* setting. In this setting, it is explicitly acknowledged that the training data cannot exhaustively cover all possible scenarios. The model must therefore be able to detect, segment, and reason about novel objects and categories at test time, without relying on them being predefined in the training set. Open-world perception thus shifts the focus from recognizing only a fixed taxonomy of classes to handling an evolving and potentially unbounded set of visual concepts. This capability is not only intellectually appealing from a research standpoint but also essential for the safe and reliable

deployment of autonomous systems in complex, ever-changing environments.

Despite its importance, open-world segmentation remains relatively underexplored, with only a few benchmarks [17, 27, 143] and methods [55, 141, 161] introduced in recent years. In parallel, the advent of large language models (LLMs) and vision-language models (VLMs) has shifted much of the community's attention towards open-vocabulary segmentation [58], which leverages textual embeddings to extend the set of recognizable categories. While related in spirit, open-vocabulary segmentation is inherently different: it assumes that novel categories can be described through language, whereas the open-world problem addresses the more fundamental challenge of discovering completely unseen objects without any prior semantic description.

In this part of the thesis, we address this challenge by developing methods for open-world segmentation, aiming to equip autonomous systems with the ability to not only interpret known categories but also reliably discover and handle the unknown. We show a schematic of our contributions to open-world segmentation in Figure 6.1, which reports only the open-world part of Figure 1.2.

6.1 Problem Definitions

Open-world segmentation remains relatively under-explored, gaining traction only recently with the release of dedicated benchmarks [17, 27, 143]. As a result, task nomenclature is often ambiguous or inconsistent. Therefore, we define a clear and unified terminology for all open-world segmentation tasks, and use it throughout the thesis. In Figure 6.2, we show a visual example of the open-world segmentation tasks on an exemplary image that has been manually labeled for illustration purposes.

6.1.1 Anomaly Segmentation

SegmentMeIfYouCan [27] formalized the task of anomaly segmentation, first introduced by Zhang *et al.* [226]. In this setting, the goal is to identify image regions that do not belong to any of the known categories provided during training, which is the so-called anomaly. There is no concept of unknown category or object. Instead, all unseen content is grouped together into a single anomaly class, regardless of whether it corresponds to a delivery robot, an animal, construction equipment, or anything else. The task can therefore be understood as a binary segmentation problem, where the input image is partitioned into two complementary regions: (i) pixels belonging to known, training-time categories, and (ii) pixels belonging to everything else, collectively labeled as anomaly.

Given an image $I \in \mathbb{R}^{H \times W}$, the prediction for anomaly segmentation is a

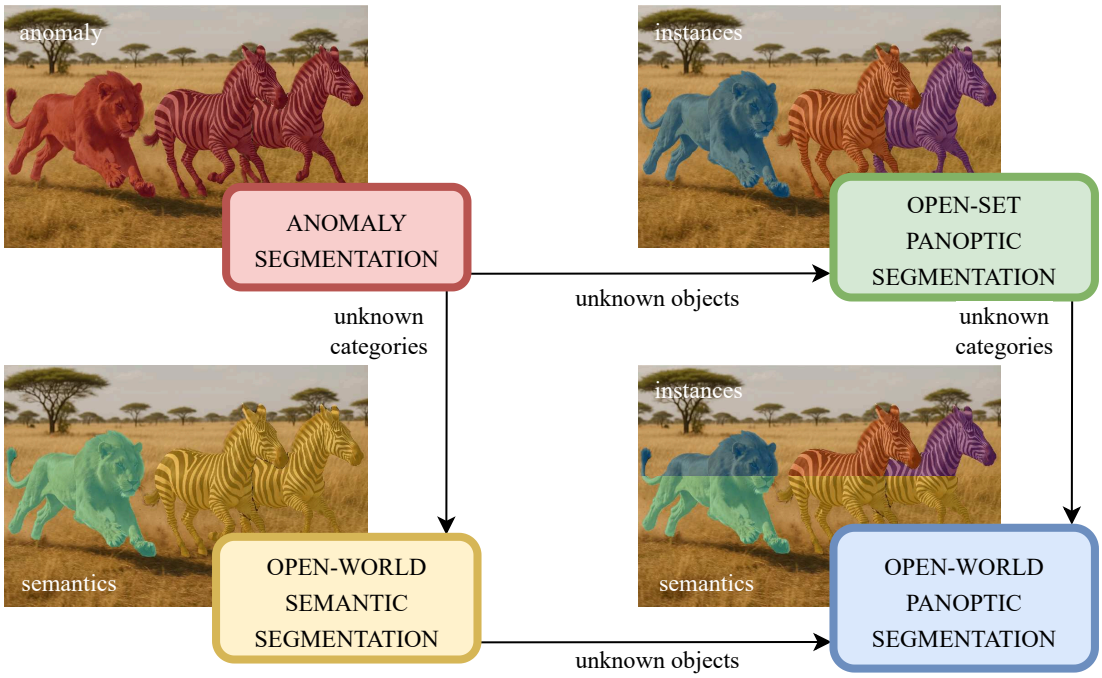


Figure 6.2: A visual breakdown of the four open-world segmentation tasks. Anomaly segmentation segments all anomalous areas as unknown (zebras and lion together). Open-world semantic segmentation separates classes (zebras segmented together, lion separate). Open-set panoptic segmentation segment separate objects but has no category information. Open-world panoptic segmentation has both, classes and objects information. Exemplary RGB image is generated with perplexity.ai [154].

mask $M_a \in \{0,1\}^{H \times W}$, where 0 indicates known and 1 indicates unknown. For example, in Figure 6.2, the zebras and the lion are segmented together in the same anomalous region.

6.1.2 Open-World Semantic Segmentation

Open-world semantic segmentation [186] extends anomaly segmentation by not only detecting anomalous regions but also assigning them to distinct semantic categories. In contrast to anomaly segmentation, where all unknown content is grouped together into a single anomaly label, open-world semantic segmentation requires the model to differentiate between multiple novel categories that were never seen during training. For example, an autonomous vehicle operating in an urban setting may encounter both a street vendor cart and an electric scooter. While they would be labeled together in anomaly segmentation, open-world semantic segmentation enforces that these objects belong to different semantic categories and assigns them consistent labels across the dataset. The output of this task is still a dense, per-pixel semantic labeling, just like in standard closed-world semantic segmentation [36, 114]. Known classes (e.g., cars, pedestrians, road)

retain their familiar labels, while novel categories discovered at test time are segmented and grouped into new, previously unknown classes. Importantly, these labels remain consistent: if a new category, such as "electric scooter", is discovered, all pixels belonging to scooters in subsequent images must be assigned the same semantic label. This consistency allows for the gradual incremental expansion of the semantic space, bridging the gap between closed-world and open-world perception.

It is worth noting that, as in standard semantic segmentation, there is no notion of individual object instances in this task. The goal is purely to assign a semantic class to each pixel, meaning that two scooters appearing in the same image will be labeled as belonging to the same category, without distinguishing them as separate entities. Given an image $I \in \mathbb{R}^{H \times W}$, the prediction for open-world semantic segmentation is a mask $M_s \in \{0, \ldots, K\}^{H \times W}$, with 0 for known and $1, \ldots, K$ for the K new categories. For example, in Figure 6.2, the zebras and the lion are segmented into two separate categories, and the two zebras share the same semantic label.

6.1.3 Open-Set Panoptic Segmentation

Open-set panoptic segmentation has recently received increasing attention in the literature [79,161] and can be seen as a natural extension of anomaly segmentation toward instance-level reasoning. Instead of merely grouping all unknown pixels into a single anomaly mask, this task requires the model to separate and segment individual object instances within the anomalous regions. The focus here is not on discovering or defining new semantic categories, but rather on identifying distinct, previously unseen objects and delineating their boundaries. Conceptually, this task could also be described as *open-world instance segmentation*. However, since the term open-set panoptic segmentation has become widely adopted in the literature, we follow this convention in this thesis.

In practice, open-set panoptic segmentation can be seen as performing class-agnostic instance segmentation inside the anomaly mask. For example, consider an autonomous vehicle encountering a construction site with previously unseen equipment, such as a forklift and a road barrier. While anomaly segmentation would simply mark the entire region as "unknown", open-set panoptic segmentation ensures that the forklift and the barrier are two separate instances, even though the system cannot assign them meaningful semantic labels.

This formulation is valuable because it adds structural detail to the anomaly space: it enables the perception system to reason about the number, shape, and extent of previously unseen objects, which is crucial for downstream tasks such as planning and collision avoidance.

Given an image $I \in \mathbb{R}^{H \times W}$, the prediction for open-set panoptic segmentation

is a mask $M_a \in \{0,1\}^{H \times W}$, where 0 indicates known and 1 indicates unknown, and a mask $M_i \in \{0, \ldots, N\}^{H \times W}$ where 0 indicates both known and no object areas, and $1, \ldots, N$ indicate the N object instances discovered. For example, in Figure 6.2, the zebras and the lion are segmented as three separate object instances, and there is no information about the two zebras belonging to the same category.

6.1.4 Open-World Panoptic Segmentation

Open-world panoptic segmentation represents the most comprehensive formulation among open-world perception tasks. It extends open-world semantic segmentation by not only discovering novel semantic categories and assigning them consistent labels across different images, but also segmenting individual object instances within those categories. At the same time, it advances beyond open-set panoptic segmentation, which remains limited to class-agnostic instance discovery within the anomalous area and does not attempt to establish new semantic categories. In open-world panoptic segmentation, the objective is to achieve a truly holistic scene understanding: every pixel in the image must be assigned to a semantic category and, whenever applicable, provided with an instance ID, regardless of whether the object belongs to a class known from training or to a completely novel, previously unseen class.

Given an image $I \in \mathbb{R}^{H \times W}$, the prediction for open-world panoptic segmentation is a mask $M_s \in \{0, \ldots, K\}^{H \times W}$, where 0 indicates known and $1, \ldots, K$ indicate the K categories discovered, and a mask $M_i \in \{0, \ldots, N\}^{H \times W}$, where 0 indicates both known and no object areas, and $1, \ldots, N$ indicate the N object instances discovered. For example, in Figure 6.2, the zebras and the lion have both, the semantic information separating them into two categories and the object instance IDs distinguishing the three individual objects.

6.2 Related Work

The first approaches that aimed to relax the closed-world assumption in order to identify previously unseen samples targeted the problem of anomaly detection and classification. Mostly, their goal was to identify and discard anomalous samples. Early approaches rely on modification of standard closed-world segmentation techniques, such as using an extra class for identifying anomalies [18,138,196], thresholding or modifying the softmax activation [9, 23, 75], and using model ensembles [94, 199]. These approaches paved the way for modern open-world segmentation techniques, despite having issues in reliably identifying anomalies, due to the fact that usually the predictions were overconfident and showed a

peak in the softmax activations also for unknown samples [38, 145]. Later on, researchers explored novel ideas for segmenting unknowns, such as maximizing entropy [46], predicting energy scores [110], and estimating the model uncertainty via Bayesian deep learning [52, 94, 173] or the network gradients [109, 121]. Many recent approaches [79, 208] instead rely on having some unknown objects, usually called out-of-distribution samples, in the training set, to be able to distinguish the known objects from all others. Generative models [91, 230] have also proven useful for this task, since in the reconstruction phase, unknown areas will have a lower reconstruction quality than the known, and can thus be recognized by looking at the most dissimilar areas between the input and the output. Due to the limitation of available training data, many unsupervised approaches use synthetic anomaly data and train an anomaly detector that is either distance-based [108, 170, 192] or reconstruction-based [11, 222, 227], with the latter sharing the same concept as the generative models mentioned above. Recently, open-world segmentation is gaining traction, and promising approaches have come to light. Most novel approaches predict the anomalous area and cluster individual instances in a class-agnostic fashion [55, 141, 142, 161]. Incrementally discovering multiple unknown semantic classes at test time is, instead, a relatively unexplored research direction [186].

A closely related line of research is open-vocabulary segmentation [58, 103, 159, 217, 224], sometimes also referred to as zero-shot segmentation. Similar to open-world segmentation, these methods aim to recognize and segment categories that were not part of the training set. However, rather than discovering unknown categories in a purely data-driven fashion, open-vocabulary approaches typically leverage the rich semantic knowledge embedded in large-scale vision-language models, such as CLIP [160]. By aligning visual features with text embeddings, these models enable the use of text prompts to guide the identification and segmentation of novel classes, even if no explicit visual examples are available during training. Despite these advances, it is important to emphasize that open-vocabulary and open-world segmentation are fundamentally different paradigms. Open-vocabulary segmentation assumes that novel categories can be described through language, effectively constraining the unknown to a predefined semantic space accessible via text. In contrast, open-world segmentation deals with the more challenging and safety-critical scenario where novel objects emerge without any prior semantic description, requiring models to first discover, and only later potentially recognize, such categories. To illustrate, consider an autonomous vehicle navigating in a city. With open-vocabulary segmentation, one must anticipate the set of possible categories by providing prompts such as car, pedestrian, or bicycle. If an unexpected category, like lion or zebra, is not prompted, the model will simply fail to detect it. Open-world segmentation, in contrast, makes

no such assumptions: the vehicle would segment these novel objects as unknown categories regardless of whether they were anticipated, thereby offering a much more reliable safety guarantee in unpredictable real-world environments.

While large-scale datasets exist for closed-world segmentation in different domains [36, 106, 206], datasets for open-world segmentation are comparably rare. The WildDash dataset [223] provides anomalous images where full-image anomalies are present. Another dataset is MVTec [10], which mostly targets the industrial scenario. BrainMRI & HeadCT [171] are two datasets targeting the medical domain for detecting lesions on different organs. ALLO [97] is a synthetic photorealistic dataset with full-image anomalies for robot operations in lunar orbit. Among the most common datasets for open-world segmentation in the autonomous driving domain are the Fishyscapes-LostAndFound benchmark [17, 155], which is based on the same setup as Cityscapes and presents anomalous objects in the middle of the street, the CAOS benchmark [73] that originates from the BDD100K [221] dataset to create an open-world test set, and TAO-OW [111] for open-world instance tracking. These datasets are, however, characterized by a low diversity of anomalies. Chan *et al.* [27] introduced the SegmentMeIfYouCan benchmark, which is a test set that relies on the known classes from Cityscapes and introduces anomalies of various kinds and sizes. However, it provides a limited number of images, and no semantic or instance annotation, but only a binary segmentation mask between known and unknown. Recently, Nekrasov *et al.* [143] extended SegmentMeIfYouCan by adding instance information, even though semantic information is still missing.

Chapter 7

PANIC: A Benchmark for Open-World Segmentation Tasks

Open-world segmentation is of paramount importance, as it removes the restrictive closed-world assumption and enables the development of algorithms capable of handling the open-ended complexity of the real world. Yet, as discussed in Chapter 6, both datasets and methods for open-world segmentation remain scarce when compared to their closed-world counterpart. The progress of deep learning research relies on annotated datasets that enable rigorous quantitative evaluation, and this requirement is not different for open-world segmentation. Even though the task aims to segment categories and objects that do not appear in the training set, an annotated test set that contains such categories and objects is crucial to ensure that the effectiveness of the developed method can be quantitatively and reliably demonstrated.

For this reason, we introduce PANIC, short for Panoptic Anomalies In Context, a dedicated test set for open-world segmentation in the autonomous driving domain. PANIC is designed to provide a realistic and diverse benchmark for evaluating algorithms beyond the closed-world assumption, ensuring that models are tested not only on familiar categories but also on objects and semantics that have never been observed during training. The dataset consists of 800 high-quality images, carefully curated and annotated, and encompasses 58 distinct unknown categories. Following the setup of SegmentMeIfYouCan [27], we adopt Cityscapes [36] as the standard training set for known classes, thereby ensuring consistency with existing autonomous driving research and facilitating comparability across different methods.

To structure the evaluation, the 58 unknown categories are further divided into validation and test subsets. The validation set contains categories that, although not explicitly labeled during training, still appear within the unlabeled regions of Cityscapes images (e.g., garbage bins, parkimeters, monuments, and

Table 7.1: Comparison of open-world segmentation datasets. In the categories, "N.A." means that there is no annotation for semantic classes.

Dataset	Images		Categories	Instances	Test Set
	Val	Test			
Lost-and-Found [17]	373	1203	N.A.	1864	✓
CAOS BDDAnomaly [73]	0	810	3	1231	✗
RoadObstacle21 [27]	0	327	N.A.	388	✓
SegmentMeIfYouCan [27]	10	100	N.A.	262	✓
PANIC (ours)	131	679	58	4029	✓

other urban elements). These categories are often referred to in the literature as *known unknowns* [9], since the model is implicitly exposed to them during training without receiving direct supervision. This setup allows researchers to tune hyperparameters and develop strategies for handling anomalies while avoiding data leakage into the final test set.

The test set, on the other hand, contains the much more challenging *unknown unknowns*: categories that are completely absent from the training data and from the open-world validation set, and therefore represent true novelty for the model. These include objects such as electric scooters, recumbent bicycles, forklifts, and other rare or emerging elements of urban traffic scenes. By incorporating such categories, the PANIC test set pushes models towards robust generalization and explicitly evaluates their ability to detect, segment, and reason about objects never encountered before. Thus, PANIC provides a comprehensive and systematic benchmark for open-world segmentation, bridging the gap between theoretical formulations of the task and the practical requirements of deploying autonomous systems in complex, ever-changing real-world environments.

We annotated PANIC to allow evaluation for all four open-world segmentation tasks introduced in Chapter 6, namely anomaly segmentation, open-world semantic segmentation, open-set panoptic segmentation, and open-world panoptic segmentation. We release image and ground truth annotations for the validation set, while for the test set, we release only the images.

A comparison with existing datasets is shown in Table 7.1. LostAndFound [17] contains 9 different types of anomalies, misclassifying common Cityscapes classes like children and bicycles as anomalies. The CAOS BDDAnomaly benchmark [73] suffers from a similar low-diversity issue, featuring just 3 unknown classes sampled from BDD100K [221]. Both, Fishyscapes and CAOS, try to mitigate this low diversity by introducing synthetic data in the dataset, but this lacks real-world realism. RoadObstacle21 and SegmentMeIfYouCan (also known as RoadAnomaly21

in the original paper) [27] offer more diverse anomalies, but only focus on anomaly segmentation. Recently, Nekrasov *et al.* [143] provided instance-level annotation for these datasets, but without semantic labels, falling under open-set panoptic segmentation, where instances are detected without category differentiation.

In PANIC, anomalies can appear anywhere in the image. They can have any size and are not limited to active traffic participants. We provide pixel-wise annotations of semantic classes and object instances.

7.1 Data Collection and Labeling

We recorded all images of PANIC in Bonn, Germany, using a car equipped with a sensor suite including common automotive setups, including multiple cameras and LiDARs [198]. For the purpose of our benchmark, we restricted the dataset to images captured by the front-facing Basler Ace acA2040-35gc camera, as this view is the most relevant for downstream perception tasks in autonomous driving, where forward scene understanding is critical.

The recordings were collected over two years, covering different times of day, seasons, and environmental conditions. This long-term acquisition strategy ensures that the dataset reflects a broad variability of urban driving scenarios, including changes in lighting, weather conditions, and seasonal characteristics. Such diversity is crucial for building robust evaluation benchmarks, as it prevents models from overfitting to overly-constrained visual conditions and exposes them to a wider spectrum of realistic challenges encountered in real-world driving.

After collection, we carefully sampled a subset of the raw recordings to construct a balanced dataset: duplicate frames and images without visible anomalies were removed in order to maximize both diversity and relevance. The selected 800 images were then annotated at pixel level using the online tool segments.ai.

To define what constitutes an anomaly, we followed the common practice of adopting the 19 Cityscapes evaluation classes [36] as the set of known categories. Any pixel belonging to these 19 classes is therefore labeled as “not anomaly”. Conversely, everything outside this set is annotated as anomaly and given both semantic- and instance-level annotations. This design choice ensures compatibility with existing closed-world benchmarks and provides a natural extension to the open-world setting, allowing researchers to directly compare how well algorithms trained on Cityscapes can generalize to unseen categories in PANIC.

Our dataset is designed for evaluation in realistic conditions, and we also accounted for privacy regulations. Specifically, we anonymized all faces, license plates, and windows of private buildings, masking them in black wherever necessary. Importantly, this does not introduce any bias into learning algorithms, as PANIC is not intended for training but rather for testing and evaluation, with

(a) RGB Image (b) Semantic Annotation (c) Instance Annotation

Figure 7.1: Our dataset, PANIC, provides pixel-wise annotations of unknown semantic categories and object instances of RGB images. In the semantic annotation, same color corresponds to same semantic category. The open-world mask is shown over the RGB image for both, semantic and instance annotation.

training carried out on Cityscapes, which already provides the known classes. In addition to anomaly and known-class annotations, we also define an "ignore''

label that encompasses pixels belonging to the ego-vehicle as well as anonymized regions. This ensures a clean separation between evaluation-relevant regions and those that should not influence results. Some sample images from the dataset, along with the semantic and instance annotations, are shown in Figure 7.1.

7.2 Benchmarks and Metrics

We propose four benchmarks with public Codabench competitions [218], one for each of the segmentation tasks described in Chapter 6, namely anomaly segmentation, open-world semantic segmentation, open-set panoptic segmentation, and open-world panoptic segmentation. Each has its own evaluation pipeline and metrics.

7.2.1 Anomaly Segmentation

Following SegmentMeIfYouCan [27], we use two groups of metrics for evaluating anomaly segmentation: pixel-level metrics and component-level metrics.

The first pixel-level metric is the area under the precision-recall curve (AUPR) that evaluates the separability of the pixel-wise anomaly scores between known and unknown, putting more emphasis on the minority class. This makes it generally good for anomaly segmentation, since often the anomalies are less prominent than the known classes. The other pixel-level metric is the false positive rate at 95% true positive rate (FPR95), which indicates how many false positive predictions must be made to reach the desired true positive rate (i.e., 95%). Here, the true positive rate specifically refers to correctly detecting pixels belonging to the unknown (anomalous) regions.

Pixel-level metrics, however, are dominated by large regions and fail to properly evaluate small anomalies. Component-level metrics are better at evaluating all anomalous regions in the scene, regardless of their size. SegmentMeIfYouCan [27] introduces three metrics for component-level evaluation: the segment-wise intersection over union (sIoU) for evaluating true positive and false negatives, the positive predicted value (PPV, or component-wise precision) for taking into account false positives, and the component-wise F1-score, which summarizes true positives, false positives, and false negatives.

We decided not to use the area under the ROC curve (AUROC), because recently several papers showed its limitations [66, 78, 203], as two models with the same performance may differ substantially in terms of how clearly they separate in-distribution and out-of-distribution samples. In general, these works argue that AUROC is not a fair metric for comparing different approaches.

The related Codabench competition is publicly available at `https://www.codabench.org/competitions/4561`.

7.2.2 Open-World Semantic Segmentation

The task of open-world semantic segmentation is not common in the literature; thus, a proper evaluation pipeline does not exist. Since standard semantic segmentation is comprehensively evaluated via the IoU [51], we aim to extend it to the open-world case.

The closed-world IoU is typically computed by building a confusion matrix $\boldsymbol{C} \in \mathbb{R}^{K \times K}$, with K the number of known classes, in which the i-th row indicates, for each class, how many pixels have been predicted as belonging to category i. In this case, predicted class i refers to ground truth class i. A perfect IoU of 1.0 is obtained when entry (ii) of the confusion matrix is equal to the number of pixels belonging to ground truth class i, and both the i-th row and i-th column have no entry different from 0 but the i-th one. In the following, we indicate as $\text{row}_i \in \mathbb{R}^K$ the i-th row of a confusion matrix. Similarly, $\text{column}_k \in \mathbb{R}^{\tilde{K}}$ is the k-th column of the confusion matrix. The closed-world IoU concept can easily be extended to the open-world case, with two major modifications:

1. The open-world confusion matrix should not be square, as it is possible that the evaluated approach discovers a different number of categories $\tilde{K}$ than the ones that are actually annotated. The confusion matrix will have dimension $\tilde{K} \times K$, where the number of rows $\tilde{K}$ refers to the newly-discovered classes, and the number of columns K is fixed and corresponds to the ground truth classes.

2. In the open-world IoU computation, the predicted class corresponding to row i of the confusion matrix does not have to be matched to ground truth class i, but rather to $\text{argmax}(\text{row}_i)$, where $\text{row}_i \in \mathbb{R}^K$ and $i \in \{1, \ldots, \tilde{K}\}$. Simply put, in open-world semantic segmentation, we do not expect a precise mapping in which predicted class i corresponds to ground truth class i, but we allow predicted class i to match to ground truth class j. This is possible because, as we do not explicitly optimize for categories, we have no control over the order in which we discover new classes.

Notice that, in this description of the open-world IoU, we did not include known categories for simplicity of notation. Known classes would contribute to the confusion matrix with a square block, as described above.

Besides the open-world IoU, we are also interested in understanding how reliable our predictions are. To this end, we report two clustering metrics that are particularly fitting for our case, namely homogeneity and completeness [169].

Homogeneity is a measure of the ratio of samples of a single class pertaining to a single cluster. The fewer different classes included in one cluster, the better. In our case, the clusters are the predicted categories, and homogeneity Hom_i measures how consistent a certain predicted label is, i.e., how closely it follows a certain ground truth class or is spread among multiple ones. It is computed as

$$\mathrm{Hom}_i = \frac{\max(\mathrm{row}_i)}{\sum \mathrm{row}_i}, \quad \forall\, i \in \{1, \ldots, \tilde{K}\}. \tag{7.1}$$

Conversely, completeness Com_i measures the ratio of pixels of a given class that are assigned to the same cluster. In our case, it measures how consistently a certain ground truth category is predicted, i.e., if it is regularly predicted as the same class or if it is spread among many predicted classes. It is computed as

$$\mathrm{Com}_k = \frac{\max(\mathrm{column}_k)}{\sum \mathrm{column}_k}, \quad \forall\, k \in \{1, \ldots, K\}. \tag{7.2}$$

The related Codabench competition is publicly available at `https://www.codabench.org/competitions/4563`.

7.2.3 Open-Set Panoptic Segmentation

Open-set panoptic segmentation is evaluated by means of the panoptic quality [88]. This evaluation procedure has been adopted by previous approaches [161], and we follow it here. In this case, there are no semantic classes but only two categories, known and unknown. The unknown area contains object instances, while the known area is treated as "stuff", so that the quality of the closed-world instance segmentation does not affect the final panoptic quality. We also report segmentation and recognition quality.

The related Codabench competition is publicly available at `https://www.codabench.org/competitions/4562`.

7.2.4 Open-World Panoptic Segmentation

Following standard panoptic segmentation, we split the unknown classes among *things* and *stuff*, and then we measure open-world mIoU, panoptic quality, recognition quality, segmentation quality, homogeneity, and completeness.

The related Codabench competition is publicly available at `https://www.codabench.org/competitions/4477`.

7.3 Conclusion

In this chapter, we introduced PANIC, a dataset specifically designed to benchmark open-world segmentation tasks in the autonomous driving domain. PANIC

includes a broad and diverse set of unknown semantic categories and object instances, enabling systematic evaluation across all four open-world segmentation tasks presented in Chapter 6. By covering anomaly segmentation, open-world semantic segmentation, open-set panoptic segmentation, and open-world panoptic segmentation, the dataset provides a comprehensive testing ground for measuring the robustness of perception systems under real-world conditions.

Beyond the dataset itself, we carefully defined evaluation protocols for each task, ensuring that results obtained with PANIC are rigorous, consistent, and reproducible. This standardization is crucial for enabling meaningful comparisons between different methods and for accelerating progress in the field. By providing a dedicated benchmark that captures the unpredictable and open-ended nature of urban environments, PANIC lays the foundation for the next generation of perception systems capable of handling the complexity of the real world.

In the following chapters, we present our methodological contributions to open-world segmentation. Specifically, in the next chapter, we address open-world semantic segmentation, where the challenge is to discover novel semantic categories at the pixel level, and evaluate our approach on multiple publicly available benchmarks, as well as on PANIC. In the following chapter, we push this further to open-world panoptic segmentation, where we not only discover new categories but also identify and segment their individual instances. Our dataset PANIC serves as a benchmark for open-world segmentation in the autonomous driving domain, providing a realistic and challenging testbed for evaluating both semantic and instance segmentation in open-world conditions, and paving the way for future research on robust perception in safety-critical environments.

Chapter 8

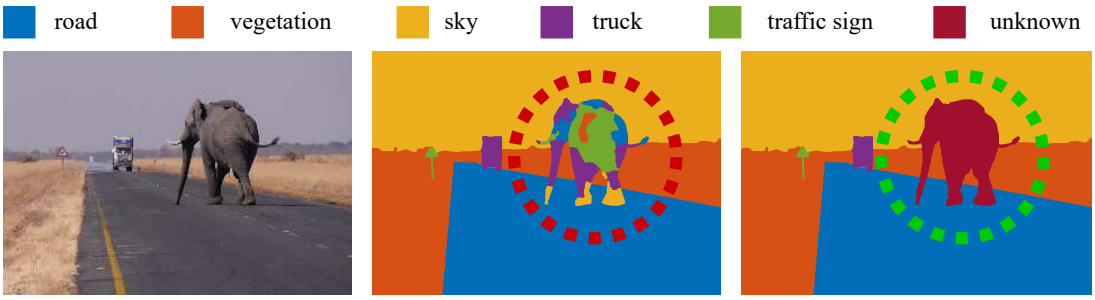
Open-World Semantic Segmentation

As introduced in the previous chapters, open-world perception aims to relax the closed-world assumption and design models that can operate in the inherently open-ended nature of real environments. In contrast to traditional closed-world settings, where all possible categories are assumed to be known in advance, open-world approaches explicitly acknowledge the possibility of encountering novel classes or objects at test time. This ability is a key requirement for deploying reliable systems in the real world. For example, an autonomous vehicle cannot rely solely on a fixed taxonomy of traffic participants, but must also handle unexpected objects without compromising safety.

Thus, in this chapter, we focus specifically on open-world semantic segmentation. This task represents the natural first step in extending segmentation into the open world: before reasoning about objects as individual instances, it is crucial to first establish a semantic understanding of the scene at the pixel level. By answering the fundamental question of *what* is present in the image, open-world semantic segmentation provides the necessary foundation upon which more advanced reasoning can be built. This makes it an essential stepping stone for open-world vision, paving the way for more advanced formulations such as open-world panoptic segmentation, which will be explored in the following chapter.

In the remainder of this chapter, we investigate how to approach open-world semantic segmentation in a neural network setting. Specifically, we extend best-practice strategies for anomaly detection and open-set recognition, originally developed for classification [9, 46, 220], and adapt them to dense prediction tasks.

The main contribution of this chapter is a novel approach for open-world semantic segmentation based on an encoder-decoder convolutional neural network. Our model is designed to simultaneously perform accurate closed-world semantic segmentation while learning discriminative feature descriptors for each



(a) Input RGB (b) Closed-world prediction (c) Open-world prediction

Figure 8.1: Given an image containing a previously-unseen object (a), closed-world methods for semantic segmentation classify the pixels belonging to that object as belonging to the known classes (b, red circle). Our goal is to segment the unknown object and identify it as a semantic class different from the previously known ones (c, green circle).

known class through a dedicated loss function we introduce. This same loss enforces compactness among features of known categories, while allowing sufficient flexibility for the emergence of novel ones. In addition, our network performs anomaly segmentation to identify regions that do not correspond to any known class, effectively distinguishing between known and unknown areas in the scene. By combining these two aspects, the model can build new class descriptors in the unknown regions at test time, enabling it to discover and consistently segment multiple novel categories, as required by the open-world semantic segmentation paradigm introduced in Chapter 6. We illustrate the expected output of our method in Figure 8.1. Beyond identifying unknown regions, our approach also quantifies their similarity to known categories, providing a meaningful measure that can inform downstream tasks. For instance, predicting that an unknown region is semantically similar to a known class such as car or truck can assist motion estimation in tracking, or help mapping systems decide whether to exclude certain regions from the generated map. Through extensive experiments on multiple datasets, including the publicly available SegmentMeIfYouCan benchmark, we demonstrate that our approach achieves state-of-the-art performance on both anomaly segmentation and open-world semantic segmentation, while maintaining strong results on known classes. These results confirm that our open-world design does not compromise closed-world performance, but instead extends it, providing a unified and robust pipeline for scene understanding.

Together, these contributions provide a comprehensive framework for tackling open-world semantic segmentation, bridging the gap between traditional closed-world models and the challenges of real-world perception, where novel objects and categories are constantly encountered.

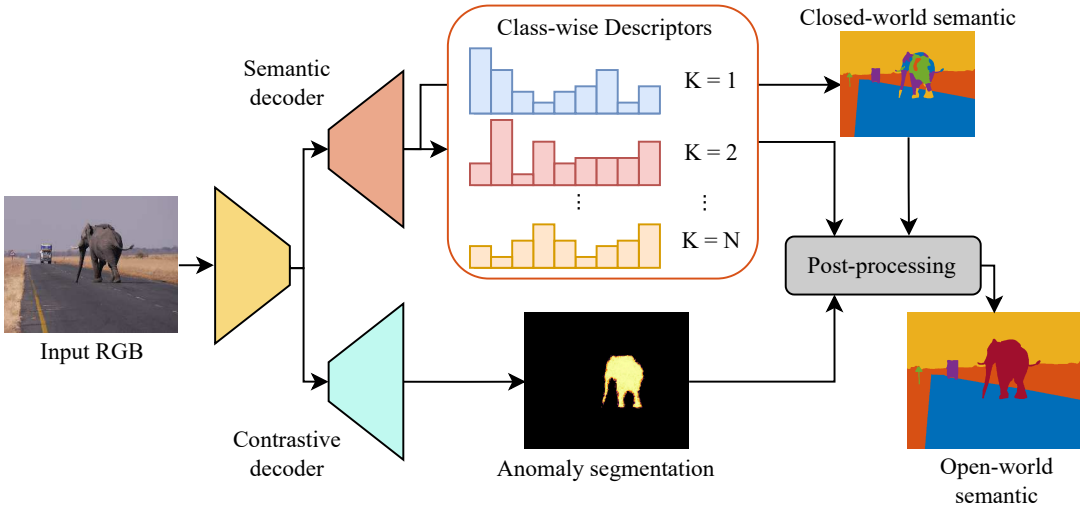


Figure 8.2: Given an RGB image as input, our network processes it by means of an encoder and two decoders. The semantic decoder produces a closed-world semantic segmentation (top right) and a descriptor for each known category. The contrastive decoder provides an anomaly segmentation output. A post-processing phase takes all intermediate results and yields the open-world semantic segmentation prediction.

8.1 Our Approach to Open-World Semantic Segmentation

In this chapter, we tackle the problem of open-world semantic segmentation. In addition to handling known classes, we are interested in segmenting all anomalous areas in an image, and differentiating between potentially multiple novel classes. We propose an approach, shown in Figure 8.2, based on a CNN with one encoder and two decoders. The first decoder tackles semantic segmentation and operates on the feature space so that, for each class, features of pixels belonging to the same class are pushed together. The mean and variance of each individual class descriptor are stored, representing Gaussian distributions that describe known classes. The second decoder performs binary anomaly segmentation. Results are finally merged to obtain open-world semantic segmentation, i.e., anomaly segmentation and novel class discovery.

8.1.1 Network Architecture

Our network for open-world semantic segmentation is composed of one encoder and two decoders. We use a ResNet34 [72] encoder, where the basic ResNet block is replaced with the NonBottleneck-1D block [166], which allows a more lightweight architecture since all 3×3 convolutions are replaced by a sequence of 3×1 and 1×3 convolutions with a ReLU in between. For open-world seg-

mentation, contextual information is valuable. Therefore, we expand the limited receptive field of ResNet by incorporating contextual information using a pyramid pooling module [229] after the encoding part. The features produced will be fed to two separate decoders that share the same structural properties. In order to preserve the lightweight nature of the network, we use three SwiftNet modules [148] where we incorporate NonBottleneck-1D blocks, and two final upsampling modules based on nearest-neighbor and depth-wise convolutions, which reduce the computational load. We use encoder-decoder skip connections after each downsampling stage of the encoder to directly propagate more fine-grained features to the decoder.

8.1.2 Decoder Architectures

Our approach for open-world segmentation builds upon the structure of the CNN we developed, and it exploits the double-decoder architecture for providing accurate segmentation of unknown regions. The first decoder, which we call "semantic decoder", targets semantic segmentation. We additionally manipulate the feature space to create a unique distinct descriptor for each known class. Our goal is to obtain a correct semantic segmentation for the known classes, but also to produce features for each pixel that are similar to the descriptor of a certain class. In this way, we aim to detect as unknown classes all pixels whose feature vectors are substantially different from the descriptor of the class they have been assigned to. The second decoder, which we call "contrastive decoder", leverages the contrastive loss [29] and objectosphere loss [46] together, to place all features of known classes on the surface of a hypersphere while pushing the ones of unknown classes towards its center. In this way, the second decoder provides an anomaly segmentation, where the anomalous regions correspond to previously unseen classes. The two results are finally merged using an automated post-processing operation to obtain the final open-world segmentation.

In the following, we call $\Omega = \{(1,\,1), \ldots, (H,\,W)\}$ the set of pixels in the input image, $Y \in \{1, \ldots, K\}^{H \times W}$ the ground truth closed-world semantic mask, and $\hat{Y} \in \{1, \ldots, K\}^{H \times W}$ the predicted semantic mask, where H and W are the dimensions of the input image. Additionally, we denote with $\Omega_k = \{p \in \Omega \mid Y_p = k\}$ the set of pixels whose ground truth label is k, and with $\hat{\Omega}_k = \{p \in \Omega \mid \hat{Y}_p = Y_p = k\}$ the set of pixels that are true positives for class k, i.e., the set of pixels whose ground truth label and predicted label are k. Finally, in this section, the square of any vector $\boldsymbol{v}$ refers to the element-wise operation (Hadamard product):

$$\boldsymbol{v}^2 = \left[v_1^2,\, \ldots,\, v_n^2\right]^\top . \tag{8.1}$$

Semantic Decoder. The aim of semantic segmentation is to predict a categorical distribution over K classes for all pixels in an image. We follow best

practice and optimize it with the weighted cross-entropy loss

$$\mathcal{L}_{\text{sem}} = -\frac{1}{|\Omega|} \sum_{p \in \Omega} \omega_k \, \boldsymbol{t}_p^\top \log\big(\sigma(\boldsymbol{f}_p^s)\big), \tag{8.2}$$

where ω_k is a class-wise weight computed via the inverse frequency of each class in the dataset, $\boldsymbol{t} \in \mathbb{R}^{H \times W \times K}$ is a one-hot encoded pixel-wise ground truth label, $\boldsymbol{t}_p \in \mathbb{R}^K$ is a one-hot encoded pixel-wise ground truth label at pixel $p \in \Omega$, σ indicates the softmax operation, and $\boldsymbol{f}_p^s$ denotes the pre-softmax feature predicted for pixel p, where the superscript s indicates the semantic decoder.

As mentioned above, we do not only want to perform standard semantic segmentation but also build a class descriptor to bring all pixels belonging to a certain class towards a certain region in the feature space. To achieve this, we accumulate the pre-softmax features, also called activation vectors, of all true positives for each class, where a true positive is a pixel that is correctly segmented. With this, we can store a running average class prototype, or mean activation vector, $\boldsymbol{\mu}_k \in \mathbb{R}^K$ for each class $k \in \{1, \ldots, K\}$:

$$\boldsymbol{\mu}_k = \frac{1}{|\hat{\Omega}_k|} \sum_{p \in \hat{\Omega}_k} \boldsymbol{f}_p^s. \tag{8.3}$$

We also iteratively compute the per-class variance $\boldsymbol{\sigma}_k^2 \in \mathbb{R}^K$ via sum of squares, as

$$\boldsymbol{\sigma}_k^2 = \frac{1}{|\hat{\Omega}_k|} \sum_{p \in \hat{\Omega}_k} \big(\boldsymbol{f}_p^s - \boldsymbol{\mu}_k\big)^2. \tag{8.4}$$

At the beginning of epoch e, we have the means $\boldsymbol{\mu}_k^{e-1}$ and variances $\boldsymbol{\sigma}_k^{e-1}$ accumulated in the previous epoch. At epoch e, we can steer the semantic segmentation to predict, for each pixel with ground truth class k, a feature vector equal to $\boldsymbol{\mu}_k^{e-1}$. For this, we introduce a feature loss function

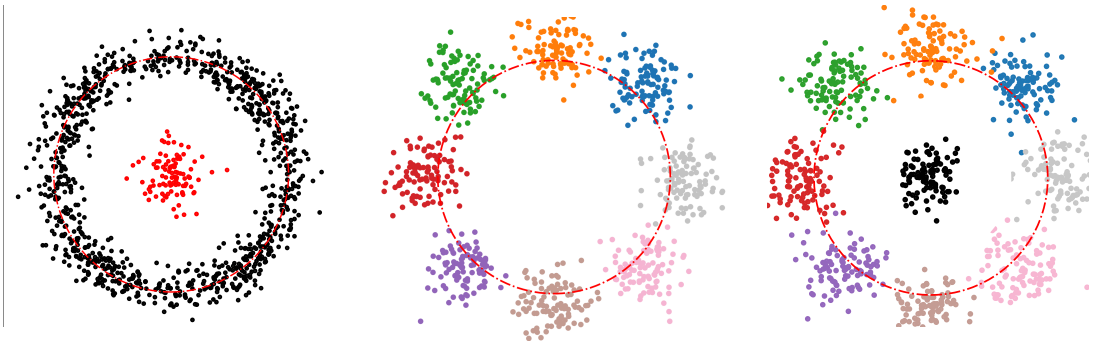
$$\mathcal{L}_{\text{feat}} = \frac{1}{|\Omega|} \sum_{k=1}^{K} \sum_{p \in \Omega_k} \frac{\big\|\boldsymbol{f}_p^s - \boldsymbol{\mu}_k^{e-1}\big\|}{\boldsymbol{\sigma}_k^{e-1}}. \tag{8.5}$$

This loss function is not active during the first epoch since there is no accumulated mean yet. Thus, we perform standard semantic segmentation during the first epoch.

The semantic decoder is thus optimized with a weighted sum of the loss functions introduced above

$$\mathcal{L}_{\text{sdec}} = w_1 \, \mathcal{L}_{\text{sem}} + w_2 \, \mathcal{L}_{\text{feat}}. \tag{8.6}$$

Contrastive Decoder. The contrastive decoder explicitly aims for anomaly segmentation. Given an image of dimensions $H \times W$, where known and unknown



(a) Objectosphere (b) Contrastive (c) Objectosphere + Contrastive

Figure 8.3: 2D visualization of the expected output of the contrastive decoder. The behavior of the objectosphere loss is shown (a), where all points coming from known classes (black) lie around the red circle of radius ξ, see Equation (8.9), and the points from unknown classes lie around the origin (red). The contrastive loss is shown in (b), where features lie on the unit circle. Together, they lead to a behavior similar to the one depicted in (c).

classes are present, the goal of the contrastive decoder is to provide the basis for a binary prediction where 0 corresponds to known classes and 1 to unknown classes. We achieve this by means of a combination of the contrastive loss [29] and the objectosphere loss [46]. First, we compute the mean feature representation $\overline{\boldsymbol{f}}_k$ for class k in the current image as

$$\overline{\boldsymbol{f}}_k = \frac{1}{|\Omega_k|} \sum_{p \in \Omega_k} \boldsymbol{f}_p^c, \tag{8.7}$$

where $\boldsymbol{f}_p^c$ is the feature predicted at pixel p from the contrastive decoder (the equivalent of $\boldsymbol{f}_p^s$ for the semantic one). Then, we compute the contrastive loss $\mathcal{L}_{\text{cont}}$ such that $\overline{\boldsymbol{f}}_k$ approximates the normalized mean representation $\bar{\boldsymbol{\mu}}_k^{e-1}$ of the corresponding class in the previous epoch $\boldsymbol{\mu}_k^{e-1}$ and gets dissimilar from the other classes mean representation:

$$\mathcal{L}_{\text{cont}} = -\sum_{k=1}^{K} \log \frac{\exp\left(\overline{\boldsymbol{f}}_k{}^{\top} \bar{\boldsymbol{\mu}}_k^{e-1} / \tau\right)}{\sum_{i=1}^{K} \exp\left(\overline{\boldsymbol{f}}_k{}^{\top} \bar{\boldsymbol{\mu}}_i^{e-1} / \tau\right)}, \tag{8.8}$$

where τ is a temperature parameter. This way, the loss aims to make the features from the same class consistent with its running mean representation $\boldsymbol{\mu}_k^{e-1}$, while scattering all K classes around the unit hypersphere.

At the same time, we use the objectosphere loss $\mathcal{L}_{\text{obj}}$ over each pixel $p \in \Omega$ given by

$$\mathcal{L}_{\text{obj}} = \begin{cases} \max\left(\xi - \left\|\boldsymbol{f}_p^c\right\|^2, 0\right) & , \text{if } p \in \mathcal{D}_k \\ \left\|\boldsymbol{f}_p^c\right\|^2 & , \text{otherwise} \end{cases}, \tag{8.9}$$

where $\mathcal{D}_k$ is the set of pixels belonging to known classes. The remaining pixels, at training time, reduce to the unlabeled (void) areas of the image. This aims to

make the norm of the feature vector $\left\|\boldsymbol{f}_p^c\right\|$ of pixels belonging to known classes $\mathcal{D}_k$ bigger than a threshold ξ, while the norm of the features of pixels belonging to unknown classes $\mathcal{D}_u$ gets reduced to 0. These two loss functions $\mathcal{L}_{\text{cont}}$ and $\mathcal{L}_{\text{obj}}$ together allows us to optimize towards a situation where the feature vectors of known classes are distributed along the surface of the K-dimensional hypersphere of radius ξ, while the feature vectors of unknown classes gets squashed to 0. A 2D example of the expected behavior is shown in Figure 8.3.

The contrastive decoder is optimized with a weighted sum of the two losses given by

$$\mathcal{L}_{\text{cdec}} = w_3\,\mathcal{L}_{\text{cont}} + w_4\,\mathcal{L}_{\text{obj}}. \tag{8.10}$$

Post-Processing for Anomaly Segmentation. To obtain the open-world predictions at test time, we fuse the outputs of the two decoders. The semantic encoder provides a standard closed-world semantic segmentation but, thanks to the loss function that operates directly on the feature space that we introduced, we can obtain an open-world segmentation. In fact, we computed mean $\boldsymbol{\mu}_k \in \mathbb{R}^K$ and variance $\boldsymbol{\sigma}_k^2 \in \mathbb{R}^K$ of each class, meaning that, for each class, we can easily build a multi-variate normal distribution $\mathcal{N}\big(\boldsymbol{\mu}_k,\, \boldsymbol{\Sigma}_k\big)$, where $\boldsymbol{\mu}_k$ is the mean, and $\boldsymbol{\Sigma}_k = \text{diag}(\boldsymbol{\sigma}_k^2)$ is the covariance matrix, which reduces to the diagonalization of the variance $\boldsymbol{\sigma}_k^2$ under the assumption that all classes are independent. After building the Gaussian model of each class in the dataset, given a pixel p whose predicted feature $\boldsymbol{f}_p^s$ would correspond to class $k, \forall k$, we compute a fitting score by means of the squared exponential kernel

$$s_k(\boldsymbol{f}_p^s) = \exp\left(-\frac{1}{2}(\boldsymbol{f}_p^s - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{f}_p^s - \boldsymbol{\mu}_k)\right). \tag{8.11}$$

Then, for each pixel, we take the highest score

$$s(p) = \max_k s_k(\boldsymbol{f}_p^s), \tag{8.12}$$

and, if it is low, then the pixel of interest is in the tail of the Gaussian, and is considered as a novel class, leading to an open-world prediction $\mathcal{U}_{\text{sem}}$ of the semantic decoder. We can obtain a pixel-wise score $s_{\text{unk},\,p}^{\text{sem}}$ for being unknown $s_{\text{unk},\,p}^{\text{sem}} = 1 - s(p)$.

The contrastive decoder leads to a second open-world prediction $\mathcal{U}_{\text{cont}}$ by considering as unknown all pixels whose feature norm is below a certain threshold. In particular, we can obtain a pixel-wise score $s_{\text{unk},\,p}^{\text{cont}}$ for being unknown

$$s_{\text{unk},\,p}^{\text{cont}} = \max\left(0,\, \left(1 - \frac{\left\|\boldsymbol{f}_p^c\right\|^2}{\xi}\right)\right), \tag{8.13}$$

where $\boldsymbol{f}_p^c$ is the predicted feature at pixel p by the contrastive decoder. This score is 1 when the norm of the feature vector is 0, and 0 when the norm is bigger than ξ, as described in Equation (8.9).

Table 8.1: Comparison between closed-world and open-world model on the known classes of the training datasets. Our open-world approach does not harm closed-world semantic segmentation. All metrics are reported as percentages (%).

Approach	**mIoU**	
	CityScapes	BDDAnomaly
Closed-world	71.1	64.1
Open-world	70.8	62.8

Finally, we fuse the two predictions to obtain a cumulative pixel-wise score for being unknown as

$$s_{\text{unk},\,p} = \frac{1}{2}\left(s^{\text{sem}}_{\text{unk},\,p} + s^{\text{cont}}_{\text{unk},\,p}\right). \tag{8.14}$$

If $s_{\text{unk},\,p}$ is above a threshold δ, the pixel is considered belonging to an unknown class.

Post-Processing for Open-World Semantic Segmentation. When a pixel is considered unknown, we need to store its activation vector and decide whether it belongs to an already-discovered class or a new one. Given the set of mean activation vectors for G unknown classes discovered so far $\mathcal{F} = \{\bar{\boldsymbol{f}}^1_u, \ldots, \bar{\boldsymbol{f}}^G_u\}$, we take the vector $\boldsymbol{f}^g_u$ that minimizes the distance from the querying vector. If the distance between $\bar{\boldsymbol{f}}^g_u$ and $\boldsymbol{f}^s_p$ is below a threshold η, then the pixel belongs to this class, and the mean activation vector gets updated, otherwise it creates a new unknown class $\bar{\boldsymbol{f}}^{g+1}_u$. This allows us to have a virtually unlimited number of novel classes.

8.1.3 Class Similarity

As a byproduct of the open-world segmentation, our method can also predict the most similar known category for each unknown sample. As explained in Section 8.1.2, it does not suffice for a feature vector to have the highest activation in the $k-$th spot for being matched to class k. A sample can have the highest activation for a certain class k, but its score computed with Equation (8.11) is higher for another class $\tilde{k} \neq k$, meaning that the sample is more inside the area of influence of class $\tilde{k}$ despite having a higher activation on class k. As the most similar class, we propose to choose the one that provides the highest score given by $\tilde{k} = \text{argmax}_k\, s_k(\boldsymbol{f}_p)$.

Table 8.2: Results from the public leaderboard of the SegmentMeIfYouCan benchmark. We separate methods that use external data, i.e. out of distribution (OoD) data with semantic labels different from the ones in Cityscapes [27], during training. Our approach ranked overall top 1 for FPR95, PPV and mean F1, and top 6 for AUPR and sIoU (fourth and sixth, respectively) on January 31st, 2024. Best results in bold. All metrics are reported as percentages (%).

Approach	OoD	Pixel-Level		Component-Level		
		AUPR	FPR95	sIoU gt	PPV	mean F1
DenseHybrid [61]	✓	78.0	9.8	54.2	24.1	31.1
RbA [141]	✓	**94.5**	4.6	**64.9**	47.5	51.9
ObsNet + LAA [13]	✗	75.4	26.7	44.2	52.6	45.1
Maskomaly [1]	✗	93.4	6.9	55.4	51.2	49.9
RbA [141]	✗	86.1	15.9	56.3	41.4	42.0
ContMAV (ours)	✗	90.2	**3.8**	54.5	**61.9**	**63.6**

8.2 Experimental Evaluation

The main focus of this work is an approach for open-world semantic segmentation that also provides a measure of class similarity. We present experiments to show the capabilities of our method. The results of our experiments show that: (i) our model achieves state-of-the-art results for anomaly segmentation while performing competitively on the known classes, (ii) our approach can distinguish between different unknown classes, and (iii) our approach can provide a similarity score for each novel class to the known ones.

8.2.1 Experimental Setup

Datasets and metrics. We use three datasets for validating our method: SegmentMeIfYouCan [27], BDDAnomaly [73], and our dataset PANIC, which we introduced in the previous chapter. SegmentMeIfYouCan relies on the semantic annotations of Cityscapes [36], and offers a public benchmark with a hidden test set for anomaly segmentation, where the goal is to segment objects that are not present on Cityscapes. Annotations are binary, since each object is either known or unknown. BDDAnomaly is a reorganization of BDD100K [221], where all images containing the classes train, motorcycle, and bicycle have been discarded from the training and validation set to create an open-world test set. Since ground truth data is available for this dataset, we use it for ablation studies and experiments on class similarity. Additionally, we report results on a further modification of BDDAnomaly proposed by Besnier *et al.* [13], which we

Table 8.3: Anomaly segmentation results on (a) BDDAnomaly and (b) BDDAnomaly*. Best results in bold. All metrics are reported as percentages (%).

(a)

Approach	AUPR	FPR95
MaxSoftmax [75]	3.7	24.5
Background [18]	1.1	40.1
MC Dropout [52]	4.3	16.6
Confidence [45]	3.9	24.5
MaxLogit [73]	5.4	14.0
ContMAV (ours)	**96.1**	**6.9**

(b)

Approach	AUPR	FPR95
MaxSoftmax [75]	80.1	63.5
Background [18]	75.3	68.1
MC Dropout [52]	82.6	61.1
ODIN [104]	81.7	60.6
ObsNet + LAA [13]	82.8	60.3
ContMAV (ours)	**92.9**	**43.9**

call BDDAnomaly*, where only train and motorcycle are considered as unknown classes.

We evaluate our methods with the metrics proposed in the SegmentMeIfYouCan public benchmark for pixel-level performance: area under the precision-recall curve (AUPR) and the false positive rate at a true positive rate of 95% (FPR95). For SegmentMeIfYouCan, we also report component-level metrics provided by the benchmark. As explained, our approach is not limited to anomaly segmentation, but performs open-world semantic segmentation. Thus, we also report the mean intersection-over-union (mIoU) on the known classes, to show that our open-world segmentation approach does not underperform on the known classes when compared to the closed-world equivalent (see Table 8.1). Finally, we report the mIoU between the newly-discovered classes and their respective highest-overlapping ground truth class to be discovered.

In all tables, we call our method "ContMAV", where "Cont" indicates the contrastive decoder and "MAV" the mean activation vectors.

Training details and parameters. In all experiments, we use the one-cycle learning rate policy [181] with an initial learning rate of 0.004. We perform random scale, crop, and flip data augmentations, and optimize with Adam [86] for 500 epochs with batch size 8. We set $\xi = 1$, $\delta = 0.6$, $\tau = 0.1$, $\eta = 0.5$, and loss weights $w_1 = 0.9$, $w_2 = 0.1$, $w_3 = 0.5$, and $w_4 = 0.5$. For SegmentMeIfYouCan, we train only on Cityscapes. For BDDAnomaly, we train only on the training set of BDDAnomaly itself.

8.2.2 Anomaly Segmentation

The first set of experiments shows that our model achieves state-of-the-art results in anomaly segmentation. Here, we aim for a binary segmentation between known classes and previously unseen classes. We report results on SegmentMeIfY-

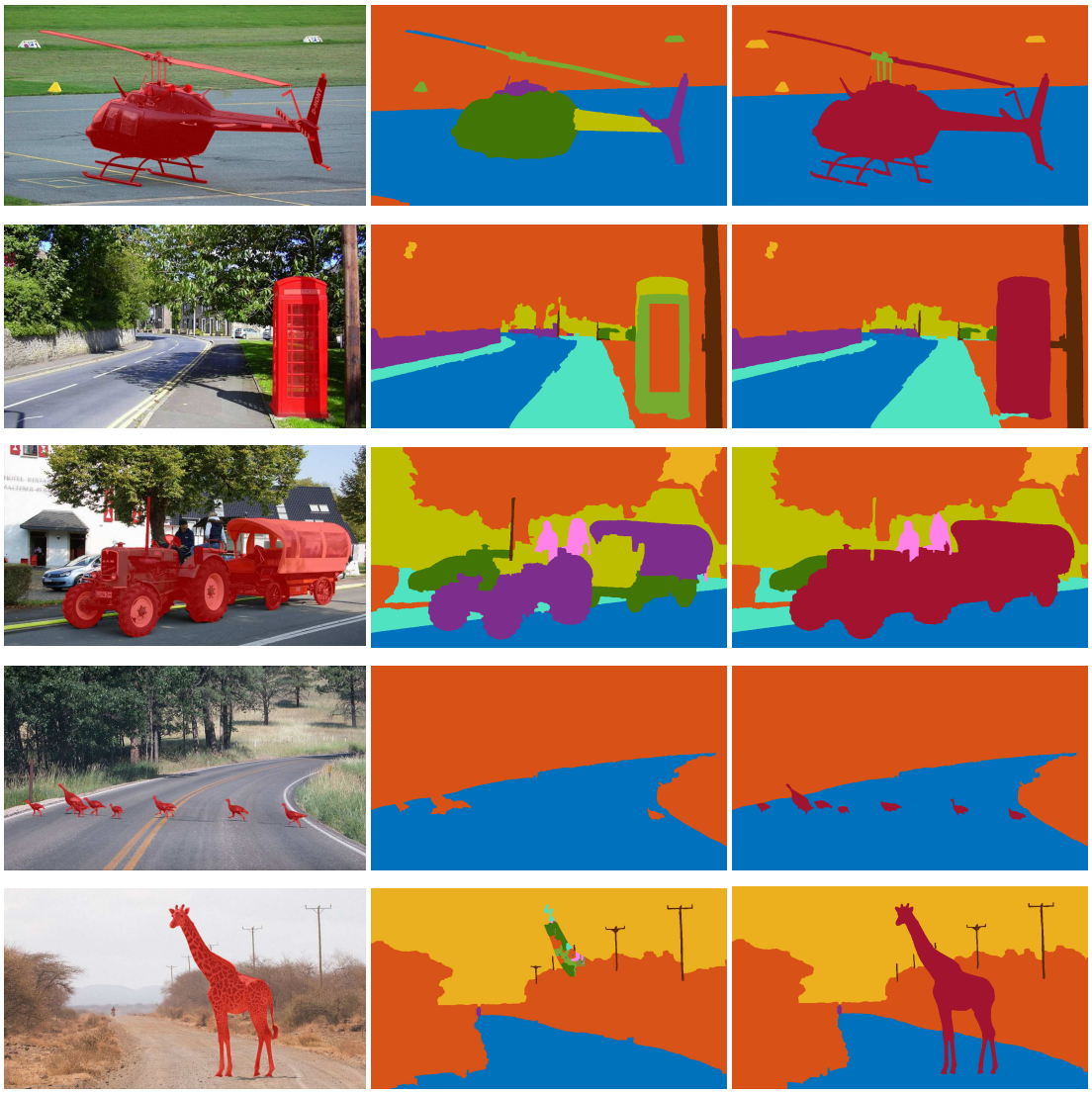
Table 8.4: Anomaly segmentation results on the hidden test set of our dataset, PANIC. All metrics are reported as percentages (%). Public competition at `https://www.codabench.org/competitions/4561`.

Approach	Pixel-Level		Component-Level		
	AUPR	FPR95	sIoU gt	PPV	mean F1
ContMAV (ours)	91.7	66.4	15.0	72.1	24.2

ouCan in Table 8.2, BDDAnomaly in Table 8.3a, BDDAnomaly* in Table 8.3b, and PANIC in Table 8.4. On SegmentMeIfYouCan, our method outperforms all baselines on FPR95 and ranks top 6 on the public leaderboard for AUPR. On the BDD datasets, our method outperforms all baselines on both metrics, providing compelling results for the task of anomaly segmentation. For the BDD datasets, in this experiment, we treat all the unknown categories as the same unknown class, without focusing on the fact that they are, originally, separate classes. Our approach shows compelling results for anomaly segmentation, successfully dealing with challenging situations such as the case in which a known and an unknown object are overlapping, see Figure 8.4. While SegmentMeIfYouCan is designed specifically for anomaly segmentation, having images where the anomalous objects are prominent, the BDD dataset is more challenging since objects belonging to bicycle or motorcycle can appear in very small areas of the image, see Figure 8.5, making the task of anomaly segmentation more challenging and harder to solve.

8.2.3 Open-World Semantic Segmentation

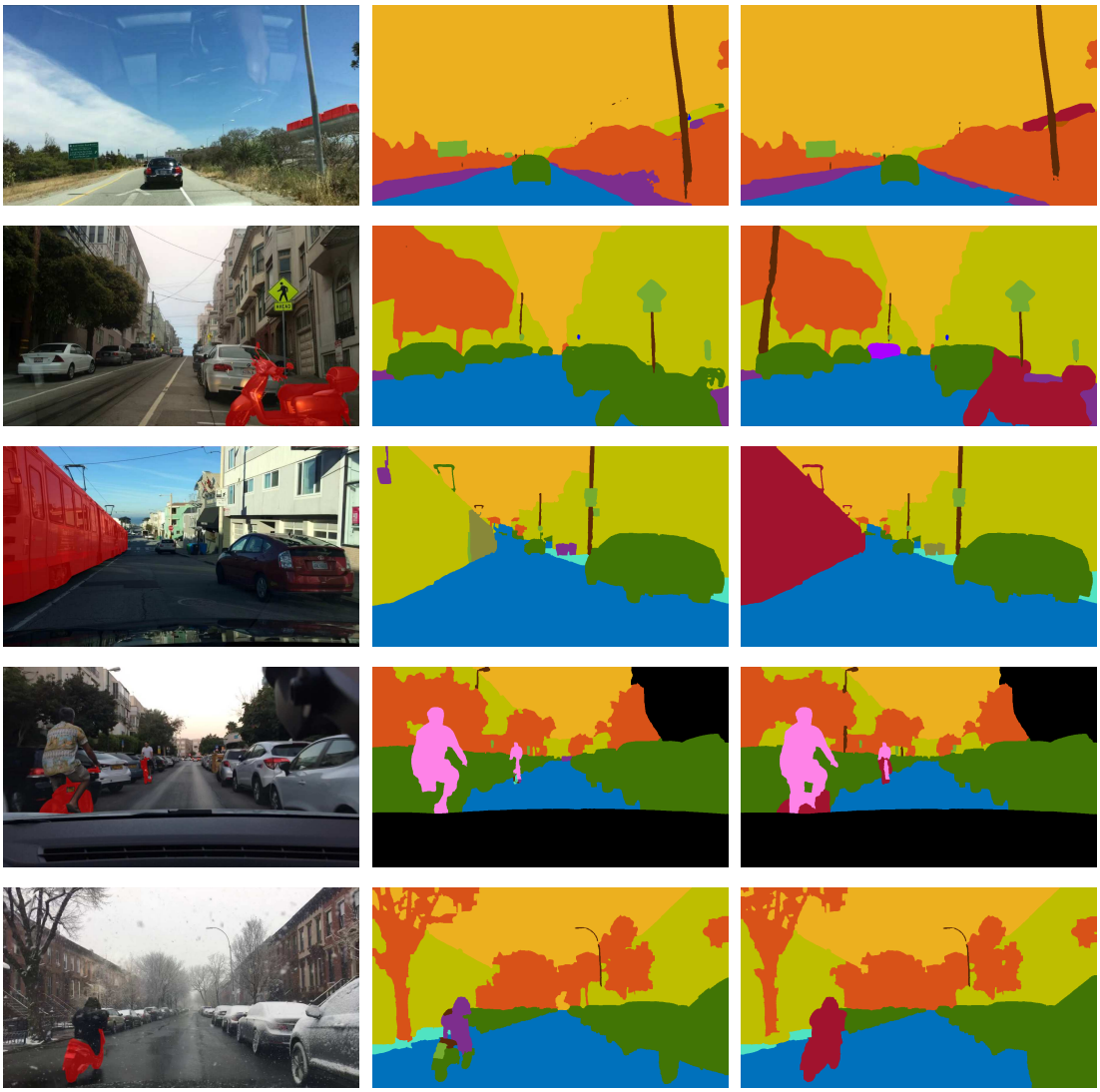
The second experiment illustrates that our approach is capable of distinguishing between different unknown classes, rather than only stating whether something is known or unknown. We achieve this thanks to the feature loss function we introduced in Equation (8.7). We conduct this experiment on BDDAnomaly, since the test set is manually generated excluding images from the training and the validation set, and thus the ground truth labels are available, and also on our proposed dataset PANIC, see Chapter 7. On BDDAnomaly, we report results for our method together with results we would achieve without the feature loss function. Since this task is uncommon in the literature, we also report one baseline approach as a performance lower bound, which uses the background class for the unknowns and performs K-means clustering in the feature space for this class. As a performance upper bound, we report the mIoU of the three classes in closed-world settings on the original BDD100K, where there is no unknown, but every class is present at training time. Results are shown in Table 8.5 for BDDAnomaly and



(a) Input RGB with overlayed anomaly mask (b) Closed-world prediction (c) Open-world prediction

Figure 8.4: Anomaly segmentation results from the validation set of SegmentMeIfYouCan. We show the input RGB overlayed with the ground truth unknown mask (a), the prediction of our closed-world model (b), and the prediction of our approach for open-world segmentation (c). In the open-world prediction, the unknown class is shown in red. Notice how the two models, that are both trained on CityScapes, perform similarly on known classes, demonstrating that our approach does not degrade closed-world performance.

in Table 8.6 for PANIC. Our approach outperforms the baseline and provides satisfying results in distinguishing among different classes. Additionally, removing the feature loss function also provides good results for open-world segmentation, outperforming the baseline by a large margin.



(a) Input RGB with overlayed anomaly mask (b) Closed-world prediction (c) Open-world prediction

Figure 8.5: Anomaly segmentation results from the test set of BDDAnomaly. We show the input RGB overlayed with the ground truth unknown mask (a), the prediction of our closed-world model (b), and the prediction of our approach for open-world segmentation (c). In the open-world prediction, the unknown class is shown in red. Notice how the two models, that are both trained on BDDAnomaly, perform similarly on known classes, demonstrating that our approach does not degrade closed-world performance.

8.2.4 Experiments on Class Similarity

The third experiment shows that our approach successfully assigns to each novel class its most similar known category. For this experiment, we need to select a ground truth category that represents the class with the highest similarity to the class of interest. We decide to use BDDAnomaly* because, differently from BDDAnomaly, it does not contain the class bicycle among the unknowns. In

Table 8.5: Open-world semantic segmentation results on BDDAnomaly. Best results in bold. All metrics are reported as percentages (%).

Approach	**IoU**			**mIoU**
	Train	Motorcycle	Bicycle	
Background + cluster	0	32.3	32.8	21.7
ContMAV (no feat loss)	48.1	53.8	39.9	55.7
ContMAV (with feat loss)	**62.4**	**62.2**	**56.8**	**60.5**
Closed-world	72.3	69.3	60.9	67.5

Table 8.6: Open-world semantic segmentation results on the hidden test set of our dataset, PANIC. All metrics are reported as percentages (%). Public competition is available at `https://www.codabench.org/competitions/4563`.

Approach	**Unknown Classes**		
	mIoU_u	Completeness	Homogeneity
ContMAV [186]	15.8	81.4	77.3

BDDAnomaly, the only vehicle that is present among the known categories is car, and in fact, our method on BDDAnomaly achieves, for bicycle, a 43.2% similarity score with it. However, car is not the perfect similarity candidate for bicycle, due to their extremely different appearance, as well as their different behavior as a traffic participant. A more modern dataset, with more vehicle classes such as electric scooters, would provide better candidates for class similarity. For this reason, we select BDDAnomaly*, in which we have "train" and "motorcycle" as unknown categories. We manually created a lookup table (see Table 8.7) in which each class is assigned a ground truth label indicating its most similar category.

We report one baseline that performs semantic segmentation on the known classes and has a stack of linear layers on the pre-softmax features that learns the lookup table. We compare with our same approach, but taking the class that has the highest activation as most similar. We report pixel-wise accuracy results in Table 8.8. The results show that the classifier does not generalize well to the unknown classes. Considering only the highest activation is better than the "specialized" classifier, but still it is not a reliable measure of class similarity. We report qualitative results on the test set of BDDAnomaly for class similarity in Figure 8.6.

Table 8.7: Look-up table for class similarity. The unknowns are specified in the context of BDDAnomaly*.

	Category	Most Similar	Category	Most Similar
known	Road	Sidewalk	Pole	Sign
	Sidewalk	Road	Light	Sign
	Building	Wall	Vegetation	Terrain
	Wall	Fence	Terrain	Vegetation
	Fence	Wall	Sky	–
	Person	Rider	Rider	Person
	Car	Truck	Truck	Bus
	Bus	Truck	Bicycle	–
unknown	Train	Truck	Motorcycle	Car

Table 8.8: Class similarity results on BDDAnomaly*. Best results in bold. All metrics are reported as percentages (%).

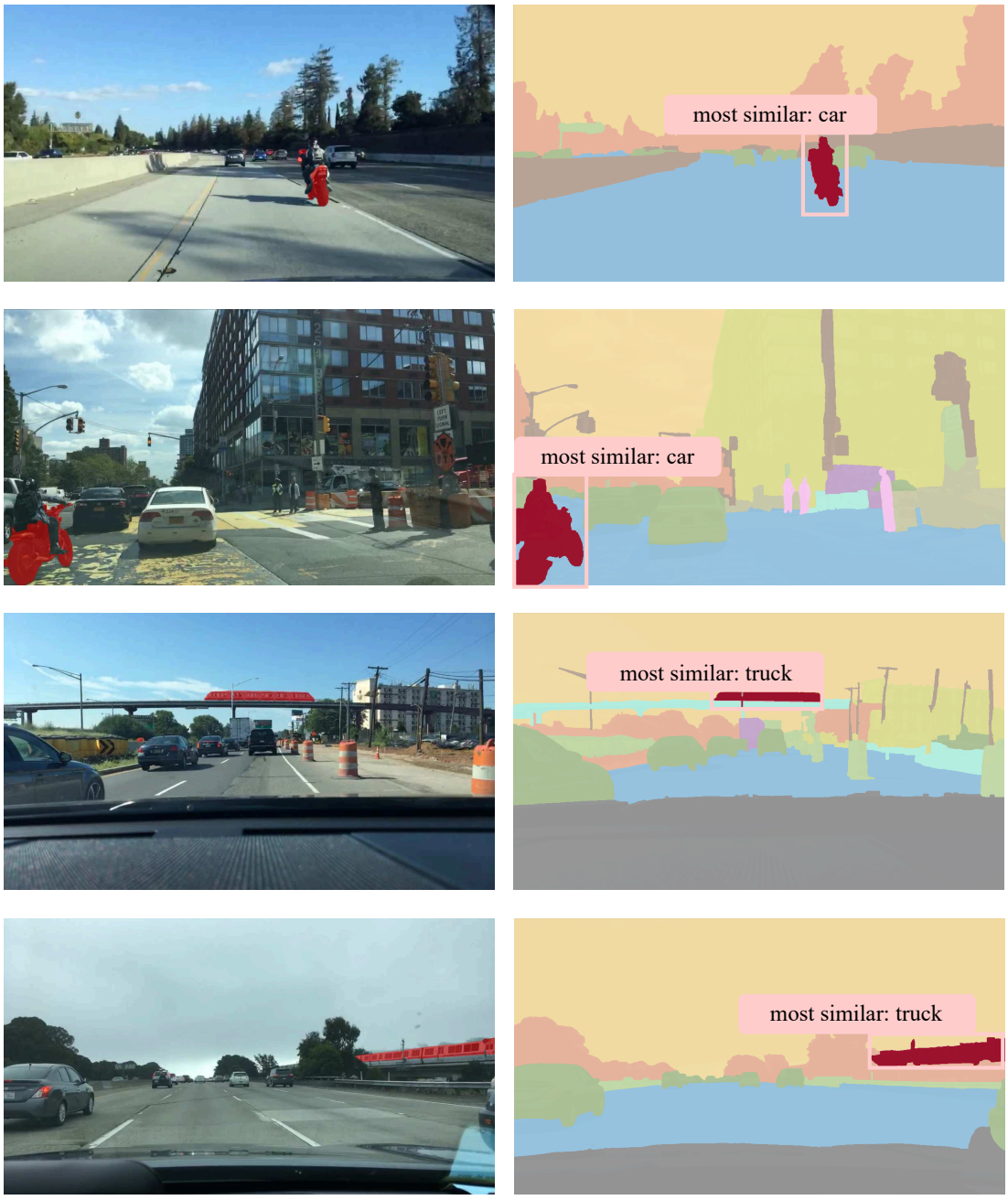
Approach	Accuracy		mAccuracy
	Motorcycle	Train	
Baseline	12.5	9.8	11.2
ContMAV with MA	39.9	27.6	33.8
ContMAV	**58.9**	**49.9**	**54.4**

8.2.5 Ablation Studies

Finally, we provide ablation studies to investigate the contribution of the modules we introduced. We refer to each ablation study in the tables by the letter in the first column.

8.2.5.1 Anomaly Segmentation

First, we perform an ablation study on the anomaly segmentation pipeline, reported in Table 8.9. We investigate the contribution of the feature loss $\mathcal{L}_{\text{feat}}$, of the Gaussian post-processing described in Section 8.1.2, and of the contrastive decoder. We evaluate different post-processing strategies. The first strategy is a softmax thresholding strategy where we consider a pixel as unknown if it has two or more activations above a threshold. The second strategy is based on the maximum softmax activation only and categorizes a pixel as unknown if its maximum activation is below a certain threshold. These two strategies yield similar performance, which is an expected outcome since they both rely on the standard final output vector. In the table, we can see that the thresholding strategy alone (A)



(a) Input RGB with overlayed anomaly mask

(b) Class similarity prediction

Figure 8.6: Class similarity results from the test set of BDDAnomaly. We show the input RGB overlayed with the ground truth unknown mask (a) and the prediction of our approach for class similarity (b). In the open-world prediction, the unknown class is shown in red, and the overall semantic segmentation is shown in transparency.

has poor results, and its performance with the feature loss (B) is close to the performance of the maximum activation strategy with feature loss (E). Additionally, we notice how the thresholding without the feature loss but with the contrastive

Table 8.9: Ablation study on our anomaly segmentation pipeline on BDDAnomaly. $\mathcal{L}_{\text{feat}}$ refers to the feature loss in Equation (8.5), and D_{cont} to whether the contrastive decoder is enabled or not. We also indicate the post-processing operation used for obtaining the open-world prediction: softmax thresholding, maximum activation, D_μ for the minimum distance from the mean activation vector, and the Gaussian inference described in Section 8.1.2. Best results in bold. All metrics are reported as percentages (%).

	$\mathcal{L}_{\text{feat}}$	D_{cont}	Post-processing	AUPR	FPR95
A			Thresholding	46.9	93.9
B	✓		Thresholding	76.4	88.6
C		✓	Thresholding	91.8	70.7
D	✓	✓	Thresholding	94.1	54.4
E	✓		Max activation	75.9	89.9
F	✓	✓	Max activation	93.9	57.6
G		✓	–	91.8	69.7
H	✓		D_μ	94.2	57.0
I	✓	✓	D_μ	94.8	29.8
J	✓		Gaussian	94.2	55.8
K	✓	✓	Gaussian	**96.1**	**6.9**

decoder (C) leads to better performance, which is, however, extremely similar to that of the contrastive decoder only (G), suggesting that the contrastive decoder alone is better than a softmax thresholding strategy for this task. A further improvement comes from putting together the feature loss and the contrastive decoder, which leads to better results with both thresholding (D) and maximum activation (F). The other two post-processing strategies we employ are based on the output of the feature loss. One takes the minimum distance D_μ of the activation vector from the mean activation vectors we built during training, while the last one is the Gaussian querying. They lead to similar performance when the contrastive decoder is not used (H and J), and yield the top 2 performance when the contrastive decoder is used (I and K). The Gaussian querying provides a further improvement and achieves the best performance for this task.

8.2.5.2 Class Similarity

The second ablation study targets the class similarity, and is reported in Table 8.10. The presence of the contrastive decoder does not substantially improve the performance, since the class similarity originates from the semantic decoder. Still, numbers when the contrastive decoder is active (M, O, Q) or inactive (L, N,

Table 8.10: Ablation study on our class similarity approach on BDDAnomaly*. $\mathcal{L}_{\text{feat}}$ refers to the feature loss in Equation (8.5), and D_{cont} to whether the contrastive decoder is enabled or not. We also indicate the post-processing operation used for obtaining the open-world prediction: maximum activation, D_μ for the minimum distance from the mean activation vector, and the Gaussian inference described in Section 8.1.2. Best results in bold. All metrics are reported as percentages (%).

	$\mathcal{L}_{\text{feat}}$	D_{cont}	Post-processing	**Accuracy**		**mAccuracy**
				Motorcycle	Train	
L	✓		Max activation	38.4	25.9	32.2
M	✓	✓	Max activation	39.9	27.6	33.8
N	✓		D_μ	53.5	41.7	47.6
O	✓	✓	D_μ	54.3	42.1	48.2
P	✓		Gaussian	57.8	48.6	53.2
Q	✓	✓	Gaussian	**58.9**	**49.9**	**54.4**

P) are slightly different since the contrastive decoder affects the shared encoder via backpropagation. The performance of class similarity is poor when we rely on the standard maximum activation (L and M), while it improves when it is based on the minimum distance D_μ of the activation vector from the mean activation vectors built during training (N and O). The Gaussian post-processing achieves the best performance for both classes (P and Q), proving the effectiveness of our approach.

8.2.5.3 Hyperparameters

Hyperparameter search is usually a challenging problem when it comes to training neural networks. Usually, they are chosen empirically, and only the configuration that works best is reported. In the following, we try to give some insight into our choice of hyperparameters and the reasoning behind them. We provide an analysis on the four hyperparameters (ξ, δ, τ, and η) in the following.

As discussed in Section 8.1.2, in the paragraph dedicated to the contrastive decoder, ξ is the radius of the hypersphere created by the objectosphere loss [46] in Equation (8.9). In principle, this hyperparameter could take any value. However, we pair the objectosphere loss to the contrastive loss [29] in Equation (8.8), which aims to distribute all feature vectors on the unit sphere. Thus, we expect that any choice of ξ that is different from 1 would harm performance, since it would reduce the synergy between the two loss functions operating on the same decoder. We report an experiment about this in Table 8.11. When $\xi < 1$, the performance is not dramatically harmed because the objectosphere loss aims to make

Table 8.11: Anomaly segmentation results on BDDAnomaly with different choices of the parameter ξ for the radius of the objectosphere loss in Equation (8.9). Best results in bold. All metrics are reported as percentages (%).

Approach	AUPR	FPR95
ContMAV ($\xi = 0.75$)	92.2	18.7
ContMAV ($\xi = 1.25$)	83.4	55.2
ContMAV ($\xi = 1$)	**96.1**	**6.9**

Table 8.12: Anomaly segmentation results on BDDAnomaly with different choices of the parameter δ, used to threshold the "unknown-ness" scores in Section 8.1.2. Best results in bold. All metrics are reported as percentages (%).

Approach	AUPR	FPR95
ContMAV ($\delta = 0.4$)	86.6	41.2
ContMAV ($\delta = 0.8$)	89.1	30.1
ContMAV ($\delta = 0.6$)	**96.1**	**6.9**

the norm of the features belonging to the known pixels greater than ξ. Thus, the two losses do not work against each other. In contrast, when $\xi > 1$, the two loss functions try to achieve two tasks which are incompatible (features on the unit circle and, at the same time, with norm greater than 1), and performance suffers.

The threshold δ, which we also introduced in Section 8.1.2, in the paragraph dedicated to the post-processing, is our "unknown-ness threshold". In fact, we obtain a score $s_{\text{unk}} \in [0, 1]$ and have to decide whether a pixel belongs to an unknown category based on this score. The score is given by

$$s_{\text{unk}} = \frac{1}{2}\left(s_{\text{unk}}^{\text{sem}} + s_{\text{unk}}^{\text{cont}}\right), \tag{8.15}$$

where $s_{\text{unk}}^{\text{seg}}$ and $s_{\text{unk}}^{\text{cont}}$ are the scores coming from the semantic and the contrastive decoders, respectively. Notice that, since the final score is a standard mean of the two, setting the threshold to a low value would make us label a pixel as unknown also in the case in which only one score is high but the other is not. This would create a lot of false positives, and we expect performance aligned with models G and J in Table 8.9. Those two models, in fact, only have one active decoder, and setting a low δ causes a similar behavior. Setting the threshold too high is, in contrast, achievable only when both decoder heads are very confident in their prediction of unknown, and it could cause a high number of false negatives. Thus, we choose $\delta = 0.6$, which is a good compromise and provides good results, see Table 8.12.

Table 8.13: Class discovery results on BDDAnomaly with different choices of the parameter η, that governs the creation of novel categories over the known and already-discovered ones. For each class of interest, the discovered one with greater IoU is chosen and reported, as well as the total number of unknown categories created N_U. Best results in bold. All metrics are reported as percentages (%).

Approach	**mIoU**			N_U
	Train	Motorcycle	Bicycle	
ContMAV ($\eta = 0.3$)	0.0	23.4	0.0	1
ContMAV ($\eta = 0.9$)	30.5	31.1	18.9	12
ContMAV ($\eta = 0.6$)	**62.4**	**62.2**	**56.8**	4

We do not optimize the temperature parameter of the contrastive loss τ and perform all experiments with $\tau = 0.1$, as suggested by Chen *et al.* [29].

The hyperparameter η, also introduced in Section 8.1.2, in the paragraph dedicated to the post-processing, does not affect the prediction of a pixel as unknown, but it plays a role in the class discovery. In fact, it represents the minimum distance needed to decide whether a feature categorized as unknown is a class of its own and does not belong to any of the already-discovered new classes. Setting this threshold heavily depends on the data distribution. A very high threshold would create a lot of classes, and its usefulness would be limited. On the other hand, a low threshold would put all classes together, providing nothing more than an anomaly segmentation. We report results in Table 8.13, where we also report the number N_U of new classes created, for which the ground truth value is 3 (i.e., the number of unknown classes in BDDAnomaly, namely bicycle, motorcycle, and train).

8.2.6 Analysis of Contrastive Decoder

The contrastive decoder, which we explain in detail in Section 8.1.2, is optimized with a combination of two loss functions, namely the objectosphere and the contrastive loss. Figure 8.3 intuitively shows the idea behind it, and what the ideal output in the 2D case would be. However, the feature vectors that the contrastive decoder predicts are K-dimensional, where K is the number of known classes (i.e., 19 in our case). In order to verify whether the output of the decoder is aligned with our expectation, we define two thresholds ζ and ρ. Then, given $\boldsymbol{f}_p^d$, i.e., the feature predicted at pixel p from the contrastive decoder, we want $1 - \zeta < ||\boldsymbol{f}_p^d||_2 < 1 + \zeta$ for all $\boldsymbol{f}_p^d$ whose ground truth label is a known class, and $||\boldsymbol{f}_p^d||_2 < \rho$ for all $\boldsymbol{f}_p^d$ whose ground truth label is an unknown class. The former means that the norms of the vectors belonging to known classes should be in a

Table 8.14: Architectural efficiency reported in terms for number of floating point operations (GFLOPs) and number of trainable parameters of the network.

Approach	GFLOPs	Training Parameters
Maskomaly [75]	937	215M
Mask2Anomaly [18]	258	**23M**
ContMAV (ours)	**84**	48M

"tube" of radius ζ around 1, which is our ξ parameter, explained in Table 8.11. The latter means that the norms of the vectors belonging to unknown classes (which, at training time, are the unlabeled portions of the image), should be smaller than ρ. We choose $\zeta = 0.2$ and $\rho = 0.4$, and we find that 86.5% of the vectors belonging to known classes fall into the tube, and 79.9% of the vectors belonging to unknown classes are smaller than ρ. This verifies that the output is aligned with our expectations. To visually show the result, we would need to apply a dimensionality reduction approach such as principal component analysis. However, linear dimensionality reduction techniques always lead to a loss of information, and the new dimensions may offer no concrete interpretability.

8.2.7 Architectural Efficiency

As pointed out in Section 8.1.1, we designed our neural network in order to be lightweight and faster at inference time, allowing inference on an image at 10 Hz. Additionally, we report the number of parameters and the GFLOPs of our model together with two state-of-the-art models from the SegmentMeIfYouCan public benchmark with code available in Table 8.14. We show that our architecture is competitive and performs very well in terms of efficiency.

8.3 Conclusion

In this chapter, we presented a novel approach for open-world semantic segmentation on RGB images based on a double-decoder architecture. Our approach enables autonomous systems to move beyond the restrictive closed-world paradigm by discovering and segmenting novel semantic categories that were never observed during training. Our method manipulates the feature space of the semantic segmentation decoder to identify novel classes, while also indicating the most similar known categories to the newly discovered ones. Through extensive experiments on multiple datasets, we demonstrated the effectiveness of our approach compared to existing baselines, showing that the proposed strategy can reliably detect anomalous regions and distinguish between different novel categories. These

results confirm the potential of our double-decoder design as a strong foundation for open-world segmentation.

Despite these promising results, our method still presents limitations that highlight avenues for further research. The semantic decoder, for instance, relies on class prototypes whose dimensionality is tied to the number of known training categories. In scenarios with only a few training classes, this representation becomes low-dimensional and may lack sufficient discriminative power for reliably discovering a wide range of novel categories. The contrastive decoder, on the other hand, leverages unlabeled pixels as "known unknowns" [9] to train the objectosphere loss, which means it inherently assumes the presence of such unlabeled regions. On fully annotated datasets, this assumption no longer holds, potentially reducing its effectiveness. Moreover, while our work advances open-world semantic segmentation by addressing anomaly segmentation and novel class discovery, it does not yet extend to instance-level reasoning. Thus, our current framework cannot provide a complete panoptic understanding of a scene.

These limitations naturally motivate the next stage of this thesis, where we move beyond semantic-level reasoning and tackle the more comprehensive task of open-world panoptic segmentation. In doing so, we aim not only to extend our approach to simultaneously handle semantics and instances but also to address the structural weaknesses identified in our current design. The following chapter will therefore focus on bridging this gap, advancing toward a holistic open-world perception framework capable of segmenting both known and novel categories at the semantic and instance levels, thereby providing a richer and more reliable understanding of complex, real-world environments.

Chapter 9

Open-World Panoptic Segmentation

Open-world semantic segmentation, addressed in the previous chapter, represents the natural first step toward moving from closed-world to open-world scene interpretation. Under the closed-world assumption, models are constrained to segment only the categories and objects encountered during training, while ignoring the possibility of encountering objects not seen during training time. In contrast, the open-world semantic segmentation method presented in the previous chapter lifts this restriction by accounting for the appearance of novel semantic classes at test time and enabling them to be incrementally discovered and consistently segmented. This ability to go beyond the training vocabulary is crucial, as it equips models with the flexibility needed to operate in real-world environments.

However, semantic segmentation alone is not sufficient for holistic scene understanding. While recognizing novel categories at the pixel level is necessary, many downstream tasks, such as motion planning, rely on reasoning about objects as distinct instances. For this reason, in this final chapter, we move beyond semantics and target open-world panoptic segmentation. In this setting, given a single RGB image, our convolutional neural network must assign both, a semantic category and an instance ID to every pixel, while simultaneously discovering categories that never appeared during training. This formulation leaves no ambiguity: rather than being limited to the set of known training categories, the model must be able to segment arbitrary objects at test time as novel. Moreover, following the paradigm of semantic segmentation, different anomalies are assigned to different novel categories, while anomalies of the same kind are grouped under the same novel category but receive distinct instance IDs.

To achieve this, we propose Con2MAV, a fully convolutional neural network-based approach that addresses the open-world panoptic segmentation problem.

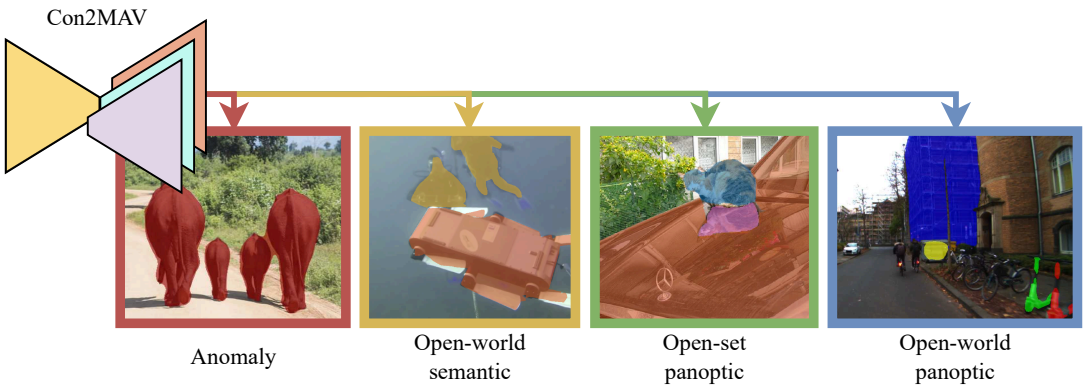


Figure 9.1: Our proposed approach, Con2MAV, is able to tackle multiple open-world tasks and segment unknown objects and categories in multiple datasets spanning multiple domains. In the figure, we show predictions on (left to right) SegmentMeIfYouCan [27], SUIM [80], COCO [106], and PANIC. We show in color only the unknown mask.

Our approach achieves state-of-the-art results across all four open-world segmentation tasks introduced in Chapter 6, namely anomaly segmentation, open-world semantic segmentation, open-set panoptic segmentation, and open-world panoptic segmentation. We extensively evaluate Con2MAV on multiple public datasets, including SegmentMeIfYouCan [27], COCO [106], BDDAnomaly [73], SUIM [80], and also PANIC, our autonomous driving dataset introduced in Chapter 7. Examples of Con2MAV predictions across different tasks and datasets are presented in Figure 9.1, that shows the flexibility and robustness of our approach across different domains.

Con2MAV extends our previous approach, ContMAV [186], a fully convolutional neural network designed for open-world semantic segmentation, which we discussed in the previous chapter. While ContMAV demonstrated the feasibility of tackling an open-world segmentation problem with a purely convolutional neural network, it also suffered from certain limitations. In particular, its reliance on several manually-tuned thresholds introduced fragility and hindered adaptability across datasets. Moreover, the dimensionality of the activation vector was tied to the number of known categories, which proved restrictive, especially when the number of known categories was limited, making the feature vector collapse to a few dimensions. Con2MAV overcomes these limitations, while also broadening the scope from open-world semantic to open-world panoptic segmentation. We incorporated an additional decoder, dedicated to segmenting individual objects, thereby addressing open-world panoptic segmentation in a unified, fully-convolutional framework.

In sum, our contributions can be summarized as follows: we present a fully convolutional neural network that achieves state-of-the-art performance on all

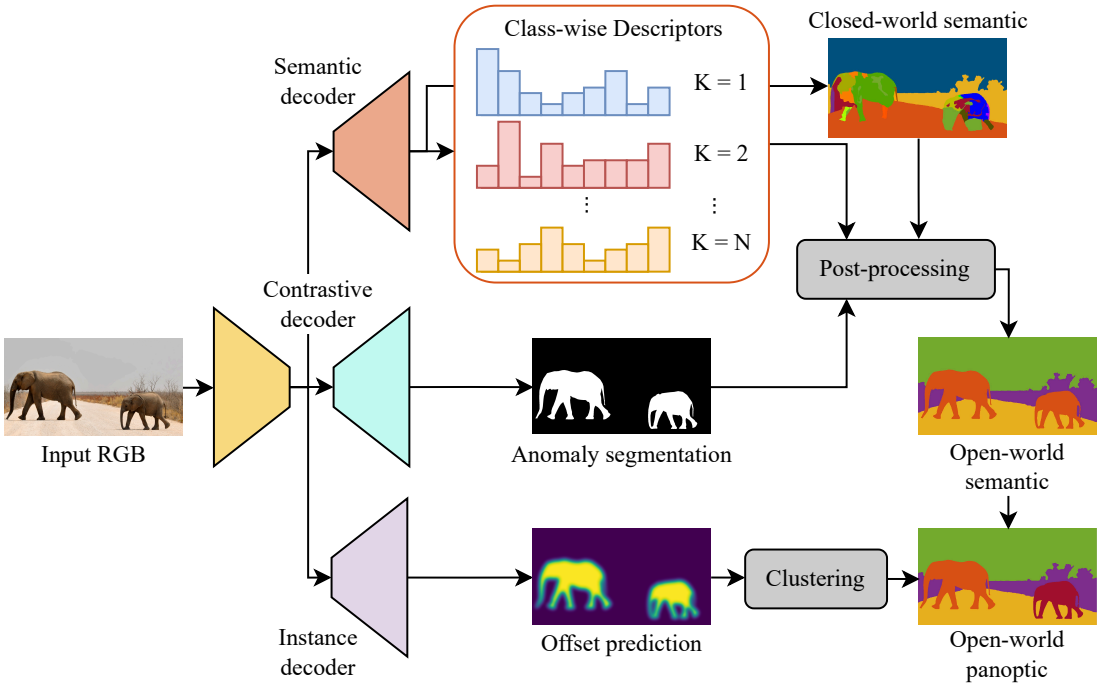


Figure 9.2: Our network processes an RGB image via an encoder and three decoders, for semantic segmentation, anomaly segmentation and instance segmentation. The semantic segmentation decoder also builds class descriptors for the known categories. Results are post-processed and yield the final open-world panoptic segmentation result.

open-world segmentation tasks discussed in Chapter 6. Our method not only surpasses the open-world semantic segmentation performance of our previous approach but also generalizes effectively on a number of different domains, such as autonomous driving, indoor navigation, and underwater monitoring. Importantly, Con2MAV achieves these results while maintaining strong closed-world performance, making it suitable for real-world deployment.

9.1 Our Approach

In this chapter, we tackle the problem of open-world panoptic segmentation. We propose an approach, shown in Figure 9.2, based on a fully-convolutional neural network with one encoder and three decoders. Our approach builds on top of our work for open-world semantic segmentation presented in Chapter 8.

9.1.1 Encoder Architecture

We use a lightweight fully-convolutional neural network to facilitate future deployment in the real world. However, our ideas and contributions can be applied to any kind of network. Our CNN is composed of one encoder and three de-

coders. We use a ResNet34 [72] encoder, where we replaced the standard ResNet block with the NonBottleneck-1D block [166]. This choice is motivated solely by the fact that we aim to make our architecture more lightweight, and the NonBottleneck-1D block replaces all 3×3 convolutions by a sequence of 3×1 and 1×3 convolutions with a ReLu in between. Additionally, we incorporate contextual information by means of a pyramid pooling module [229] at the end of the encoding part.

9.1.2 Decoder Architectures

Our three decoders have the same structure and are composed of three SwiftNet modules [148] with NonBottleneck-1D blocks, and two final upsampling modules based on nearest-neighbor and depth-wise convolutions [34], that have the major advantage of substantially reducing the computation needed. Following the standard UNet [168] design, we also employ encoder-decoder skip connections after each downsampling stage of the encoder to propagate fine-grained features to the three decoders.

The first decoder, called "semantic decoder", targets semantic segmentation and additionally builds a class descriptor for each known category. At inference time, we compare the final features to the known descriptors to discriminate between known and unknown classes, building descriptors for unknown classes as well. The second decoder, called "contrastive decoder", targets anomaly segmentation by mapping the features of pixels belonging to known classes on a unit hypersphere, and the ones belonging to unknown classes to 0. The third decoder, called "instance decoder" targets class-agnostic instance segmentation by means of vector fields-inspired loss functions [205]. The outputs of the decoders go through different post-processing phases to obtain the final open-world panoptic segmentation result.

Consider an image $I \in \mathbb{R}^{H \times W}$. We refer to the set of pixels in the image as $\Omega = \{(1, 1), \ldots, (H, W)\}$. Additionally, we call $Y \in \{1, \ldots, K\}^{H \times W}$ the ground truth semantic mask of I, where K is the number of known categories used at training time. In contrast, we refer to the predicted (closed-world) semantic mask as $\hat{Y} \in \{1, \ldots, K\}^{H \times W}$. Similarly, we call Z the ground truth and $\hat{Z}$ the predicted instance mask, respectively. Finally, we denote with $\Omega_k = \{p \in \Omega \mid Y_p = k\}$ the set of pixels in the image whose ground truth label is k, and with $\hat{\Omega}_k = \{p \in \Omega \mid \hat{Y}_p = Y_p = k\}$ the set of pixels whose predicted label $\hat{Y}_p$ and ground truth label Y_p are both equal to k, i.e., the set of true positives for class k. We do not use bold notation for the pixel tuple p for the sake of clarity.

Semantic Decoder. The semantic decoder targets semantic segmentation,

and it leverages the standard weighted cross-entropy loss:

$$\mathcal{L}_{\text{sem}} = -\frac{1}{|\Omega|} \sum_{p \in \Omega} \omega_k \, \boldsymbol{t}_p^\top \, \log\big(\sigma(\boldsymbol{f}_p^s)\big), \tag{9.1}$$

where ω_k is a class-wise weight computed via the inverse frequency of each class in the dataset, $\boldsymbol{t} \in \mathbb{R}^{H \times W \times K}$ is the one-hot encoded ground truth annotation of the image, $\boldsymbol{t}_p \in \mathbb{R}^K$ is the one-hot encoded ground truth annotation at pixel location p, $\sigma(\cdot)$ denotes the softmax operation, and $\boldsymbol{f}_p^s$ denotes the pre-softmax feature predicted at pixel p, where the superscript s indicates the semantic decoder.

Our semantic decoder aims to learn a unique class descriptor, or class prototype, for each known class. In ContMAV [186], we used a task-specific loss on pre-softmax features $\boldsymbol{f}_p^s \in \mathbb{R}^K$, but with few known classes, this low-dimensional space limited the quality of the descriptor. To address this, we now apply the loss earlier, on the higher-dimensional pre-logit features $\boldsymbol{\ell}^s$, which are independent of the number of classes and have fixed dimension D. This change significantly improves performance in low-class-count scenarios, as we show in the experiments.

To build the class descriptors, during training, we accumulate the pre-logits of all true positives for each class, where a true positive is a pixel that is correctly segmented. By doing this, we can build a running average class descriptor $\boldsymbol{\mu}_k \in \mathbb{R}^D$ for each class $k \in \{1, \ldots, K\}$ that appears at training time:

$$\boldsymbol{\mu}_k = \frac{1}{|\hat{\Omega}_k|} \sum_{p \in \hat{\Omega}_k} \boldsymbol{\ell}_p^s. \tag{9.2}$$

Together with the mean, we also compute the per-class variance $\boldsymbol{\sigma}_k^2 \in \mathbb{R}^D$ via the sum of squares:

$$\boldsymbol{\sigma}_k^2 = \frac{1}{|\hat{\Omega}_k|} \sum_{p \in \hat{\Omega}_k} (\boldsymbol{\ell}_p^s - \boldsymbol{\mu}_k) \odot (\boldsymbol{\ell}_p^s - \boldsymbol{\mu}_k), \tag{9.3}$$

where $\odot$ indicates the element-wise product (Hadamard product).

At the beginning of epoch e, each category has a mean and a variance $\boldsymbol{\mu}_k^{e-1}$ and $\boldsymbol{\sigma}_k^{e-1}$, computed at the previous epoch $e-1$. The goal of having a unique class descriptor for each known class is that the prediction at each pixel belonging to that class must be equal to the descriptor. Thus, we use the feature loss function $\mathcal{L}_{\text{feat}}$ at the pre-logit level given by

$$\mathcal{L}_{\text{feat}} = \frac{1}{|\Omega|} \sum_{k=1}^{K} \sum_{p \in \Omega_k} \frac{\left|\left|\boldsymbol{\ell}_p^s - \boldsymbol{\mu}_k^{e-1}\right|\right|^2}{\left(\boldsymbol{\sigma}_k^{e-1}\right)^2}. \tag{9.4}$$

The idea is that true positives help build the class descriptors in Equation (9.2), which are then used as supervision in the next epoch. Thus, this loss is inactive in the first epoch and is updated continuously throughout training.

The semantic decoder is optimized with a weighted sum of the loss functions introduced above:

$$\mathcal{L}_{\text{sdec}} = w_1 \, \mathcal{L}_{\text{sem}} + w_2 \, \mathcal{L}_{\text{feat}}. \tag{9.5}$$

Contrastive Decoder. The contrastive decoder also operates at the pre-logit level and tackles anomaly segmentation by combining the contrastive loss [29] and the objectosphere loss [46]. To do so, we first compute the mean pre-logit $\bar{\boldsymbol{\ell}}_k^c \in \mathbb{R}^D$ for each class k in the current image from the pre-logit predicted by this decoder. We compute the contrastive loss $\mathcal{L}_{\text{cont}}$ such that $\bar{\boldsymbol{\ell}}_k^c$ approximates the normalized mean representation $\bar{\boldsymbol{\mu}}_k^{e-1}$ of the corresponding class in the previous epoch $\boldsymbol{\mu}_k^{e-1}$ and gets dissimilar from the other classes mean representation:

$$\mathcal{L}_{\text{cont}} = -\sum_{k=1}^{K} \log \frac{\exp\left(\bar{\boldsymbol{\ell}}_k^{c\,\top} \bar{\boldsymbol{\mu}}_k^{e-1}/\tau\right)}{\sum_{i=1}^{K} \exp\left(\bar{\boldsymbol{\ell}}_k^{c\,\top} \bar{\boldsymbol{\mu}}_i^{e-1}/\tau\right)}, \tag{9.6}$$

where τ is a temperature parameter. The goal is to have $\bar{\boldsymbol{\ell}}_k^c$ approximate the corresponding vector in the feature bank, i.e., the normalized class descriptors $\bar{\boldsymbol{\mu}}_k^{e-1}$. To achieve this, we employ the pre-logit feature predicted by this decoder at each pixel p in this loss function, namely $\boldsymbol{\ell}_p^c$, in order to have dimensionally-consistent vectors. The outcome of the contrastive loss is to scatter the class-specific pre-logits on the unit hypersphere. Additionally, we use the objectosphere loss function

$$\mathcal{L}_{\text{obj}} = \begin{cases} \max\left(1 - \left\|\boldsymbol{\ell}_p^c\right\|^2, 0\right) & \text{, if } p \in \Omega_k \\ \left\|\boldsymbol{\ell}_p^c\right\|^2 & \text{, otherwise} \end{cases}, \tag{9.7}$$

where Ω_k is the set of pixels in the image belonging to known classes, while the others belong to unlabeled (void) areas of the image and are treated as unknown. The loss pushes known pre-logits to have norm ≥ 1 and unknowns to have norm 0. Although other values could be used, norm 1 aligns with the contrastive loss in Equation (9.6), which maps features to the unit sphere. Using a different norm would conflict with this goal, especially values larger than 1.

We optimize the contrastive decoder with a weighted sum of the loss functions introduced above:

$$\mathcal{L}_{\text{cdec}} = w_3 \, \mathcal{L}_{\text{cont}} + w_4 \, \mathcal{L}_{\text{obj}}. \tag{9.8}$$

Instance Decoder. The instance decoder addresses class-agnostic instance segmentation, where we predict a 2D offset for each pixel, so that pixels belonging to the same object instance point to the same area in the image and can be clustered together.

Given an instance $\mathcal{C}_j \subset \Omega$, defined as the set of pixels of the images belonging to the j-th object instance, with a centroid $\boldsymbol{c}_j$, we want to obtain offset vectors for all pixels belonging to $\mathcal{C}_j$ that point toward $\boldsymbol{c}_j$, while the remaining offset vectors

should not. We achieve this by means of the Lovász Hinge loss [144]

$$\mathcal{L}_{\text{off}} = \frac{1}{|\mathcal{C}|} \sum_{j=1}^{|\mathcal{C}|} \text{Lovasz}(\boldsymbol{F}_{\mathcal{C}_j}, \boldsymbol{G}_{\mathcal{C}_j}). \tag{9.9}$$

In this equation, $\mathcal{C}$ is the set of instances in the image, $\boldsymbol{F}_{\mathcal{C}_j} \in \mathbb{R}^{H \times W}$ is a soft-mask obtained by the offset predictions, while $\boldsymbol{G}_{\mathcal{C}_j} \in \{0, 1\}^{H \times W}$ denotes the binary ground truth mask of the j-th instances. The soft mask $\boldsymbol{F}_{\mathcal{C}_j}$ for instance $\mathcal{C}_j$ is obtained from the offset prediction: each pixel gets a score that depends on how far from the instance centroid its offset points to. This is formalized as

$$f_{\mathcal{C}_j} = \exp\left(- \frac{||\boldsymbol{e}_p - \boldsymbol{c}_j||^2}{2\eta^2} \right), \tag{9.10}$$

where $\boldsymbol{e}_p$ indicates the pixel location pointed by the prediction at pixel p, and η is a hyperparameter that defines an isotropic clustering region around the centroid. This loss function encourages the network to predict for all pixels associated with a specific instance, an offset vector that points towards its corresponding centroid. Simultaneously, it penalizes offset vectors that point towards a different centroid than their own. This forms distinct clusters in 2D space, even if the centroid lies outside the instance.

Following the ideas formulated by Weyler *et al.* [205], we adopt divergence and curl loss functions to refine offset predictions. These two loss functions are inspired by the concept of vector field in physics. Divergence captures how pixels flow toward or away from a point, aligning well with the idea of pixels pointing to a centroid. Curl measures rotation in the field, helping to regularize the vector pattern. For formalizing the divergence and curl loss functions, we need a simplified vector field defined by a continuous multivariable function $\boldsymbol{O}(h, w) : \mathbb{R}^2 \to \mathbb{R}^2$ given by:

$$\boldsymbol{O}(h, w) = \begin{bmatrix} o_h(h, w) \\ o_w(h, w) \end{bmatrix} = \begin{bmatrix} -h + c_h \\ -w + c_w \end{bmatrix}, \tag{9.11}$$

that behaves as a perfect sink, i.e., all offset vectors point toward $\boldsymbol{c} = \begin{bmatrix} c_h, c_w \end{bmatrix}^\top$. In 2D, divergence is defined as

$$\text{div}\, \boldsymbol{O} := \frac{\partial o_h(h, w)}{\partial h} + \frac{\partial o_w(h, w)}{\partial w}. \tag{9.12}$$

Thus $\frac{\partial o_h(h,w)}{\partial h} = \frac{\partial o_w(h,w)}{\partial w} = -1$ and $\text{div}\, \boldsymbol{O} = -2$. The divergence should then be equal to -2, and additionally, its two components must be equal. This is achieved by two regression loss functions:

$$\begin{aligned} \mathcal{L}_{\text{div}} &= \frac{1}{|\Omega|} \sum_{p \in \Omega} \rho\left((\text{div}\, \boldsymbol{O})_p - (-2) \right), \\ \mathcal{L}_{\text{div}}^{\text{aux}} &= \frac{1}{|\Omega|} \sum_{(h,w) \in \Omega} \rho\left(\frac{\partial o_h(h, w)}{\partial h} - \frac{\partial o_w(h, w)}{\partial w} \right), \end{aligned} \tag{9.13}$$

where $\rho(\cdot)$ denotes the Geman-McClure loss [7], and (h, w) denote the image coordinates of pixel p.

The curl is defined as

$$\operatorname{curl} \boldsymbol{O} := \frac{\partial o_h\,(h,w)}{\partial w} - \frac{\partial o_w\,(h,w)}{\partial h}. \tag{9.14}$$

When applying the curl to the vector field in Equation (9.11), we obtain

$$\frac{\partial o_h\,(h,w)}{\partial w} = \frac{\partial o_w\,(h,w)}{\partial h} = 0, \tag{9.15}$$

which implies $\operatorname{curl} \boldsymbol{O} = 0$. Simply put, this means we want no rotational behavior in our vector field. We formalize two loss functions to have the curl equal to zero and the two partial derivatives to be equal to each other:

$$\begin{aligned} \mathcal{L}_{\text{curl}} &= \frac{1}{|\Omega|} \sum_{p \in \Omega} \rho \left((\operatorname{curl} \boldsymbol{O})_p \right) \\ \mathcal{L}^{\text{aux}}_{\text{curl}} &= \frac{1}{|\Omega|} \sum_{(h,w) \in \Omega} \rho \left(\frac{\partial o_h\,(h,w)}{\partial w} - \frac{\partial o_w\,(h,w)}{\partial h} \right). \end{aligned} \tag{9.16}$$

For further details, we refer to the work by Weyler *et al.* [205]. The instance decoder is optimized with the weighted sum of all losses introduced so far

$$\mathcal{L}_{\text{ins}} = w_5\,\mathcal{L}_{\text{off}} + w_6\,\mathcal{L}_{\text{div}} + w_7\,\mathcal{L}^{\text{aux}}_{\text{div}} + w_8\,\mathcal{L}_{\text{curl}} + w_9\,\mathcal{L}^{\text{aux}}_{\text{curl}}. \tag{9.17}$$

9.1.3 Post-Processing

Given the outputs of the three decoders, we now describe the generation of the corresponding masks.

Post-Processing for Anomaly Segmentation. We obtain anomaly segmentation results by fusing the outputs of the semantic and contrastive decoders. The semantic decoder provides class descriptors (mean $\boldsymbol{\mu}_k$ and variance $\boldsymbol{\sigma}_k^2$) in the pre-logit space for each known class, allowing us to construct a multi-variate normal distribution $\mathcal{N}_k(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, where $\boldsymbol{\Sigma}_k = \operatorname{diag}(\boldsymbol{\sigma}^2)$ is the covariance matrix, which we obtain from the variance assuming that all classes are independent. In this way, for each predicted pre-logit $\boldsymbol{\ell}_p$ at pixel p, we can compute its fitting score to each known category by means of the squared exponential kernel

$$s_k(\boldsymbol{\ell}_p) = \exp\left(-\frac{1}{2} (\boldsymbol{\ell}^s_p - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{\ell}^s_p - \boldsymbol{\mu}_k) \right). \tag{9.18}$$

Then, we take the highest score

$$s(p) = \max_k s_k(\boldsymbol{\ell}^s_p), \tag{9.19}$$

which corresponds to the closest known category to the predicted pixel. At test time, this process replaces the standard procedure for closed-world semantic segmentation (i.e., argmax(·) on pre-softmax outputs). To decide if a pixel is known or unknown, we use the $1\boldsymbol{\sigma}$ bound: if the pre-logit is within 1 standard deviation from the class mean, it is considered known; otherwise, it is marked as unknown. Each pixel gets a binary score $s^{\text{sem}}_{\text{unk},\,p} \in \{0,\,1\}$, indicating whether it is unknown or not.

The contrastive decoder aims to have all features of pixels belonging to known classes with norm close to 1, and features of pixels belonging to unknown classes with norm close to 0. Assuming unknowns follow a normal distribution $\mathcal{N}(0,\,1)$, known class features become outliers. We use the $1\boldsymbol{\sigma}$ bound to get $s^{\text{con}}_{\text{unk},\,p} \in \{0,\,1\}$, a boolean variable indicating whether the pixel is unknown or not.

Finally, we consider a pixel to be unknown only if both conditions hold, namely $s^{\text{sem}}_{\text{unk},\,p} = 1$ and $s^{\text{con}}_{\text{unk},\,p} = 1$.

Post-Processing for Open-World Semantic Segmentation. At test time, we have K pre-logit descriptors $\boldsymbol{\ell}^s_k$ for the known classes. For each pixel, we assign the closest category with Equation (9.19). As described in the open-world semantic segmentation task, we want our discovered classes to be consistent in the dataset. When a new anomalous pixel is detected, its descriptor $\boldsymbol{\ell}^s_p \in \mathbb{R}^D$ is saved with the existing ones. Basically, we treat our newly-discovered category as a "new known" class. Our database of descriptors is composed of $\bar{K} = K + K_{\text{unk}}$ descriptors, where K_{unk} is the number of unknown classes discovered so far. While pre-logit descriptors for known classes are fixed after training, those for discovered unknowns are updated online using a running mean and variance, similar to training but without ground truth annotation.

Post-Processing for Open-World Panoptic Segmentation. We obtain instance predictions by clustering the offsets with HDBScan [131]. We execute the clustering only in the "thing" areas, which means all the known "thing" classes, and all the unknown categories. To ensure consistency, we filter the clustering results using semantic predictions, preventing pixels from different semantic classes from being grouped into the same instance. In essence, we prioritize semantic consistency over instance grouping.

9.2 Experimental Evaluation

The main focus of this work is an approach, Con2MAV, for open-world panoptic segmentation. We present extensive experiments on multiple datasets to show the capabilities of our method. The results of our experiments show that our model achieves state-of-the-art results for all open-world tasks described in Chapter 6, while performing competitively on the known classes. Additionally, we show how

Table 9.1: Results comparison between our model trained in a closed- and open-world fashion on the different datasets we use. Metrics are computed only on known classes, to show that the open-world modality does not significantly harm closed-world performance. In COCO, K indicates the percentage of semantic classes discarded from the training set to form the open-world test set (more details in Section 9.2.1). All metrics are reported as percentages (%).

Dataset	Modality	mIoU	PQ
Cityscapes	Closed-world	71.1	-
	Open-world	68.3	-
COCO	Closed-world	71.9	43.8
	Open-world (K = 5%)	70.0	43.6
	Open-world (K = 10%)	68.5	39.8
	Open-world (K = 20%)	65.6	39.1
BDDAnomaly	Closed-world	64.6	-
	Open-world	62.4	-
SUIM	Closed-world	68.6	-
	Open-world	60.0	-

our new dataset, PANIC, enables various kinds of open-world segmentation tasks while being extremely challenging.

9.2.1 Experimental Setup

Dataset and metrics. We use multiple datasets for validating our method, using the metrics discussed in the previous section. For anomaly segmentation, we use SegmentMeIfYouCan [27] and submit our prediction to the public challenge, BDDAnomaly, and our dataset PANIC described in Chapter 7. For open-world semantic segmentation, we use BDDAnomaly [73], COCO [106], SUIM [80], and PANIC. BDDAnomaly samples from BDD100K and removes all images containing bicycles, motorcycles, and trains from the training set, to create an open-world test set. Similarly, Mask2Anomaly [161] proposes three open-world splits of COCO, in which 5%, 10%, and 20% of ground truth categories are discarded from the training set to create an open-world test set, respectively. Classes are removed cumulatively. For the 5% they remove "car", "cow", "pizza", "toilet". For the 10%, they additionally remove "boat", "tie", "zebra", "stop sign". Finally, for the 20%, they also remove "dining table", "banana", "bicycle", "cake", "sink", "cat", "keyboard", and "bear". U3HS [55] and Prior2Former [175] remove other classes from COCO to generate a test set for open-set and open-world panoptic segmentation, respectively, but their code is not publicly available. Thus, our

Table 9.2: Anomaly segmentation results on the SegmentMeIfYouCan test set. We report only methods that do not use external data, i.e. out of distribution (OoD) data wih semantic labels different from the ones in Cityscapes, during training. Best results in bold. All metrics are reported as percentages (%).

Approach	Pixel-Level		Component-Level		
	AUPR	FPR95	sIoU gt	PPV	mean F1
Maskomaly [1]	93.4	6.9	55.4	51.2	49.9
RbA [141]	86.1	15.9	56.3	41.4	42.0
ContMAV [186]	90.2	3.8	54.5	61.9	63.6
UNO [43]	**96.1**	**2.3**	**68.0**	51.9	58.9
Con2MAV (ours)	90.0	2.7	59.1	**68.3**	**69.4**

generation of the COCO open-world test set follows Mask2Anomaly. Similarly, for the underwater dataset SUIM, we discarded all images containing annotations of "humans", "robots", and "wrecks & ruins". For open-set panoptic segmentation, we use COCO and PANIC. Finally, for open-world panoptic segmentation, we use PANIC.

Training details and parameters. In all experiments, we use the one-cycle learning rate policy [181] starting from 0.004. We perform random scale, crop, and flip data augmentations, and optimize with Adam [86] for 200 epochs with batch size 16. We set loss weights $w_1 = 0.8$, $w_2 = 0.2$, $w_3 = 0.5$, $w_4 = 0.5$, $w_5 = 0.4$, $w_6 = 0.2$, $w_7 = 0.1$, $w_8 = 0.2$, and $w_9 = 0.1$, to have losses in the same order of magnitude. We set $\tau = 0.1$, following Chen *et al.* [29]. For SegmentMeIfYouCan and PANIC, we train on Cityscapes. For BDDAnomaly, COCO, and SUIM, we train on their own training sets.

9.2.2 Closed-World Performance

While tailored for open-world segmentation, our approach maintains strong performance on closed-world categories. As shown in Table 9.1, the mIoU and PQ scores on known classes remain comparable when comparing the full open-world model with its closed-world variant, where open-world components are removed. This confirms that our design does not compromise closed-world performance.

9.2.3 Anomaly Segmentation

This set of experiments shows that our approach achieves state-of-the-art results on anomaly segmentation. We report results on SegmentMeIfYouCan in Table 9.2, PANIC in Table 9.3, and BDDAnomaly in Table 9.4. We achieve com-

Table 9.3: Anomaly segmentation results on the hidden test set of our dataset, PANIC. Best results in bold. All metrics are reported as percentages (%). Public competition at `https://www.codabench.org/competitions/4561`.

Approach	Pixel-Level		Component-Level		
	AUPR	FPR95	sIoU gt	PPV	mean F1
ContMAV [186]	91.7	66.4	15.0	**72.1**	24.2
Con2MAV (ours)	**95.7**	**35.3**	**20.9**	64.7	**31.2**

Table 9.4: Anomaly segmentation results on BDDAnomaly. Best results in bold. All metrics are reported as percentages (%).

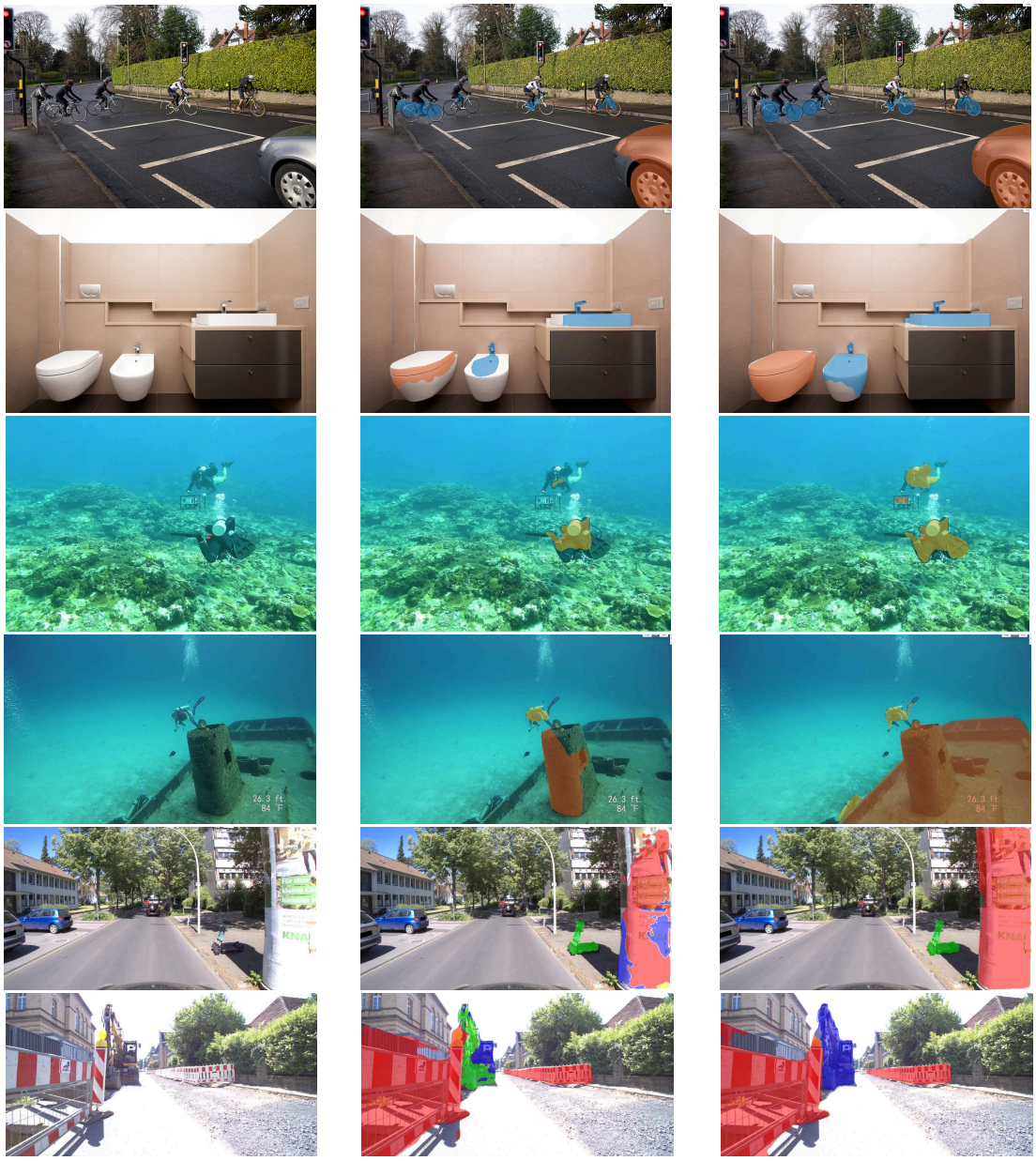
Approach	AUPR	FPR95
MaxSoftmax [75]	3.7	24.5
MC Dropout [52]	4.3	16.6
Confidence [45]	3.9	24.5
MaxLogit [73]	5.4	14.0
ContMAV [186]	96.1	6.9
Con2MAV (ours)	**97.9**	**5.8**

pelling results on all three datasets, consistently outperforming the baselines. Additionally, we outperform our previous approach, ContMAV, on all datasets. While gains over ContMAV are moderate on SegmentMeIfYouCan and BDDAnomaly, Con2MAV shows a clear advantage on PANIC. Notably, on SegmentMeIfYouCan, we rank top 1 in two component-level metrics, outperforming even methods trained with out-of-distribution data.

The performance of ContMAV and Con2MAV on PANIC highlight its difficulty, even for anomaly segmentation. Compared to SegmentMeIfYouCan, results drop notably, though both methods still achieve good AUPR and PPV. Intuitively, AUPR is high when there is a sharp difference in the probability heatmap between unknown and known areas, while PPV is high when there are few false positive predictions. High scores mean that both approaches act quite conservatively on PANIC, being very confident in the prediction of anomalous areas, despite having false negatives. This is a further measure of how challenging our dataset is compared to others.

9.2.4 Open-World Semantic Segmentation

The second set of experiments is about open-world semantic segmentation. Since the task is uncommon in the literature, we compare against our previous ap-



(a) Input RGB (b) ContMAV prediction (c) Con2MAV prediction

Figure 9.3: Qualitative results on open-world semantic segmentation. We show the input RGB image on the left, the prediction of our previous approach ContMAV [186] in the middle, and the prediction of our current approach Con2MAV on the right. The top two images belong to COCO [106], the middle two images belong to SUIM [80], and the bottom two images belong to our dataset PANIC.

proach, ContMAV [186]. We include a lower-bound baseline that treats unknowns as background and applies K-means clustering in the background feature space to identify unknown classes. We report results on COCO in Table 9.5, BDDAnomaly in Table 9.6, SIUM in Table 9.7, and PANIC in Table 9.8. On COCO, we consistently outperform ContMAV by at least 10% in the mIoU computation. On

Table 9.5: Open-world semantic segmentation results on the COCO validation set on three different known-unknown splits. K denotes the percentage of unknown classes present in the dataset. Best results in bold. All metrics are reported as percentages (%).

K %	**Approach**	**mIoU**$_u$
5	Background + cluster	8.4
	ContMAV (no feat loss) [186]	23.5
	ContMAV (with feat loss) [186]	40.0
	Con2MAV (ours)	**50.5**
10	Background + cluster	6.1
	ContMAV (no feat loss) [186]	18.8
	ContMAV (with feat loss) [186]	38.5
	Con2MAV (ours)	**48.5**
20	Background + cluster	1.2
	ContMAV (no feat loss) [186]	15.0
	ContMAV (with feat loss) [186]	33.4
	Con2MAV (ours)	**48.5**

Table 9.6: Open-world semantic segmentation results on BDDAnomaly. Best results in bold. All metrics are reported as percentages (%).

Approach	**IoU**$_u$			**mIoU**$_u$
	Train	Motorcycle	Bicycle	
Background + cluster	0	32.3	32.8	21.7
ContMAV [186]	62.4	62.2	**56.8**	60.5
Con2MAV (ours)	**66.5**	**64.4**	53.8	**61.6**
Closed-world	72.3	69.3	60.9	67.5

BDDAnomaly, we outperform ContMAV in two out of three unknown classes, while getting close to the performance of the closed-world model, reported as a performance upper bound. On PANIC, we improve ContMAV by almost 5% mIoU, despite our previous approach achieving a better completeness.

The use of the SUIM dataset is to test how Con2MAV compares to ContMAV when only few known training classes are available, which is one of the main limitations of our previous work. We reduce the number of training classes to just 4. Our goal is to validate our choice of building the class descriptor at pre-logit level, rather than at pre-softmax level as we did in ContMAV. The

Table 9.7: Open-world semantic segmentation results on the SUIM dataset. Best results in bold. All metrics are reported as percentages (%).

Approach	$\mathbf{IoU}_u$			$\mathbf{mIoU}_u$
	Human	Wrecks & Ruins	Robot	
ContMAV [186]	46.2	36.9	46.2	43.1
Con2MAV (ours)	**69.4**	**64.3**	**53.0**	**62.2**
Closed-world	82.0	71.6	80.2	77.9

Table 9.8: Open-world semantic segmentation results on the hidden test set of our dataset, PANIC. Best results in bold. All metrics are reported as percentages (%). Public competition is available at `https://www.codabench.org/competitions/4563`.

Approach	Unknown Classes		
	mIoU_u	Completeness	Homogeneity
ContMAV [186]	15.8	**81.4**	77.3
Con2MAV (ours)	**20.2**	77.6	**78.3**

results in Table 9.7 show that our design choice successfully addressed this issue, as we outperform ContMAV by 19% mIoU, and up to 28% in a single individual category ("wreck & ruins"). We show qualitative results for this task in Figure 9.3.

9.2.5 Open-Set Panoptic Segmentation

Here, we present experiments on open-set panoptic segmentation on COCO and PANIC. On COCO, we compare with other state-of-the-art approaches and report results in Table 9.9. Void-train is a baseline that leverages an instance segmentation head and a dedicated category class for the unknown (i.e., the void). EOPSN [79] identifies novel classes based on exemplars. Mask2Anomaly [161] is a mask-based method that showed state-of-the-art results in many open-world segmentation tasks. Our approach, Con2MAV, consistently outperform all baselines in all metrics. Additionally, while the baselines drop in performance when increasing the open-world set dimension by reducing the training data available (i.e., when K grows, meaning that more categories, and thus more images, are removed from the training set to join the open-world test set), our performance remains similar. We report results on PANIC in Table 9.10. It is interesting to note that our performance on PANIC is similar to what we achieve on COCO, especially in the more challenging splits. This is further indication of how challenging our dataset is. We show qualitative results in Figure 9.4.

Table 9.9: Open-set panoptic segmentation results on the COCO validation set on three different known-unknown splits. K denotes the percentage of unknown classes present in the dataset. Best results in bold. All metrics are reported as percentages (%).

K %	**Approach**	**Unknown Classes**		
		PQ_u	SQ_u	RQ_u
5	Void-train	8.6	72.7	11.8
	EOPSN [79]	23.1	74.7	30.9
	Mask2Anomaly [161]	24.3	78.2	32.1
	Con2MAV (ours)	**25.1**	**78.4**	**34.6**
10	Void-train	8.1	72.6	11.2
	EOPSN [79]	17.9	76.8	23.3
	Mask2Anomaly [161]	19.7	77.0	25.7
	Con2MAV (ours)	**21.3**	**77.1**	**27.6**
20	Void-train	7.5	72.9	10.3
	EOPSN [79]	11.3	73.8	15.3
	Mask2Anomaly [161]	14.6	76.2	19.1
	Con2MAV (ours)	**22.1**	**88.5**	**24.9**

Table 9.10: Open-set panoptic segmentation results on the hidden test set of our dataset, PANIC. All metrics are reported as percentages (%). Public competition is available at `https://www.codabench.org/competitions/4562`.

Approach	**Unknown Classes**		
	PQ_u	SQ_u	RQ_u
Con2MAV (ours)	21.6	72.4	28.4

Table 9.11: Open-world panoptic segmentation results on the hidden test set of our dataset, PANIC. All metrics are reported as percentages (%). Public competition is available at `https://www.codabench.org/competitions/4477`.

Approach	**Unknown Classes**			
	PQ_u	$mIoU_u$	Completeness	Homogeneity
Con2MAV (ours)	25.1	20.2	77.6	78.3

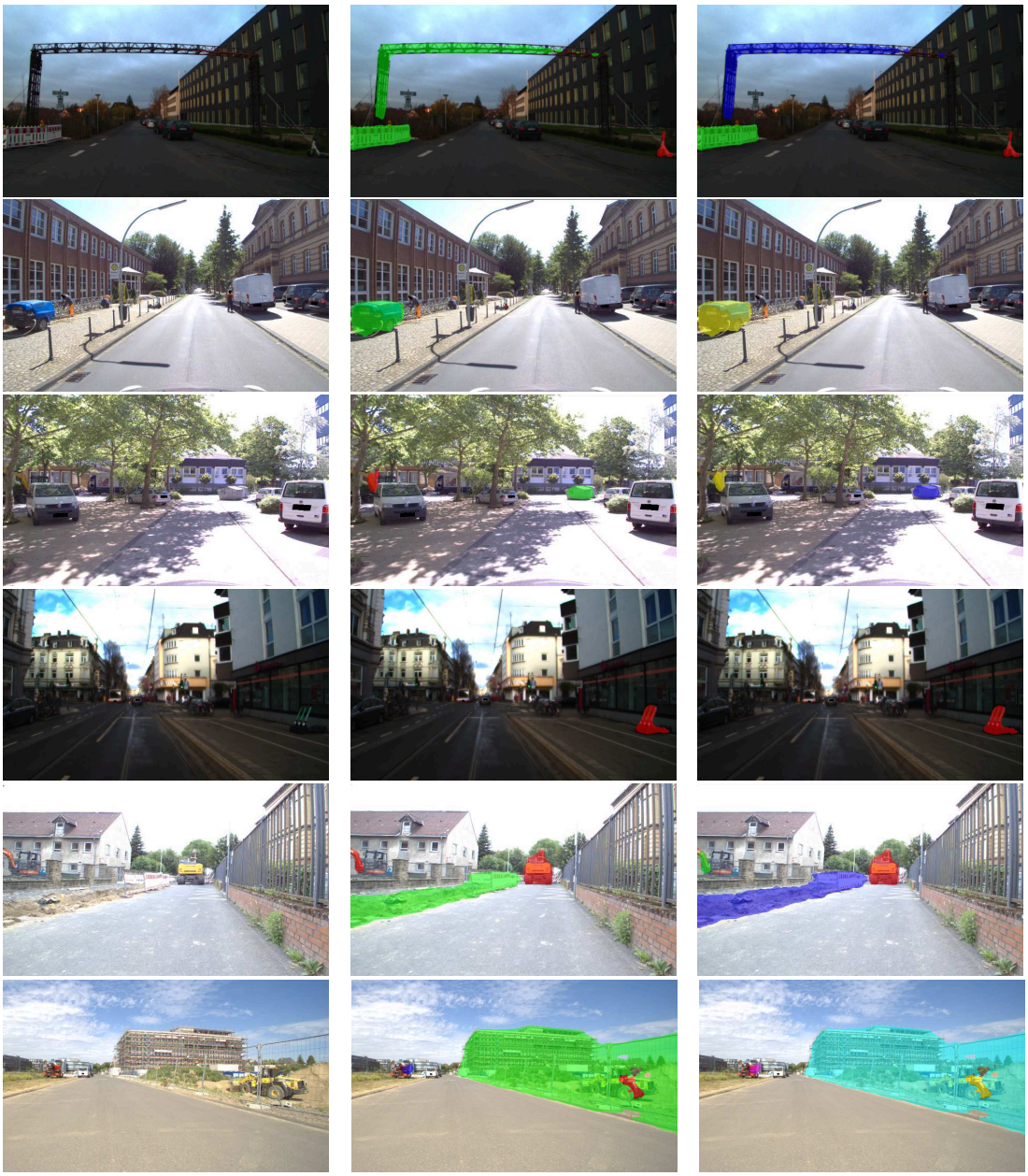
9.2.6 Open-World Panoptic Segmentation

The last experiment shows our performance on open-world panoptic segmentation on PANIC. The results are reported in Table 9.11. For this novel task there is no

(a) Input RGB (b) Semantic prediction (c) Instance prediction

Figure 9.4: Qualitative results on open-set panoptic segmentation. We show the input RGB image on the left, the anomaly prediction of Con2MAV in the middle, and the instance prediction of Con2MAV on the right. The top three images belong to COCO [106], the bottom three images belong to our dataset PANIC. In the anomaly prediction, we overlayed the prediction mask over the RGB image in red. In the instance prediction, we overlayed the prediction mask over the RGB image in different colors, where different colors indicate different predicted instances.

baseline to compare to, as the only existing approach does not have code available [175]. Our approach still achieves a satisfactory 24.3% panoptic quality, and



(a) Input RGB (b) Semantic prediction (c) Instance prediction

Figure 9.5: Qualitative results on open-world panoptic segmentation. We show the input RGB image on the left, the open-world semantic prediction of Con2MAV in the middle, and the instance prediction of Con2MAV on the right. All six images belong to our dataset PANIC. In the semantic prediction, we overlayed the prediction mask over the RGB image in red, where different colors indicate different predicted categories. In the instance prediction, we overlayed the prediction mask over the RGB image in different colors, where different colors indicate different predicted instances.

especially has good performance when it comes to completeness and homogeneity. This means that the categories we predict do not span multiple ground truth classes but stay close to a single one, and that our ground truth categories are

Table 9.12: Ablation studies. (a) Decoder components. (b) Instance-related loss functions. Last rows correspond to Con2MAV. Best results in bold. All metrics are reported as percentages (%).

(a)

	$\boldsymbol{D}_C$	PL	LTh	Unk. Classes	
				mIoU_u	PQ_u
A	✓			50.3	14.4
B		✓		45.5	10.2
C			✓	42.3	8.3
D		✓	✓	48.2	11.9
E	✓	✓		54.7	19.2
F	✓		✓	50.2	15.8
G	✓	✓	✓	**57.3**	**22.4**

(b)

	Div.	Curl	Unk. Classes	
			mIoU_u	PQ_u
H			57.1	12.4
I	✓		56.7	17.8
J		✓	56.9	15.0
K	✓	✓	**57.3**	**22.4**

not predicted as many categories at once, but the prediction is quite consistent. We show qualitative results in Figure 9.5.

9.2.7 Ablation Studies

Finally, we provide ablation studies to investigate the individual contribution of our modules. We perform all ablation studies on BDDAnomaly since we have access to test set ground truth labels. Additionally, we used BDDAnomaly also for the ablation studies we presented in the previous chapter for open-world semantic segmentation, so we do the same to facilitate the comparison.

The first ablation study shows the contribution of (i) the contrastive decoder (called $\boldsymbol{D}_C$ in the table), (ii) the pre-logit instead of the pre-softmax features for computing the loss functions of the semantic and contrastive decoder (called PL), and (iii) the impact of having auto-tuned thresholds in the post-processing mechanisms (called LTh). The approach without both pre-logit and auto-tuned thresholds corresponds to our previous approach, ContMAV. Results are shown in Table 9.12a. The contrastive decoder plays a key role in achieving strong performance by complementing the semantic decoder and reinforcing its predictions (see entries A vs. B, C, D in the table). As observed in our previous work [186] as well, it acts as a "stabilizer" for open-world semantic segmentation, benefiting from its simpler objective, anomaly segmentation, which makes it more robust to noise and errors. Notice how, without using the pre-logit, using the learned thresholds achieves extremely similar performance to not using them (A vs. F). This result is expected, as in the approach we proposed in the previous chapter,

Table 9.13: Architectural efficiency reported in terms for number of floating point operations (GFLOPs) and number of trainable parameters of the network.

Approach	GFLOPs	Params
Maskomaly [75]	937	215M
Mask2Anomaly [18]	258	**23M**
ContMAV [186]	84	48M
Con2MAV (ours)	**50**	65M

we had an intensive phase of hyperparameter tuning to achieve optimal performance. In this case, without any tuning, we managed to get similar performance for both mIoU and panoptic quality. The use of pre-logit instead of pre-softmax boosts performance by $4-5\%$ on both metrics (A vs. E). The learned thresholds bring a further improvement when paired with the pre-logit, as likely the class descriptors are more robust and the post-processing manages to better separate the newly-discovered classes (G).

The second ablation study evaluates the impact of our vector field-inspired loss function on the instance segmentation. We compare our full model with one lacking both divergence and curl (H in the table), or using only one (I and J). Results are shown in Table 9.12b. The mIoU is always very similar, because these loss functions do not have any impact on the semantic segmentation. The small differences are due to the fact that they all affect the shared encoder through backpropagation. Both divergence and curl individually improve the basic model. Additionally, the experiment suggests that divergence is more useful than curl for instance segmentation. Combining the two proves to be the best possible combination (K).

9.2.8 Architectural Efficiency

As pointed out in Section 9.1, we designed our neural network in order to be lightweight. In Table 9.13, we report the number of parameters and the GFLOPs of our model together with three state-of-the-art models with code available from the SegmentMeIfYouCan public benchmark. We show that our architecture is competitive and performs well in terms of efficiency.

9.3 What Could This Enable?

Open-world panoptic segmentation shares similarities with open-vocabulary segmentation, which aims to segment novel classes and objects by leveraging large

language models for class discovery. However, open-vocabulary approaches typically require explicit category prompts to guide segmentation. When we evaluated such methods [238] on PANIC, performance was poor unless the target categories were clearly specified. This, however, defeats the purpose of identifying unknown objects autonomously, without any prior knowledge.

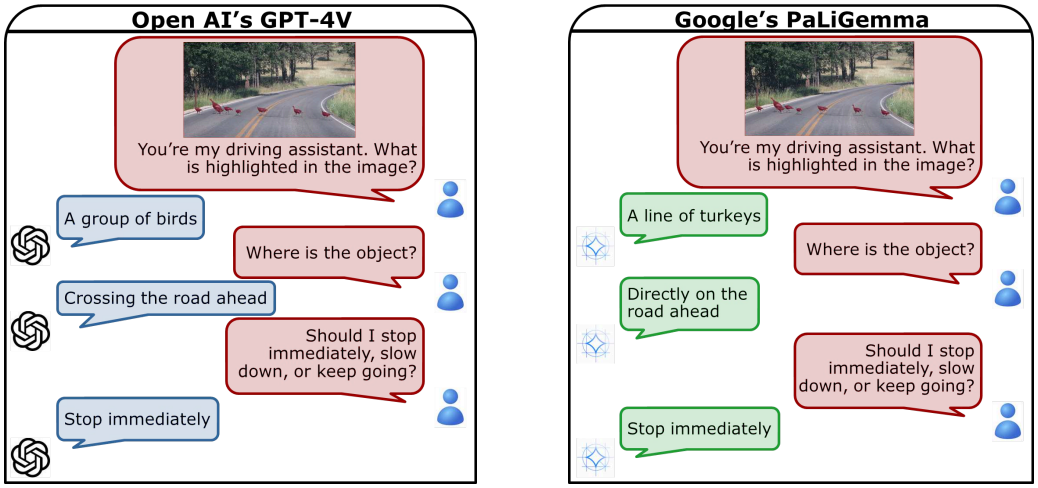
To address this limitation, we used a complementary strategy. We fed our open-world panoptic segmentation predictions to two vision-language models: GPT-4V [147] and PaliGemma [188]. We provided the input RGB image overlayed with our open-world prediction, and queried the vision-language models to identify the unknown objects and suggest appropriate actions for driving. Both models were prompted identically to assess their responses under the same conditions.

We show three exemplary results in Figure 9.6a, Figure 9.6b, and Figure 9.6c, organized as chat interactions for visualization. In Figure 9.6a, both models correctly identify birds crossing the road and propose appropriate behavioral suggestions. In Figure 9.6b, however, PaliGemma fails to recognize the electric scooters and entirely disregards the anomaly masks, focusing on the road ahead. Although its recommendation to slow down is generally reasonable in uncertain scenarios, the suggestion results from a perceptual oversight. The most interesting example is the one in Figure 9.6c, where both models fail to recognize the recumbent bicycle. However, GPT-4V shows good spatial awareness and recognizes that the object is entering the road on the left and suggests slowing down. PaliGemma, in contrast, says the object is standing ahead and suggests slowing down: this reveals not only a misunderstanding of the object's location, but also an inconsistent behavioral decision, as if there is an object standing ahead of a driving car, stopping would be the safer recommendation.

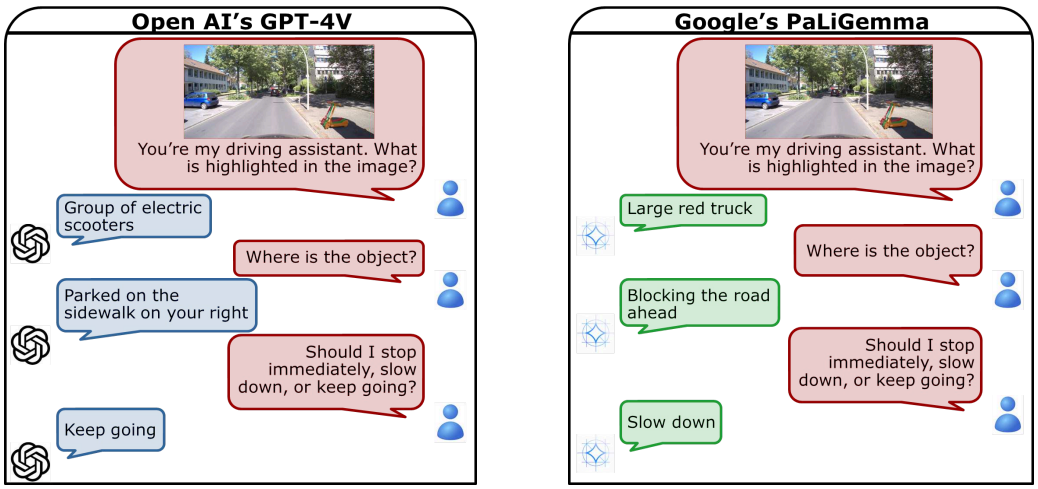
These results highlight an intriguing direction: models like ours, capable of detecting unknown instances without prior class information, can serve as a front-end to vision-language models for visual prompting, enabling contextual understanding and decision-making in complex, open-world environments.

9.4 Conclusion

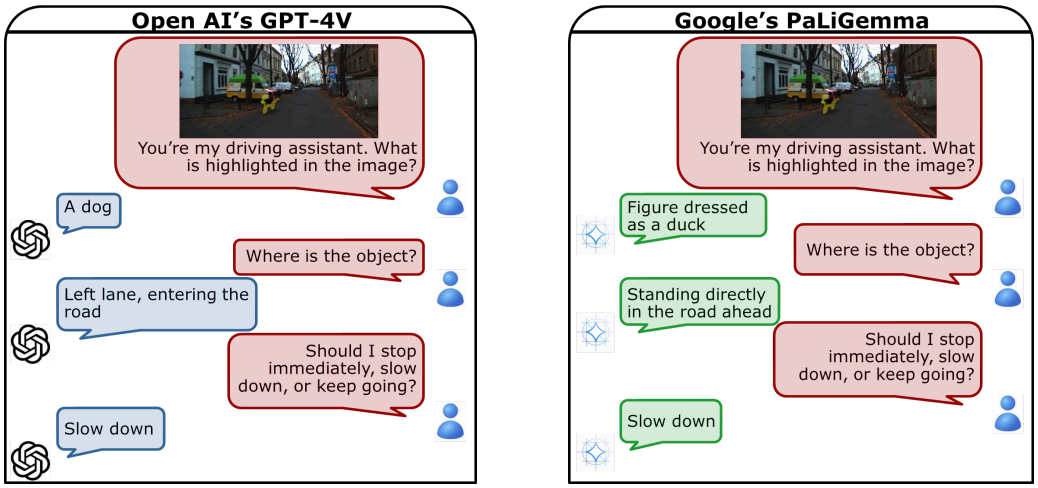
In this final chapter, we presented a novel approach for open-world panoptic segmentation, building upon the foundations introduced in Chapter 6. Our approach enables autonomous systems to move beyond the limitations of closed-world perception and achieve holistic scene understanding by segmenting both known and previously unseen categories at the semantic and instance levels. Our method takes as input a single RGB image and jointly produces a semantic segmentation mask and an instance segmentation mask for both known and unknown



(a)



(b)



(c)

Figure 9.6: GPT-4V and PaliGemma results on images from SegmentMeIfYouCan (a) and PANIC (b and c).

categories. In doing so, it achieves truly holistic scene understanding, extending open-world perception from the pixel level to the object level.

Through extensive quantitative experiments on multiple datasets, we demonstrated the effectiveness of our approach on Con2MAV across diverse domains, including autonomous driving, indoor environments, and underwater monitoring. The results consistently show strong performance across all open-world segmentation tasks, confirming the robustness and generality of our framework.

Beyond this, we highlighted how integrating vision-language models into open-world segmentation pipelines opens up a promising direction for future research. Such integration enables not only the discovery of arbitrary categories and objects, but also their subsequent naming through the vision-language model, all without supervision or explicit prompting. This stands in contrast to open-vocabulary segmentation, where predefined prompts or categories must be supplied. By combining segmentation with language understanding, we move closer to systems capable of autonomous, unsupervised, and semantically rich scene interpretation.

Chapter 10

Conclusion

Autonomous systems must be able to interpret and understand their surroundings in order to achieve true autonomy. Perception, and especially vision, is a core capability that is necessary for a number of applications such as navigation, interaction with the environment, monitoring, and more. Robots, unlike humans, do not possess any innate abilities to perceive and make sense of the world around them. In this sense, perception is one of the fundamental building blocks that contribute to making an engineered system into one that can act autonomously. Within visual perception, panoptic segmentation has emerged as a task of paramount importance. Its goal is to provide a holistic understanding of the environment by recognizing all different semantic categories in the scene, as well as all individual object instances. This unified formulation bridges the gap between semantic and instance segmentation, offering a complete and structured representation of the environment that is essential for autonomous systems to operate safely and effectively.

The main contribution of this thesis is a set of novel approaches for panoptic segmentation. Traditionally, this task has been formulated under the closed-world assumption, where the set of categories and objects of interest is fixed during training and remains unchanged at test time. Within the boundaries set by this underlying assumption, we first showed how to merge different sensor modalities while being robust to any of them missing. Then, we targeted the problem of nested instances, formulating the task of hierarchical panoptic segmentation in both, 2D and 3D. Finally, we got rid of the closed-world assumption and aimed to develop an algorithm for open-world panoptic segmentation, in order to be able to segment everything in the scene without being limited by pre-defined choices. Altogether, the approaches presented in this thesis advance the state of the art in panoptic segmentation and make a significant contribution to the broader field of visual perception, taking important steps toward more general, flexible, and reliable perception systems for real-world applications.

10.1 Summary of the Key Contributions to Panoptic Segmentation

The first contribution of this thesis is an approach for RGB-D panoptic segmentation. Our goal was to design a model capable of processing RGB and depth cues simultaneously, showing that depth enriches the information coming from the RGB, improving segmentation performance. To achieve this, we developed a novel fusion module that we used in our convolutional neural network. As a result, our network can handle not only RGB-D data, but also RGB-only and depth-only without the need to retrain. This flexibility is useful for robots equipped with different sensors, as well as situations in which one modality is not reliable.

Our second contribution is an approach for 2D hierarchical panoptic segmentation. Focusing on the agricultural domain, where more fine-grained information is often required than standard panoptic segmentation provides, we proposed a method that goes beyond segmenting crops as single instances. Specifically, we also segment their leaves, introducing a nested instance level that extends the standard task formulation. We achieve this by proposing a novel skip connection scheme that takes into account the underlying hierarchy among the different segmentation subtasks, thus using the skip connections to convey task-specific information from one decoder of the neural network to another.

The third contribution of this thesis builds directly on this idea by extending 2D hierarchical panoptic segmentation to the 3D domain. To address this, we release an annotated dataset of a real apple orchard, where each tree instance contains the tree trunk and its apples, and the second instance level separates the trunk and the individual apples. Our dataset is recorded with a variety of sensors, including TLS, RGB-D, and RGB cameras. Alongside the dataset, we also propose an approach to tackle this task, showing state-of-the-art performance even when compared to approaches limited to a single level of segmentation. Our approach, which employs the hierarchical skip connection scheme first introduced in our previous approach, shows compelling segmentation performance also when generalizing to the different sensors we employ in the dataset.

Moving beyond the classical closed-world assumption of panoptic segmentation, our fourth contribution is a public dataset that allows consistent and reproducible benchmarking for all open-world segmentation tasks. While such tasks are crucial for realistic scene understanding, they have been significantly less explored than the closed-world ones. A key motivation behind this dataset was to address the lack of benchmarks for open-world evaluation. We propose a thorough evaluation pipeline for four open-world segmentation tasks, namely anomaly segmentation, open-world semantic segmentation, open-set panoptic segmentation, and open-world panoptic segmentation, and release public Codabench competi-

tions for future research.

The fifth contribution of this thesis is an approach for open-world semantic segmentation. Here, we aim to segment semantic categories that did not appear during the training phase. We achieve this with a neural network with an encoder and two decoders, employing a specific loss function we designed for creating class-specific descriptors for all known categories. At test time, we use this ability of the network to either assign pixels to existing descriptors, corresponding to known categories, or create new ones, thus enabling novel class discovery. This allows us to discover a potentially unlimited number of novel categories.

Our sixth and last contribution is an approach for open-world panoptic segmentation. With this, we are able to discover not only novel categories but also individual objects. By removing the constraint of a fixed set of known classes, our approach enables the discovery of a virtually unlimited number of novel objects, moving closer to the goal of holistic scene understanding.

In summary, we have proposed a set of approaches and datasets that collectively push the boundaries of panoptic segmentation. Throughout this thesis, we have shown the effectiveness of our proposed approaches, often exceeding state-of-the-art performance, on real-world datasets, some of which we collected ourselves and shared with the research community. While this thesis does not address every challenge faced by robotic systems operating in the wild, the contributions presented here constitute key building blocks for advancing autonomous perception. By combining insights from computer vision and machine learning, this thesis strengthens the foundation for systems that must operate robustly and intelligently in real-world environments.

10.2 Future Work

The novel vision systems presented throughout this thesis show promising results for robotics perception across different tasks, sensors, and domains of applications. In the following, we want to discuss potential new research directions that could build upon our advances.

Our algorithm for RGB-D panoptic segmentation was designed to merge different modalities while being robust when one of them is missing. The goal was to demonstrate how geometry can enrich color information and improve segmentation. A natural extension of this work is to incorporate additional modalities, since our fusion module is not inherently tied to RGB or depth but can process arbitrary tensor inputs. Another attractive research direction consists of relaxing the closed-world assumption and moving towards open-world segmentation. In this context, depth could serve as a powerful prior, providing structural cues that may prove extremely useful for discovering novel objects and categories.

The work on 2D hierarchical panoptic segmentation is a valuable tool for obtaining fine-grained information about the crops in a field. One exciting research avenue is the integration of this approach with real in-field interventions and affordance detection, where robots must identify and act upon specific targets. Examples are weed removal, measuring, and cutting diseased leaves. Another important direction in agriculture is plant growth monitoring. Hierarchical segmentation can be integrated with a temporal-aware model to continuously track both plants and leaves over time, enabling detailed analysis of crop development.

Similarly, 3D hierarchical panoptic segmentation can be extended with temporal models for monitoring tree growth and with yield estimation techniques to precisely quantify production at the level of individual trees. Also in this case, the integration of affordance detection techniques with segmentation is a challenging research direction, supporting tasks such as autonomous fruit picking or branch pruning, where fine-grained scene understanding directly informs in-field actions.

Furthermore, an open-world formulation of hierarchical panoptic segmentation represents another compelling research direction. In agriculture, anomalies could manifest as previously unseen weeds that require removal, or as early signs of crop disease. Detecting and segmenting such unknown structures would extend the applicability of hierarchical segmentation, enabling not only fine-grained understanding of known categories, but also proactive identification of novel and potentially critical phenomena.

Our approach for open-world semantic segmentation allowed us to consistently discover novel semantic categories unseen during training, yet it also revealed certain limitations that constrained its broader applicability. In particular, the dimensionality of the feature vector proved restrictive when working with datasets containing only a few known categories, and the method did not account for object-level segmentation. These issues were successfully addressed in our subsequent work on open-world panoptic segmentation, which extended semantic discovery to the instance level.

Despite its strong results and wide applicability across domains, our open-world panoptic segmentation framework opens several promising directions for future research. Thus far, we have applied it only to RGB images. However, the method itself is not tied to RGB specifically, since it primarily relies on tailored loss functions and training strategies. Extending it to other modalities such as RGB-D frames or LiDAR point clouds would be a natural and valuable next step.

Two additional research avenues stem from the integration of vision-language models and object tracking. As qualitatively illustrated in Chapter 9, vision-language models provide contextual understanding and semantic grounding that go beyond what segmentation alone can offer, thereby supporting higher-level reasoning and decision-making. Object tracking, on the other hand, is a critical

requirement in domains such as autonomous driving, where agents must monitor all participants in the scene and anticipate their motion. Coupling open-world panoptic segmentation with tracking would provide richer temporal context and, when combined with vision-language models, could pave the way toward safer and more reliable navigation in real-world environments.

The discussion offered in this section is just a glimpse of the many challenges awaiting robots in perception and scene understanding. Yet, addressing these challenges brings us closer to realizing the timeless dream of autonomous machines, once confined to myth and science fiction, that can truly understand and interact with the world around them.

Bibliography

[1] J. Ackermann, C. Sakaridis, and F. Yu. Maskomaly: Zero-shot mask anomaly segmentation. In *Proc. of the British Machine Vision Conf. (BMVC)*, 2023.

[2] E.H. Adelson. On seeing stuff: the perception of materials by humans and machines. *Human Vision and Electronic Imaging (HVEI)*, 4299:1–12, 2001.

[3] Apollodorus. *Bibliotheca*. Romae: Apud B. Aegius, 1555.

[4] B. Arad, J. Balendonck, R. Barth, O. Ben-Shahar, Y. Edan, T. Hellström, J. Hemming, P. Kurtser, O. Ringdahl, T. Tielen, and B. van Tujil. Development of a sweet pepper harvesting robot. *Journal of Field Robotics (JFR)*, 37(6):1027–1039, 2020.

[5] D.H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.

[6] G. Bardak, M. Sodano, and M. Scholz. Integration of HD Maps and Point Clouds: An Efficient 3D Reconstruction Framework for Autonomous Driving Applications. *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 48:49–56, 2025.

[7] J.T. Barron. A General and Adaptive Robust Loss Function. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[8] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2019.

[9] A. Bendale and T.E. Boult. Towards open set deep networks. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[10] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger. MVTec AD–A comprehensive real-world dataset for unsupervised anomaly detection. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[11] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger. Uninformed Students: Student-Teacher Anomaly Detection With Discriminative Latent Embeddings. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[12] M. Berman, A.R. Triki, and M.B. Blaschko. The Lovász-Softmax Loss: A Tractable Surrogate for the Optimization of the Intersection-Over-Union Measure in Neural Networks. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[13] V. Besnier, A. Bursuc, D. Picard, and A. Briot. Triggering Failures: Out-Of-Distribution detection by learning from local adversarial attacks in Semantic Segmentation. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2021.

[14] F. Blochliger, M. Fehr, M. Dymczyk, T. Schneider, and R. Siegwart. TopoMap: Topological mapping and navigation based on visual SLAM maps. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2018.

[15] P.M. Blok, F. Magistri, C. Stachniss, H. Wang, J. Burridge, and W. Guo. High-throughput 3D shape completion of potato tubers on a harvester. *Computers and Electronics in Agriculture*, 228:109673, 2025.

[16] H. Blum, A. Gawel, R. Siegwart, and C. Cadena. Modular Sensor Fusion for Semantic Segmentation. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2018.

[17] H. Blum, P.E. Sarlin, J. Nieto, R. Siegwart, and C. Cadena. Fishyscapes: A benchmark for safe semantic segmentation in autonomous driving. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition Workshops*, 2019.

[18] H. Blum, P.E. Sarlin, J. Nieto, R. Siegwart, and C. Cadena. The Fishyscapes Benchmark: Measuring blind spots in semantic segmentation. *Intl. Journal of Computer Vision (IJCV)*, 129:3119–3135, 2021.

[19] P. Bosilj, E. Aptoula, T. Duckett, and G. Cielniak. Transfer learning between crop types for semantic segmentation of crops versus weeds in precision agriculture. *Journal of Field Robotics (JFR)*, 37(1):7–19, 2020.

[20] A. Bouguettaya, H. Zarzour, A. Kechida, and A.M. Taberkit. A survey on deep learning-based identification of plant and crop diseases from uav-based aerial images. *Cluster Computing*, 26(2):1297–1317, 2023.

[21] L. Calvin. *The US produce industry and labor: Facing the future in a global economy*. DIANE Publishing, 2010.

[22] Y.P. Cao, L. Kobbelt, and S.M. Hu. Real-time high-accuracy three-dimensional reconstruction with consumer rgb-d cameras. *ACM Trans. on Graphics (TOG)*, 37(5):1–16, 2018.

[23] D.O. Cardoso, J. Gama, and F.M. França. Weightless neural networks for open set recognition. *Machine Learning*, 106(9-10):1547–1567, 2017.

[24] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2020.

[25] E. Castro, J.S. Cardoso, and J.C. Pereira. Elastic deformations for data augmentation in breast cancer mass detection. In *Proc. of the Intl. Conf. on Biomedical & Health Informatics (BHI)*, 2018.

[26] J. Champ, A. Mora-Fallas, H. Goëau, E. Mata-Montero, P. Bonnet, and A. Joly. Instance segmentation for the fine detection of crop and weed plants by precision agricultural robots. *Applications in Plant Sciences*, 8(7):e11373, 2020.

[27] R. Chan, K. Lis, S. Uhlemeyer, H. Blum, S. Honari, R. Siegwart, P. Fua, M. Salzmann, and M. Rottmann. SegmentMeIfYouCan: A Benchmark for Anomaly Segmentation. In *Proc. of the Conf. on Neural Information Processing Systems (NeurIPS)*, 2021.

[28] A. Chaudhury, F. Boudon, and C. Godin. 3D plant phenotyping: All you need is labelled point cloud data. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2020.

[29] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In *Proc. of the Intl. Conf. on Machine Learning (ICML)*, 2020.

[30] B. Cheng, M.D. Collins, Y. Zhu, T. Liu, T.S. Huang, H. Adam, and L.C. Chen. Panoptic-DeepLab: A Simple, Strong, and Fast Baseline for Bottom-Up Panoptic Segmentation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[31] B. Cheng, I. Misra, A.G. Schwing, A. Kirillov, and R. Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[32] B. Cheng, A.G. Schwing, and A. Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *Proc. of the Conf. on Neural Information Processing Systems (NeurIPS)*, 2021.

[33] W. Chidsin, Y. Gu, and I. Goncharenko. AR-based navigation using RGB-D camera and hybrid map. *Sustainability*, 13(10):5585, 2021.

[34] F. Chollet. Xception: Deep Learning With Depthwise Separable Convolutions. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[35] C. Choy, J. Gwak, and S. Savarese. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[36] M. Cordts, S.M. Omran, Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[37] C. Couprie, C. Farabet, L. Najman, and Y. Lecun. Indoor semantic segmentation using depth information. In *Proc. of the Intl. Conf. on Learning Representations (ICLR)*, 2013.

[38] G. Csurka. *Domain adaptation in computer vision applications*. Springer, 2017.

[39] L. da Vinci. *Codex Atlanticus*. Mediolani: Apud P. Leonem, 1519.

[40] A. Dai, A. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[41] A. Dai and M. Nießner. 3DMV: Joint 3D-Multi-View Prediction for 3D Semantic Scene Segmentation. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2018.

[42] Y. Dai, F. Gieseke, S. Oehmcke, Y. Wu, and K. Barnard. Attentional feature fusion. In *Proc. of the IEEE Winter Conf. on Applications of Computer Vision (WACV)*, 2021.

[43] A. Delić, M. Grcić, and S. Segvić. Outlier detection by ensembling uncertainty with negative objectness. In *Proc. of the British Machine Vision Conf. (BMVC)*, 2024.

[44] X. Deng, A. Mousavian, Y. Xiang, F. Xia, T. Bretl, and D. Fox. PoseRBPF: A Rao–Blackwellized Particle Filter for 6-D Object Pose Tracking. *IEEE Trans. on Robotics (TRO)*, 37(5):1328–1342, 2021.

[45] T. DeVries and G.W. Taylor. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint*, arXiv:1802.04865, 2018.

[46] A.R. Dhamija, M. Günther, and T. Boult. Reducing network agnostophobia. In *Proc. of the Conf. on Neural Information Processing Systems (NeurIPS)*, 2018.

[47] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *Proc. of the Intl. Conf. on Learning Representations (ICLR)*, 2021.

[48] H. Dutagaci, P. Rasti, G. Galopin, and D. Rousseau. ROSE-X: an annotated data set for evaluation of 3D plant organ segmentation methods. *Plant Methods*, 16:1–14, 2020.

[49] F. Engelmann, M. Bokeloh, A. Fathi, B. Leibe, and M. Nießner. 3D-MPA: Multi-Proposal Aggregation for 3D Semantic Instance Segmentation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[50] M. Ester, H. Kriegel, J. Sander, and X.Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of the Conf. on Knowledge Discovery and Data Mining (KDD)*, 1996.

[51] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *Intl. Journal of Computer Vision (IJCV)*, 88(2):303–338, 2010.

[52] Y. Gal and Z. Ghahramani. Dropout as a Bayesian Approximation: Representing model uncertainty in deep learning. In *Proc. of the Intl. Conf. on Machine Learning (ICML)*, 2016.

[53] N. Gao, Y. Shan, Y. Wang, X. Zhao, Y. Yu, M. Yang, and K. Huang. SSAP: Single-Shot Instance Segmentation With Affinity Pyramid. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2019.

[54] S. Gasperini, M.N. Mahani, A. Marcos-Ramiro, N. Navab, and F. Tombari. Panoster: End-To-End Panoptic Segmentation of LiDAR Point Clouds. *IEEE Robotics and Automation Letters (RA-L)*, 6(2):3216–3223, 2021.

[55] S. Gasperini, A. Marcos-Ramiro, M. Schmidt, N. Navab, B. Busam, and F. Tombari. Segmenting known objects and unseen unknowns without prior knowledge. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2023.

[56] J. Gené-Mola, R. Sanz-Cortiella, J.R. Rosell-Polo, J.R. Morros, J. Ruiz-Hidalgo, V. Vilaplana, and E. Gregorio. Fuji-SfM dataset: A collection of annotated images and point clouds for Fuji apple detection and location using structure-from-motion photogrammetry. *Data in Brief*, 30:105591, 2020.

[57] J. Gené-Mola, R. Sanz-Cortiella, J.R. Rosell-Polo, J.R. Morros, J. Ruiz-Hidalgo, V. Vilaplana, and E. Gregorio. Fruit detection and 3d location using instance segmentation neural networks and structure-from-motion photogrammetry. *Computers and Electronics in Agriculture*, 169:105165, 2020.

[58] G. Ghiasi, X. Gu, Y. Cui, and T.Y. Lin. Scaling open-vocabulary image segmentation with image-level labels. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2022.

[59] R. Girshick. Fast R-CNN. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2015.

[60] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proc. of the Intl. Conf. on Artificial Intelligence and Statistics*, 2010.

[61] M. Grcić, P. Bevandić, and S. Šegvić. Densehybrid: Hybrid anomaly detection for dense open-set recognition. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2022.

[62] T. Guadagnino, X. Chen, M. Sodano, J. Behley, G. Grisetti, and C. Stachniss. Fast Sparse LiDAR Odometry Using Self-Supervised Feature Selection on Intensity Images. *IEEE Robotics and Automation Letters (RA-L)*, 7(3):7597–7604, 2022.

[63] R. Güldenring, R.E. Andersen, and L. Nalpantidis. Zoom in on the Plant: Fine-Grained Analysis of Leaf, Stem, and Vein Instances. *IEEE Robotics and Automation Letters (RA-L)*, 9(2):1588–1595, 2024.

[64] Y. Guo, Y. Liu, T. Georgiou, and M.S. Lew. A review of semantic segmentation using deep neural networks. *Intl. Journal of Multimedia Information Retrieval*, 7(2):87–93, 2018.

[65] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik. Learning rich features from RGB-D images for object detection and segmentation. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2014.

[66] S. Halligan, D.G. Altman, and S. Mallett. Disadvantages of using the area under the receiver operating characteristic curve to assess imaging tests: a discussion and proposal for an alternative approach. *European Radiology*, 25:932–939, 2015.

[67] L. Han, T. Zheng, L. Xu, and L. Fang. OccuSeg: Occupancy-Aware 3D Instance Segmentation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[68] N. Häni, P. Roy, and V. Isler. Minneapple: a benchmark dataset for apple detection and segmentation. *IEEE Robotics and Automation Letters (RA-L)*, 5(2):852–858, 2020.

[69] P.E. Hart, N.J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. on Systems Science and Cybernetics (TSSC)*, 4(2):100–107, 1968.

[70] C. Hazirbas, L. Ma, C. Domokos, and D. Cremers. Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture. In *Proc. of the Asian Conf. on Computer Vision (ACCV)*, 2016.

[71] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, 2017.

[72] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[73] D. Hendrycks, S. Basart, M. Mazeika, A. Zou, J. Kwon, M. Mostajabi, J. Steinhardt, and D. Song. Scaling out-of-distribution detection for real-world settings. *Proc. of the Intl. Conf. on Machine Learning (ICML)*, 2022.

[74] D. Hendrycks and K. Gimpel. Gaussian Error Linear Units (GELUs). *arXiv preprint*, arXiv:1606.08415, 2016.

[75] D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *Proc. of the Intl. Conf. on Learning Representations (ICLR)*, 2017.

[76] J. Hou, A. Dai, and M. Nießner. 3D-SIS: 3D Semantic Instance Segmentation of RGB-D Scans. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[77] J. Hu, L. Shen, and G. Sun. Squeeze-and-Excitation Networks. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[78] G. Humblot-Renaux, S. Escalera, and T.B. Moeslund. Beyond AUROC & co. for evaluating out-of-distribution detection performance. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[79] J. Hwang, S.W. Oh, J.Y. Lee, and B. Han. Exemplar-based open-set panoptic segmentation network. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[80] M.J. Islam, C. Edge, Y. Xiao, P. Luo, M. Mehtaz, C. Morse, S.S. Enan, and J. Sattar. Semantic segmentation of underwater imagery: Dataset and benchmark. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2020.

[81] Y. Itzhaky, G. Farjon, F. Khoroshevsky, A. Shpigler, and A. Bar-Hillel. Leaf Counting: Multiple Scale Regression and Detection Using Deep CNNs. In *Proc. of the British Machine Vision Conf. (BMVC)*, 2018.

[82] W.S. Jeon, G. Cielniak, and S.Y. Rhee. Semantic segmentation using trade-off and internal ensemble. *Intl. Journal of Fuzzy Logic and Intelligent Systems*, 18(3):196–203, 2018.

[83] H.R.V. Joze, A. Shaban, M.L. Iuzzolino, and K. Koishida. MMTM: Multimodal transfer module for CNN fusion. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[84] J. Kierdorf, L.V. Junker-Frohn, M. Delaney, M.D. Olave, A. Burkart, H. Jaenicke, O. Muller, U. Rascher, and R. Roscher. Growliflower: An image time series dataset for growth analysis of cauliflower. *Journal of Field Robotics (JFR)*, 40(2):173–192, 2022.

[85] T. Kim, S. Lim, G. Shin, G. Sim, and D. Yun. An open-source low-cost mobile robot system with an rgb-d camera and efficient real-time navigation algorithm. *IEEE Access*, 10:127871–127881, 2022.

[86] D. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *Proc. of the Intl. Conf. on Learning Representations (ICLR)*, 2015.

[87] A. Kirillov, R. Girshick, K. He, and P. Dollár. Panoptic Feature Pyramid Networks. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[88] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár. Panoptic Segmentation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[89] 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 *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2023.

[90] M. Kolodiazhnyi, A. Vorontsova, A. Konushin, and D. Rukhovich. Top-down beats bottom-up in 3D instance segmentation. In *Proc. of the IEEE Winter Conf. on Applications of Computer Vision (WACV)*, 2024.

[91] S. Kong and D. Ramanan. Opengan: Open-set recognition via open data generation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[92] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Proc. of the Conf. Neural Information Processing Systems (NIPS)*, 2012.

[93] V. Kulikov and V. Lempitsky. Instance Segmentation of Biological Images using Harmonic Embeddings. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[94] B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles. In *Proc. of the Conf. Neural Information Processing Systems (NIPS)*, 2017.

[95] P.D. Lancashire, H. Bleiholder, T.V.D. Boom, P. Langelüddeke, R. Stauss, E. Weber, and A. Witzenberger. A uniform decimal code for growth stages of crops and weeds. *Annals of Applied Biology*, 119(3):561–601, 1991.

[96] C. Lehnert, C. McCool, Y. Sa, and T. Perez. Performance improvements of a sweet pepper harvesting robot in protected cropping environments. *Journal of Field Robotics (JFR)*, 37(7):1197–1223, 2020.

[97] S. Leveugle, C.W. Lee, S. Stolpner, C. Langley, P. Grouchy, S. Waslander, and J. Kelly. ALLO: A Photorealistic Dataset and Data Generation Pipeline for Anomaly Detection During Robotic Proximity Operations in Lunar Orbit. *arXiv preprint*, arXiv:2409.20435, 2024.

[98] B. Li, J. Lecourt, and G. Bishop. Advances in non-destructive early assessment of fruit ripeness towards defining optimal time of harvest and yield prediction—a review. *Plants*, 7(1):3, 2018.

[99] L. Li, T. Zhou, W. Wang, J. Li, and Y. Yang. Deep hierarchical semantic segmentation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[100] X. Li, H. Ding, H. Yuan, W. Zhang, J. Pang, G. Cheng, K. Chen, Z. Liu, and C.C. Loy. Transformer-based visual segmentation: A survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 46(12):10138–10163, 2024.

[101] Y. Li, X. Chen, Z. Zhu, L. Xie, G. Huang, D. Du, and X. Wang. Attention-Guided Unified Network for Panoptic Segmentation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[102] Y. Li, H. Zhao, X. Qi, L. Wang, Z. Li, J. Sun, and J. Jia. Fully convolutional networks for panoptic segmentation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[103] 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 *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[104] S. Liang, Y. Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *Proc. of the Intl. Conf. on Learning Representations (ICLR)*, 2018.

[105] K. Lin, L. Wang, K. Luo, Y. Chen, Z. Liu, and M.T. Sun. Cross-domain complementary learning using pose for multi-person part segmentation. *IEEE Trans. on Circuits and Systems for Video Technology (TCSVT)*, 31(3):1066–1078, 2020.

[106] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C.L. Zitnick. Microsoft COCO: Common Objects in Context. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2014.

[107] H. Liu, C. Peng, C. Yu, J. Wang, X. Liu, G. Yu, and W. Jiang. An End-To-End Network for Panoptic Segmentation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[108] J. Liu, C. Chang, J. Liu, X. Wu, L. Ma, and X. Qi. MarS3D: A Plug-and-Play Motion-Aware Model for Semantic Segmentation on Multi-Scan 3D Point Clouds. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[109] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia. Path Aggregation Network for Instance Segmentation. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[110] W. Liu, X. Wang, J. Owens, and Y. Li. Energy-based out-of-distribution detection. In *Proc. of the Conf. on Neural Information Processing Systems (NeurIPS)*, 2020.

[111] Z. Liu, H. Mao, C.Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie. A convnet for the 2020s. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[112] L. Lobefaro, M. Sodano, D. Fusaro, F. Magistri, M. Malladi, T. Guadagnino, A. Pretto, and C. Stachniss. Spatio-Temporal Consistent Semantic Mapping for Robotics Fruit Growth Monitoring. *IEEE Robotics and Automation Letters (RA-L)*, 10(9):9470–9477, 2025.

[113] M. Loghmani, M. Planamente, B. Caputo, and M. Vincze. Recurrent Convolutional Fusion for RGB-D Object Recognition. *IEEE Robotics and Automation Letters (RA-L)*, 4(3):2878–2885, 2019.

[114] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[115] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *Proc. of the Intl. Conf. on Learning Representations (ICLR)*, 2019.

[116] P. Lottes, J. Behley, N. Chebrolu, A. Milioto, and C. Stachniss. Joint Stem Detection and Crop-Weed Classification for Plant-specific Treatment in Precision Farming. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2018.

[117] P. Lottes, J. Behley, N. Chebrolu, A. Milioto, and C. Stachniss. Robust Joint Stem Detection and Crop-Weed Classification using Image Sequences for Plant-Specific Treatment in Precision Farming. *Journal of Field Robotics (JFR)*, 37(1):20–34, 2020.

[118] P. Lottes, J. Behley, A. Milioto, and C. Stachniss. Fully Convolutional Networks with Sequential Information for Robust Crop and Weed Detection

in Precision Farming. *IEEE Robotics and Automation Letters (RA-L)*, 3(4):3097–3104, 2018.

[119] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *Intl. Journal of Computer Vision (IJCV)*, 60(2):91–110, 2004.

[120] Y. Lu and S. Young. A survey of public datasets for computer vision tasks in precision agriculture. *Computers and Electronics in Agriculture*, 178:105760, 2020.

[121] K. Maag and T. Riedlinger. Pixel-wise gradient uncertainty for convolutional neural networks applied to out-of-distribution segmentation. In *Intl. Conf. on Computer Vision Theory and Applications (VISAPP)*, 2024.

[122] F. Magistri, R. Marcuzzi, E. Marks, M. Sodano, J. Behley, and C. Stachniss. Efficient and Accurate Transformer-Based 3D Shape Completion and Reconstruction of Fruits for Agricultural Robots. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2024.

[123] F. Magistri, Y. Pan, J. Bartels, J. Behley, C. Stachniss, and C. Lehnert. Improving Robotic Fruit Harvesting Within Cluttered Environments Through 3D Shape Completion. *IEEE Robotics and Automation Letters (RA-L)*, 9(8):7357–7364, 2024.

[124] M. Malladi, N. Chebrolu, I. Scacchetti, L. Lobefaro, T. Guadagnino, B. Casseau, H. Oh, L. Freissmuth, M. Karppinen, J. Schweier, S. Leutenegger, J. Behley, C. Stachniss, and M. Fallon. DigiForests: A Longitudinal LiDAR Dataset for Forestry Robotics. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2025.

[125] R. Marcuzzi, L. Nunes, L. Wiesmann, J. Behley, and C. Stachniss. Mask-Based Panoptic LiDAR Segmentation for Autonomous Driving. *IEEE Robotics and Automation Letters (RA-L)*, 8(2):1141–1148, 2023.

[126] E. Marks, L. Nunes, F. Magistri, M. Sodano, R. Marcuzzi, L. Zimmermann, J. Behley, and C. Stachniss. Tree Skeletonization from 3D Point Clouds by Denoising Diffusion. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2025.

[127] E. Marks, M. Sodano, F. Magistri, L. Wiesmann, D. Desai, R. Marcuzzi, J. Behley, and C. Stachniss. High precision leaf instance segmentation for phenotyping in point clouds obtained under real field conditions. *IEEE Robotics and Automation Letters (RA-L)*, 8(8):4791–4798, 2023.

[128] E. Marks, J. Bömer, F. Magistri, A. Sah, J. Behley, and C. Stachniss. BonnBeetClouds3D: A Dataset Towards Point Cloud-Based Organ-Level Phenotyping of Sugar Beet Plants Under Real Field Conditions. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2024.

[129] C. McCool, T. Perez, and B. Upcroft. Mixtures of Lightweight Deep Convolutional Neural Networks: Applied to Agricultural Robotics. *IEEE Robotics and Automation Letters (RA-L)*, 2(3):1344–1351, 2017.

[130] C. McCool, T. Perez, and B. Upcroft. Mixtures of Lightweight Deep Convolutional Neural Networks: Applied to Agricultural Robotics. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2017.

[131] L. McInnes, J. Healy, and S. Astels. HDBSCAN: Hierarchical sensity based clustering. *Journal of Open Source Software*, 2(11):205–206, 2017.

[132] A. Milioto, J. Behley, C. McCool, and C. Stachniss. LiDAR Panoptic Segmentation for Autonomous Driving. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2020.

[133] A. Milioto, P. Lottes, and C. Stachniss. Real-time blob-wise sugar beets vs weeds classification for monitoring fields using convolutional neural networks. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2017.

[134] A. Milioto, P. Lottes, and C. Stachniss. Real-time Semantic Segmentation of Crop and Weed for Precision Agriculture Robots Leveraging Background Knowledge in CNNs. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2018.

[135] A. Milioto, L. Mandtler, and C. Stachniss. Fast Instance and Semantic Segmentation Exploiting Local Connectivity, Metric Learning, and One-Shot Detection for Robotics. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2019.

[136] A. Milioto and C. Stachniss. Bonnet: An Open-Source Training and Deployment Framework for Semantic Segmentation in Robotics using CNNs. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2019.

[137] H.A. Montes, J. Le Louedec, G. Cielniak, and T. Duckett. Real-time detection of broccoli crops in 3d point clouds for autonomous robotic harvesting. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2020.

[138] N. Mor and L. Wolf. Confidence prediction for lexicon-free ocr. In *Proc. of the IEEE Winter Conf. on Applications of Computer Vision (WACV)*, 2018.

[139] D. Morris. A Pyramid CNN for Dense-Leaves Segmentation. In *Proc. of the Conf. on Computer and Robot Vision (CRV)*, 2018.

[140] Y.Z. Mu, C.Y. Dong, Q.M. Chen, B.C. Li, and Z.Q. Fan. Research on navigation and path planning of mobile robot based on vision sensor. In *Proc. of the Intl. Conf. on Computing and Artificial Intelligence (ICCAI)*, 2020.

[141] N. Nayal, M. Yavuz, J.F. Henriques, and F. Güney. RbA: Segmenting unknown regions rejected by all. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2023.

[142] A. Nekrasov, A. Hermans, L. Kuhnert, and B. Leibe. Ugains: Uncertainty guided anomaly instance segmentation. In *Proc. of the German Conf. on Pattern Recognition (GCPR)*, 2023.

[143] A. Nekrasov, R. Zhou, M. Ackermann, A. Hermans, B. Leibe, and M. Rottmann. Oodis: Anomaly instance segmentation benchmark. *arXiv preprint*, arXiv:2406.11835, 2024.

[144] D. Neven, B.D. Brabandere, M. Proesmans, and L.V. Gool. Instance Segmentation by Jointly Optimizing Spatial Embeddings and Clustering Bandwidth. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[145] A. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[146] N.J. Nilsson. Shakey the Robot. https://ai.stanford.edu/ nilsson/OnlinePubs-Nils/shakey-the-robot.pdf, Accessed: Mar. 4, 2026.

[147] OpenAI. GPT-4V(ision) System Card. https://openai.com/index/gpt-4v-system-card/, Accessed: Mar. 4, 2026.

[148] M. Orsić, I. Kreso, P. Bevandić, and S. Segvić. In Defense of Pre-Trained ImageNet Architectures for Real-Time Semantic Segmentation of Road-Driving Images. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[149] O.K. Oyedotun, K. Al Ismaeil, and D. Aouada. Why is everyone training very deep neural network with skip connections? *IEEE Trans. on Neural Networks and Learning Systems (TNNLS)*, 34(9):5961–5975, 2022.

[150] Y. Pan, F. Magistri, T. Läbe, E. Marks, C. Smitt, C. McCool, J. Behley, and C. Stachniss. Panoptic Mapping with Fruit Completion and Pose Estimation for Horticultural Robots. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2023.

[151] Y. Pan, C. Chen, D. Li, Z. Zhao, and J. Hong. Augmented reality-based robot teleoperation system using rgb-d imaging and attitude teaching device. *Robotics and Computer-Integrated Manufacturing*, 71:102167, 2021.

[152] J. Park, Q. Zhou, and V. Koltun. Colored Point Cloud Registration Revisited. In *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, 2017.

[153] S. Park, K. Hong, and S. Lee. RDFNet: RGB-D Multi-Level Residual Feature Fusion for Indoor Semantic Segmentation. In *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, 2017.

[154] Perplexity Deep Research. Perplexity AI. https://www.perplexity.ai/hub/blog/introducing-perplexity-deep-research, Accessed: Mar. 4, 2026.

[155] P. Pinggera, S. Ramos, S. Gehrig, U. Franke, C. Rother, and R. Mester. Lost and found: detecting small road hazards for self-driving vehicles. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2016.

[156] S.J.D. Prince. *Understanding Deep Learning*. MIT Press, 2023.

[157] I. Pérez-Borrero, D. Marín-Santos, M.E. Gegúndez-Arias, and E. Cortés-Ancos. A fast and accurate deep learning method for strawberry instance segmentation. *Computers and Electronics in Agriculture*, 178:105736, 2020.

[158] X. Qi, R. Liao, J. Jia, S. Fidler, and R. Urtasun. 3D Graph Neural Networks for RGBD Semantic Segmentation. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2017.

[159] Z. Qin, J. Chen, C. Chen, X. Chen, and X. Li. UniFusion: Unified Multi-View Fusion Transformer for Spatial-Temporal Representation in Bird's-Eye-View. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[160] A. Radford, J.W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In *Proc. of the Intl. Conf. on Machine Learning (ICML)*, 2021.

[161] S.N. Rai, F. Cermelli, B. Caputo, and C. Masone. Mask2anomaly: Mask transformer for universal open-set segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 46(12):9286–9302, 2024.

[162] R. Reiter. On closed world data bases. In *Readings in Artificial Intelligence*, 1981.

[163] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Proc. of the Conf. Neural Information Processing Systems (NIPS)*, 2015.

[164] M. Roberts, J. Ramapuram, A. Ranjan, A. Kumar, M.A. Bautista, N. Paczan, R. Webb, and J.M. Susskind. Hypersim: A Photorealistic Synthetic Dataset for Holistic Indoor Scene Understanding. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2021.

[165] G. Roggiolani, M. Sodano, T. Guadagnino, F. Magistri, J. Behley, and C. Stachniss. Hierarchical Approach for Joint Semantic, Plant Instance, and Leaf Instance Segmentation in the Agricultural Domain. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2023.

[166] E. Romera, J.M. Alvarez, L.M. Bergasa, and R. Arroyo. ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation. *IEEE Trans. on Intelligent Transportation Systems (TITS)*, 19(1):263–272, 2018.

[167] B. Romera-Paredes and P. Torr. Recurrent Instance Segmentation. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2016.

[168] O. Ronneberger, P.Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Proc. of the Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.

[169] A. Rosenberg and J. Hirschberg. V-measure: A Conditional Entropy-Based External Cluster Evaluation Measure. In *Proc. of the Joint Conf. on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007.

[170] K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox, and P. Gehler. Towards Total Recall in Industrial Anomaly Detection. In *Proc. of the*

IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), 2022.

[171] M. Salehi, N. Sadjadi, S. Baselizadeh, M.H. Rohban, and H.R. Rabiee. Multiresolution Knowledge Distillation for Anomaly Detection. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[172] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.C. Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[173] H. Sapkota and Q. Yu. Bayesian Nonparametric Submodular Video Partition for Robust Anomaly Detection. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[174] H. Scharr, M. Minervini, A. Fischbach, and S.A. Tsaftaris. Annotated Image Datasets of Rosette Plants. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2014.

[175] S. Schmidt, J. Körner, D. Fuchsgruber, S. Gasperini, F. Tombari, and S. Günnemann. Prior2former–evidential modeling of mask transformers for assumption-free open-world panoptic segmentation. *arXiv preprint*, arXiv:2504.04841, 2025.

[176] D. Schunck, F. Magistri, R. Rosu, A. Cornelißen, N. Chebrolu, S. Paulus, J. Léon, S. Behnke, C. Stachniss, H. Kuhlmann, and L. Klingbeil. Pheno4D: A spatio-temporal dataset of maize and tomato plant point clouds for phenotyping and advanced plant analysis. *PLoS ONE*, 16(8):1–18, 2021.

[177] D. Seichter, M. Köhler, B. Lewandowski, T. Wengefeld, and H. Gross. Efficient RGB-D Semantic Segmentation for Indoor Scene Analysis. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2021.

[178] W. Shi, R. van de Zedde, H. Jiang, and G. Kootstra. Plant-part segmentation using deep learning and multi-view vision. *Biosystems Engineering*, 187:81–95, 2019.

[179] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. of the Intl. Conf. on Learning Representations (ICLR)*, 2015.

[180] K. Sirohi, R. Mohan, D. Büscher, W. Burgard, and A. Valada. EfficientLPS: Efficient LiDAR Panoptic Segmentation. *IEEE Trans. on Robotics (TRO)*, 38(3):1894–1914, 2021.

[181] L.N. Smith and N. Topin. Super-convergence: Very fast training of neural networks using large learning rates. *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, 11006:369–386, 2019.

[182] C. Smitt, M. Halstead, P. Zimmer, T. Läbe, E. Guclu, C. Stachniss, and C. McCool. PAg-NeRF: Towards fast and efficient end-to-end panoptic 3D representations for agricultural robotics. *IEEE Robotics and Automation Letters (RA-L)*, 9(1):907–914, 2024.

[183] C. Smitt, M. Halstead, T. Zaenker, M. Bennewitz, and C. McCool. PATHoBot: A robot for glasshouse crop phenotyping and intervention. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2021.

[184] M. Sodano, F. Magistri, J. Behley, and C. Stachniss. Open-World Panoptic Segmentation. *arXiv preprint*, arXiv:2412.12740, 2024.

[185] M. Sodano, F. Magistri, E. Marks, F. Hosn, A. Zurbayev, R. Marcuzzi, M. Malladi, J. Behley, and C. Stachniss. 3D Hierarchical Panoptic Segmentation in Real Orchard Environments Across Different Sensors. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2025.

[186] M. Sodano, F. Magistri, L. Nunes, J. Behley, and C. Stachniss. Open-World Semantic Segmentation Including Class Similarity. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[187] M. Sodano, F. Magistri, T. Guadagnino, J. Behley, and C. Stachniss. Robust Double-Encoder Network for RGB-D Panoptic Segmentation. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2023.

[188] A. Steiner, A.S. Pinto, M. Tschannen, D. Keysers, X. Wang, Y. Bitton, A. Gritsenko, M. Minderer, A. Sherbondy, S. Long, S. Qin, R. Ingle, E. Bugliarello, S. Kazemzadeh, T. Mesnard, I. Alabdulmohsin, L. Beyer, and X. Zhai. Paligemma 2: A family of versatile VLMs for transfer. *arXiv preprint*, arXiv:2412.03555, 2024.

[189] J.R. Teasdale and D.W. Shirley. Influence of herbicide application timing on corn production in a hairy vetch cover crop. *Journal of Production Agriculture*, 11(1):121–125, 1998.

[190] Z. Tian, C. Shen, and H. Chen. Conditional convolutions for instance segmentation. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2020.

[191] B. Triggs, P.F. McLauchlan, R.I. Hartley, and A.W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *Proc. of the Intl. Workshop on Vision Algorithms: Theory and Practice*, 1999.

[192] C.C. Tsai, T.H. Wu, and S.H. Lai. Multi-scale patch-based representation learning for image anomaly detection and segmentation. In *Proc. of the IEEE Winter Conf. on Applications of Computer Vision (WACV)*, 2022.

[193] A. Valada, R. Mohan, and W. Burgard. Self-supervised model adaptation for multimodal semantic segmentation. *Intl. Journal of Computer Vision (IJCV)*, 128(5):1239–1285, 2019.

[194] T. van Klompenburg, A. Kassahun, and C. Catal. Crop yield prediction using machine learning: A systematic literature review. *Computers and Electronics in Agriculture*, 177:105709, 2020.

[195] B.M. van Marrewijk, T. van Daalen, K. Smoleňová, B. Xin, G. Polder, and G. Kootstra. TomatoWUR: An annotated dataset of tomato plants to quantitatively evaluate segmentation, skeletonisation, and plant-trait extraction algorithms for 3D plant phenotyping. *Data in Brief*, 61:111852, 2025.

[196] S. Vaze, K. Han, A. Vedaldi, and A. Zisserman. Open-set recognition: A good closed-set classifier is all you need. In *Proc. of the Intl. Conf. on Learning Representations (ICLR)*, 2021.

[197] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2001.

[198] I. Vizzo, B. Mersch, L. Nunes, L. Wiesmann, T. Guadagnino, and C. Stachniss. Toward reproducible version-controlled perception platforms: Embracing simplicity in autonomous vehicle dataset acquisition. In *Proc. of the IEEE Intl. Conf. on Intelligent Transportation Systems (ITSC)*, 2023.

[199] A. Vyas, N. Jammalamadaka, X. Zhu, D. Das, B. Kaul, and T.L. Willke. Out-of-distribution detection using an ensemble of self supervised leave-out classifiers. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2018.

[200] H. Wang, Y. Zhu, H. Adam, A. Yuille, and L.C. Chen. MaX-DeepLab: End-to-end panoptic segmentation with mask transformers. In *Proc. of the*

IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), 2021.

[201] H. Wang, Y. Zhu, B. Green, H. Adam, A. Yuille, and L.C. Chen. Axial-deeplab: Stand-alone axial-attention for panoptic segmentation. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2020.

[202] W. Wang and U. Neumann. Depth-aware cnn for rgb-d segmentation. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2018.

[203] Z. Wang, Q. Xu, Z. Yang, Y. He, X. Cao, and Q. Huang. OpenAUC: Towards AUC-Oriented Open-Set Recognition. *Proc. of the Conf. on Neural Information Processing Systems (NeurIPS)*, 2022.

[204] J. Weyler, , F. Magistri, P. Seitz, J. Behley, and C. Stachniss. In-Field Phenotyping Based on Crop Leaf and Plant Instance Segmentation. In *Proc. of the IEEE Winter Conf. on Applications of Computer Vision (WACV)*, 2022.

[205] J. Weyler, T. Läbe, J. Behley, and C. Stachniss. Panoptic Segmentation with Partial Annotations for Agricultural Robots. *IEEE Robotics and Automation Letters (RA-L)*, 9(2):1660–1667, 2024.

[206] J. Weyler, F. Magistri, E. Marks, Y.L. Chong, M. Sodano, G. Roggiolani, N. Chebrolu, C. Stachniss, and J. Behley. PhenoBench: A Large Dataset and Benchmarks for Semantic Image Interpretation in the Agricultural Domain. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 46(12):9583–9594, 2024.

[207] J. Weyler, J. Quakernack, P. Lottes, J. Behley, and C. Stachniss. Joint plant and leaf instance segmentation on field-scale uav imagery. *IEEE Robotics and Automation Letters (RA-L)*, 7(2):3787–3794, 2022.

[208] K. Wong, S. Wang, M. Ren, M. Liang, and R. Urtasun. Identifying unknown instances for autonomous driving. In *Proc. of the Conf. on Robot Learning (CoRL)*, 2020.

[209] S. Woo, J. Park, J.Y. Lee, and I.S. Kweon. Cbam: Convolutional block attention module. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2018.

[210] S.C. Wu, J. Wald, K. Tateno, N. Navab, and F. Tombari. SceneGraphFusion: Incremental 3D Scene Graph Prediction From RGB-D Sequences. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[211] Z. Wu, G. Allibert, C. Stolz, C. Ma, and C. Demonceaux. Depth-adapted CNNs for RGB-D semantic segmentation. In *Proc. of the Asian Conf. on Computer Vision (ACCV)*, 2020.

[212] Y. Xing, J. Wang, X. Chen, and G. Zeng. 2.5D convolution for RGB-D semantic segmentation. In *Proc. of the IEEE Intl. Conf. on Image Processing (ICIP)*, 2019.

[213] Y. Xing, J. Wang, and G. Zeng. Malleable 2.5D convolution: Learning receptive fields along the depth-axis for RGB-D scene parsing. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2020.

[214] Y. Xiong, R. Liao, H. Zhao, R. Hu, M. Bai, E. Yumer, and R. Urtasun. UPSNet: A Unified Panoptic Segmentation Network. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[215] J. Xu, R. Zhang, J. Dou, Y. Zhu, J. Sun, and S. Pu. RPVNet: A Deep and Efficient Range-Point-Voxel Fusion Network for LiDAR Point Cloud Segmentation. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2021.

[216] X. Xu, P. Gao, X. Zhu, W. Guo, J. Ding, C. Li, M. Zhu, and X. Wu. Design of an integrated climatic assessment indicator (icai) for wheat production: A case study in jiangsu province, china. *Ecological Indicators*, 101:943–953, 2019.

[217] X. Xu, T. Xiong, Z. Ding, and Z. Tu. Masqclip for open-vocabulary universal image segmentation. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2023.

[218] Z. Xu, S. Escalera, A. Pavao, M. Richard, W.W. Tu, Q. Yao, H. Zhao, and I. Guyon. Codabench: Flexible, easy-to-use, and reproducible meta-benchmark platform. *Patterns*, 3(7), 2022.

[219] T.J. Yang, M.D. Collins, Y. Zhu, J.J. Hwang, T. Liu, X. Zhang, V. Sze, G. Papandreou, and L.C. Chen. Deeperlab: Single-shot image parser. *arXiv preprint*, arXiv:1902.05093, 2019.

[220] X. Yao, R. Li, J. Zhang, J. Sun, and C. Zhang. Explicit Boundary Guided Semi-Push-Pull Contrastive Learning for Supervised Anomaly Detection. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[221] 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 *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[222] V. Zavrtanik, M. Kristan, and D. Skoćaj. Draem-a discriminatively trained reconstruction embedding for surface anomaly detection. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2021.

[223] O. Zendel, K. Honauer, M. Murschitz, D. Steininger, and G.F. Dominguez. Wilddash-creating hazard-aware benchmarks. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2018.

[224] H. Zhang, F. Li, X. Zou, S. Liu, C. Li, J. Yang, and L. Zhang. A simple framework for open-vocabulary segmentation and detection. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2023.

[225] W. Zhang, J. Pang, K. Chen, and C.C. Loy. K-net: Towards unified image segmentation. *Proc. of the Conf. on Neural Information Processing Systems (NeurIPS)*, 2021.

[226] X. Zhang and Y. LeCun. Universum prescription: Regularization using unlabeled data. In *Proc. of the Conf. on Advancements of Artificial Intelligence (AAAI)*, 2017.

[227] X. Zhang, S. Li, X. Li, P. Huang, J. Shan, and T. Chen. DeSTSeg: Segmentation Guided Denoising Student-Teacher for Anomaly Detection. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[228] Y. Zhang, P.E. Latham, and A. Saxe. Understanding unimodal bias in multimodal deep linear networks. In *Proc. of the Intl. Conf. on Machine Learning (ICML)*, 2024.

[229] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid Scene Parsing Network. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[230] Y. Zhao. OmniAL: A Unified CNN Framework for Unsupervised Anomaly Localization. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[231] X. Zheng, Y. Lyu, and L. Wang. Learning modality-agnostic representation for semantic segmentation from any modalities. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2024.

[232] X. Zheng, Y. Lyu, J. Zhou, and L. Wang. Centering the value of every modality: Towards efficient and resilient modality-agnostic semantic segmentation. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2024.

[233] X. Zheng, H. Xue, J. Chen, Y. Yan, L. Jiang, Y. Lyu, K. Yang, L. Zhang, and X. Hu. Learning robust anymodal segmentor with unimodal and cross-modal distillation. *arXiv preprint*, arXiv:2411.17141, 2024.

[234] Z. Zhou, Y. Zhang, and H. Foroosh. Panoptic-PolarNet: Proposal-Free LiDAR Point Cloud Panoptic Segmentation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[235] J. Zhu, R. Zhai, H. Ren, K. Xie, A. Du, X. He, C. Cui, Y. Wang, J. Ye, J. Wang, X. Jiang, Y. Wang, C. Huang, and W. Yang. Crops3D: a diverse 3D crop dataset for realistic perception and segmentation toward agricultural applications. *Scientific Data*, 11(1):1438, 2024.

[236] Z. Zhu, F. Xu, C. Yan, X. Hao, X. Ji, Y. Zhang, and Q. Dai. Real-time indoor scene reconstruction with rgbd and inertial input. In *Proc. of the Intl. Conf. on Multimedia and Expo (ICME)*, 2019.

[237] N. Zimmerman, M. Sodano, E. Marks, J. Behley, and C. Stachniss. Constructing Metric-Semantic Maps using Floor Plan Priors for Long-Term Indoor Localization. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2023.

[238] X. Zou, Z.Y. Dou, J. Yang, Z. Gan, L. Li, C. Li, X. Dai, H. Behl, J. Wang, L. Yuan, N. Peng, L. Wang, Y. Lee, and J. Gao. Generalized decoding for pixel, image, and language. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.

List of Figures

List of Tables