Understanding and Mitigating Corner Cases with Contextual Knowledge in Deep Learning for Automated Driving

Zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

von der KIT-Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

M.Sc.

Jingxing Zhou

aus Hunan, China

Tag der mündlichen Prüfung:	24.07.2026
Erster Gutachter:	Prof. Dr.-Ing. habil. Jürgen Beyerer
Zweiter Gutachter:	Prof. Dr.-Ing. Steven Peters

Abstract

Data-driven computer vision has advanced significantly through machine learning approaches that do not require extensive feature engineering. However, deep neural networks can easily produce overconfident predictions when data during inference are not independent and identically distributed as the training set. For safety-critical systems like automated driving, neural networks must accommodate continuously changing input data distributions, such as the shifting weather conditions that can affect the sensors in real-world driving situations. Such properties of data-driven methods inhibit perception models from being deployed in mass-production vehicles, as the driving functions are mandatory to fulfill the quality requirements according to safety regulations and technical standards for automotive software development. In the automotive industry, the aforementioned driving scenarios that lead to inappropriate perception results and, by implication, generate undesirable driving behavior, are often regarded as corner cases. In these cases, vehicles equipped with automated driving functions encounter previously unknown objects or observe abnormal behavior of other traffic participants. Previous work that aimed at providing taxonomies of corner cases did not facilitate classifying such scenarios in this particular field of application and thus solving them. Therefore, this thesis not only strives to provide insights into defining those safety-critical driving situations from the data perspective but also to mitigate them efficiently in advance by systematic processes.

Starting with a preliminary study, it demonstrates that utilizing essential data augmentation methods, which maintain the integrity of the data distribution, can enhance the robustness of data-driven models, while distorted data for the training leads to increasing detection errors and, therefore, more corner cases. With such knowledge, this thesis systematically evaluates the causes

roots of corner cases in the context of automated driving functions, unveiling their deep connection to the availability of the data during development.

To mitigate those corner cases without acquiring extensive annotations, approaches are proposed to better utilize readily available prior knowledge, and are evaluated to determine how such knowledge can help deep learning-based image perception systems overcome their limitations dealing with unknown or underrepresented objects. Current research has not adequately addressed the concept of class hierarchies despite their natural discriminating properties connected with object classes. Starting from the object detection task, leveraging the knowledge learned from some frequent classes also helps to ensure the generalization ability of the neural networks so that a pre-defined hierarchical class taxonomy facilitates detecting previously unknown classes that belong to the same object category. Such a principle also applies to the segmentation tasks, where hierarchical class taxonomies provide valuable insights into semantic segmentation models dealing with previously unknown classes that belong to a known object category. With the help of the more effective feature representations brought by hierarchical taxonomies, learning a novel class with limited training data for corner case elimination demonstrates the potential of leveraging such a priori knowledge. Last but not least, the usage of other publicly available contextual knowledge for efficient domain adaptation is demonstrated with lane detection models. The performance of adapting a lane detection model to a previously unknown domain improves significantly while leveraging only meta information from the target domain.

In summary, this thesis contributes to the systematic understanding of corner cases for automated driving functions based on the availability of data. Leveraging contextual knowledge from various sources improves the models' capability and helps to proactively mitigate possible task-agnostic corner cases due to the knowledge gap without acquiring an extensive amount of data annotation. These observations hold the potential for more robust, reliable, and affordable assisted and automated driving systems.

Kurzfassung

Datengetriebenes maschinelles Sehen hat in dem letzten Jahrzehnt dank neuer maschineller Lernverfahren, die keine explizite und manuelle Merkmalsentwicklung für Bilder erfordern, enorme Fortschritte erzielt. Allerdings neigen tiefe neuronale Netze dazu, unzuverlässige Prädiktionen zu liefern, wenn die Daten während der Inferenz nicht unabhängig und identisch verteilt sind wie die Trainingsdaten. Für viele sicherheitskritische Systeme, wie automatisierte und assistierte Fahrfunktionen, ist es daher essenziell, dass neuronale Netze in der Lage sind, sich an die kontinuierlich ändernden Eingabedatenverteilungen anzupassen, beispielsweise wechselnde Wetterbedingungen, die die Fahrzeugsensorik in realen Fahrsituationen beeinflussen können.

Solche Eigenschaften erschweren den Einsatz datengetriebener Wahrnehmungsmodellen in Serienfahrzeugen, da die Fahrfunktionen zwingend die Qualitätsanforderungen gemäß Sicherheitsvorschriften und technischen Standards der automobilen Softwareentwicklung erfüllen müssen. In der Automobilindustrie werden die beschriebenen Fahrszenarien, die zu unangemessenen Wahrnehmungsergebnissen und folglich zu unerwünschtem Fahrverhalten führen, häufig als Corner Cases bezeichnet. In diesen Fällen begegnen Fahrzeuge mit automatisierten Fahrfunktionen zuvor unbekannten Objekten oder beobachten abnormales Verhalten anderer Verkehrsteilnehmer. Frühere Arbeiten, die Taxonomien von Corner Cases bereitstellten, können solche Szenarien im spezifischen Anwendungsfeld nicht ausreichend oder eindeutig genug klassifizieren und somit die Problematik nicht lösen. Diese Arbeit verfolgt daher das Ziel, nicht nur Einblicke in die Definition solcher sicherheitskritischen Fahrsituationen aus datenbezogener Perspektive zu geben, sondern diese auch systematisch im Vorfeld während der Funktionsentwicklung zu mitigieren.

Ausgehend von einer Vorstudie in der Arbeit wird gezeigt, dass der Einsatz wesentlicher Datenaugmentationsmethoden, die die Integrität der Datenverteilung in den Datensätzen bewahren, die Robustheit datengetriebener Modelle verbessert, während verzerrte Trainingsdaten zu erhöhten Erkennungsfehlern und somit potenziell zu mehr Corner Cases führen. Aufbauend auf diesem Wissen werden in der Arbeit die Ursachen von Corner Cases im Kontext der automatisierten Fahrfunktionen systematisch untersucht und deren enge Verbindung zur Verfügbarkeit der Datenpunkte während der Entwicklung aufgedeckt.

Um Corner Cases ohne umfangreiche manuelle Annotationen zu reduzieren, werden Ansätze analysiert und vorgeschlagen, die vorhandenes Vorwissen über Objekte und die Fahrumgebung effektiv in den maschinellen Lernprozess zu integrieren. Diese werden zuerst evaluiert, um zu bestimmen, wie solches Wissen die Wahrnehmungssysteme dabei unterstützt, effektiver mit unbekannten oder unterrepräsentierten Objekten umgehen zu können. Die aktuelle Forschung zu maschinellen Lernverfahren in der Bildwahrnehmung hat das Konzept von Klassenhierarchien trotz ihrer natürlichen diskriminierenden Eigenschaften im Zusammenhang mit Objektklassen bislang unzureichend berücksichtigt. Ausgehend von der Objekterkennung trägt das Lernen von häufig aufgetretenen Klassen dazu bei, die Generalisierungsfähigkeit neuronaler Netze zu gewährleisten, sodass eine vordefinierte hierarchische Klassentaxonomie die Erkennung zuvor unbekannter Klassen derselben Objektkategorie erleichtert.

Dieses Prinzip findet auch bei Segmentierungsaufgaben Anwendung, bei denen hierarchische Klassentaxonomien wertvolle Erkenntnisse für semantische Segmentierungsmodelle liefern, wie diese mit zuvor unbekannten Klassen innerhalb bekannter Objektkategorien umgehen können. Mithilfe der durch hierarchische Taxonomien verbesserten Merkmalsrepräsentationen zeigt das Lernen neuer Klassen mit begrenzten Trainingsdaten für die Mitigation von Corner Cases das Potenzial, solches Vorwissen effektiv zu nutzen. Nicht zuletzt wird der Einsatz weiterer öffentlich verfügbarer kontextueller Informationen für eine effiziente Domänenanpassung anhand von Spurerkennungsmodellen demonstriert. Die Leistung der Anpassung des

Modells an eine zuvor unbekannte Domäne verbessert sich signifikant, wenn lediglich Metainformationen aus der Ziel-Domäne genutzt werden.

Zusammenfassend leistet diese Arbeit einen Beitrag zum systematischen Verständnis von Corner Cases für automatisierte Fahrfunktionen basierend auf der Verfügbarkeit von Daten. Die Nutzung kontextuellen Wissens aus verschiedenen Quellen verbessert die Leistungsfähigkeit der Modelle und unterstützt die proaktive Mitigation möglicher Aufgaben-agnostischer Corner Cases, die durch epistemische Unsicherheit entstehen, ohne dass umfangreiche Datenannotationen erforderlich sind. Diese Erkenntnisse bergen das Potenzial für robustere, zuverlässigere und kostengünstigere assistierte und automatisierte Fahrsysteme.

Acknowledgements

This dissertation is the result of a rewarding journey at the Karlsruhe Institute of Technology and Porsche Engineering that would not have been possible without the support of many people.

First and foremost, I would like to express my sincere gratitude to my primary supervisor, Prof. Dr. Jürgen Beyerer, for his guidance and our many productive exchanges. I greatly appreciate the freedom he granted me in defining the thematic focus of this research. His valuable insights and suggestions have been instrumental in shaping this thesis.

I would also like to thank my co-supervisor, Prof. Dr. Steven Peters at TU Darmstadt, for his valuable perspectives and for our thought-provoking exchanges on how methods and ideas developed in this work could also be transferred to emerging concepts in automated driving.

I am grateful to Jun.-Prof. Dr. Maike Schwammberger for serving as examiner and for her suggestions during the PhD progress meeting. I thank Prof. Dr. Marvin Künnemann-Goos and Prof. Dr. Hannes Hartenstein for taking on the role of committee members and for their time in evaluating this thesis, as well as Prof. Dr. Jan Niehues for his feedback during the final phase of the work.

This work benefited greatly from the valuable guidance and support of Jens Ziehn, Dr. Masoud Roschani and Dr. Miriam Ruf at Fraunhofer IOSB. Our discussions were open, constructive, and always enriching. Their attention to even the smallest details helped strengthen many of the ideas and findings in this work.

My heartfelt thanks go to Dr. Joachim Schaper, my advisor at Porsche Engineering, for his continued guidance. His pragmatic approach and ability to distill complex problems down to their essentials were a great help in tackling numerous technical and organizational challenges. I thank Nikola Aschoff and Stefan Rathgeber at Porsche Engineering for their trust and for creating the space to pursue this research.

Special thanks go to Dr. Tobias Kalb for countless exchanges, debugging sessions, and our joint search for bugs that often proved more stubborn than expected. The hours we spent training models, analyzing results, and occasionally pushing the DGXs to their limits are among my most memorable experiences.

I would also like to thank colleagues and collaborators at Porsche, Fraunhofer IOSB and KIT — Anne, Chongzhe, Daniel, Hans, Johann, Lars, Leon, Marcel, Nick, Nicole, Raphael, Ruei-Bo, Sebastian, and Stefan — for their constructive feedback, inspiring conversations, and the many challenges we overcame together.

My deepest gratitude goes to my family. Their patience, understanding, and unwavering support sustained me throughout this endeavor. During times of setbacks and doubt, their encouragement gave me the strength to continue. They played an indispensable role in bringing this work to completion.

Contents

Appendix

Notation

This chapter introduces the notation and symbols which are used in this thesis.

Numbers and indexing

$\mathbb{N}$	natural numbers
$\mathbb{R}$	real numbers
ϵ_0	small positive real constant
t	discrete points in time or step number
i, j	indexing for objects, measurements and points
c	indexing for classes
u, v	indexing for pixels in image coordinate

Neural Networks

w_i	weight of a neuron
b_i	bias of a neuron
f_{act}	activation function
ϵ	learning rate
α	momentum
μ	arithmetic mean

σ^2	variance
$\mathcal{L}$	loss function
g	gradient from the loss function
ϕ	mapping function of a neural network
θ	network parameters
Q	query matrices in attention mechanism
K	key matrices in attention mechanism
V	value matrices in attention mechanism
d_k	dimension of keys in attention mechanism
λ	thresholds
ω	weighting parameters
A^k	the k-th feature map
M^c	class activation map of the class c
S_c	class score of the class c

Datasets

$\mathcal{D}$	dataset
x	input image
y	label of the image
H	height of the image or kernel
W	width of the image or kernel
N	number of data samples
$\mathcal{C}$	set of classes
N_{p}	number of pixels in the image
N_{bb}	number of bounding boxes in object detection

N_{a}	number of lane anchors
m	binary mask in few-shot semantic segmentation
$\mathcal{U}$	unknown class set in open set object detection
$\mathcal{T}$	class hierarchy
$\mathcal{V}$	set of nodes in the class hierarchy
$\mathcal{E}$	set of edges in the class hierarchy
v	node in the class hierarchy
$\chi(v)$	a function returning a set of node v's child nodes
$\mathcal{K}_c$	the set of classes in the same category as class c
v^r	root node of the class hierarchy
$\mathcal{V}_\mu$	leaf nodes in the class hierarchy

1 Introduction

The main goal of this thesis is to understand the underlying principles behind frequently discussed corner cases in deep learning-based perception systems. Subsequent steps involve leveraging contextual knowledge, which obviates the necessity for manual annotation, to mitigate certain corner cases. Sections 1.1 and 1.2 give an overview of the recent development of data-driven automated driving and the latest advancement in understanding corner cases, while Sec. 1.3 summarizes the main contributions of the work, and Sec. 1.4 outlines the remaining chapters. Although this work primarily focuses on the data-driven perception module for automated driving, the methodologies developed and presented are articulated in a generalized manner to enhance the applicability of the results to a broad spectrum of data-driven applications.

1.1 Data-Driven Automated Driving

In recent years, Advanced Driver Assistance Systems (ADAS) and Automated Driving (AD) functions have progressed significantly. Approximately 30 years ago, the notion of hands-free driving from coast to coast in the USA was merely a dream [Pom96], while today there are already robotaxis operating in certain cities without any safety drivers [Way24, Rob24b], handling challenging rush hour traffic and dynamic traffic scenes. In addition to the urban automated driving functions, highway travel is increasingly augmented by SAE Level 2 systems in many production vehicles [Ebe24, Nor23], where the driver is obligated to monitor the assisted driving functions for lateral and longitudinal control of the vehicle at all times. SAE Level 3 systems [Mer21, Koe23] can take over the driving task under certain conditions and take responsibility to notify the driver promptly for a takeover when necessary.

An exemplary sensor setup of such a vehicle capable of assisted driving can be found in Fig. 1.1. The vehicles with automated driving capability are usually equipped with sensors from multiple modalities to ensure redundancy, including cameras, Radio Detection and Ranging (RaDAR) sensors and Light Detection and Ranging (LiDAR) sensors. In some pre-approved areas, e.g., in parking garages, the automated parking function, supported by installed sensing infrastructure that is able to communicate with the vehicles, can completely take over the task of finding a suitable space and parking the vehicle without a driver present, fulfilling the SAE Level 4 requirements [Rob24a].

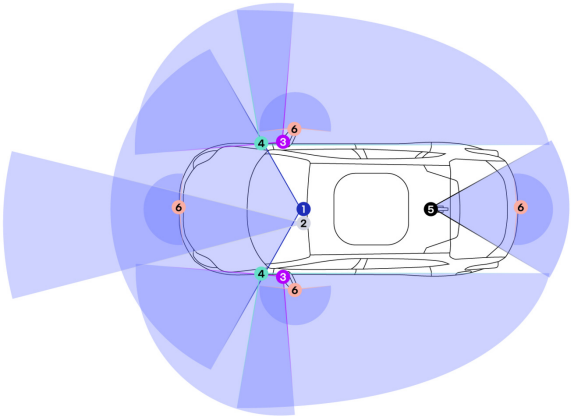


Figure 1.1: Schematic representation of an assisted driving vehicle that relies heavily on visual perception [Mob24]

One significant factor contributing to the success of integrating those functions into production vehicles is the utilization of data-driven methods. While hand-engineered features dominated early perception systems for automated

driving vehicles, ranging from lane detection methods based on color gradients [Can86, Dud72] to occupancy prediction that relies on stixel representations [Pfe10, Fra13], recent work leverages a large amount of real-world data that requires extensive manual annotation for scene understanding. The potential of the data-driven methods was evident in 2015 as the Convolutional Neural Network (CNN) proposed by He et al. [He15] surpassed human experts for image recognition tasks without explicitly relying on conventional feature engineering.

Since then, such data-driven methods have been successfully applied to various perception tasks for automated driving across multiple modalities, including object detection [Zhu21, Ren15, How17, Liu16, Red16, Red17], lane detection [Liu21a, Zhe22, Tab21, Wan22b, Hon24], semantic segmentation [Wan23a, Xia18, Guo22], and instance segmentation [Bol19b, Jai23, Che22]. Furthermore, such a learning-based paradigm has also been utilized for other downstream tasks in the processing pipeline for automated driving, with object tracking [Woj17, Cui22, Zha22d], motion prediction [But17, Zho22c, Shi24], and trajectory planning [Kar24, Boj16, Vit22] empowered by the data-driven solutions. By adopting a modular design of the aforementioned building blocks, those modules are able to replace the rule-based and manually engineered subsystems in assisted and automated driving vehicles. Recent approaches that embrace an end-to-end framework make it possible to mitigate errors across several modules and demonstrate superior effectiveness over conventional modular designs [Hu23b, Sha23, Jae23, Wan25]. This accelerates the development of data-driven automated driving, because further improvements can be achieved simply by scaling the amount of data used for training and validation.

1.2 Corner Cases for Automated Driving

Despite the success of data-driven models for automated driving, there are discussions and concerns regarding the compliance of the data-driven automated driving functions with the established safety regulations and technical standards for automotive software development, e.g., ISO 8800 [ISO24], ISO

26262 [ISO18], and ISO 21448 [ISO22]. Hasterok et al. [Has21] have defined systematic development processes for sophisticated machine learning systems. Furthermore, the quality and variability of the data directly impact the trained deep neural networks, affecting their generalization ability [Gad24, Zho23b]. When neural networks receive input data whose distribution deviates (i.e., is out-of-distribution) from that of the training set (in-distribution), they often become overconfident and produce *abnormally high softmax confidences* in their predictions [Ngu15]. Consequently, the European Artificial Intelligence Act ("AI Act", [Eur21]) that took effect in 2024 imposes stringent requirements on AI-based solutions regarding the coverage of the datasets used for training and validation. As an inherently complex system exposed to dynamic and changing conditions, automated driving functions may still generate unintended actions. A large number of factors may contribute to the undesired behavior of the system, including the reliability of the sensors or actuators, a sudden change in the driving environment, or failure in communication. For instance, deploying the driving function in unseen regions frequently introduces novel traffic participants and unique traffic signs. Despite extensive efforts during the initial development of such ADAS and Highly Automated Driving (HAD) functions, it remains impractical and infeasible to collect and annotate every possible driving scenario worldwide. In addition, changes in mobility concepts and means of transportation, which can frequently be observed over a long-term time horizon, also implicitly shift the distribution of driving surroundings. Such error-prone situations are often regarded as *corner cases* in the development of assisted and automated driving functions.

As required by systematic safety regulations and guidelines, such as ISO 21448, previous work concentrated on classifying and providing taxonomies of corner cases in automated driving [Bol19a, Bre20, Bog21, Hei21], but did not sufficiently address the fact that the corner cases are frequently caused by multiple factors, e.g., hardware properties and data coverage of the training dataset. For instance, performance degradation of a data-driven computer vision model in rainy conditions can be explained by the combined factor caused by domain shifts on the domain level and local outliers impacted by raindrops on the vehicle's windscreen [Bre20]. This leads to ambiguities

concerning when such corner cases are supposed to be mitigated, as some of those observations are inevitable.

Recent advancements in foundation models with sophisticated pretraining on large-scale datasets facilitate the analysis of such corner cases [Rad21, Ram21, Rom22, Ach23, Xia24], where better scalability of the system and automatic data pipeline incorporating video and language pairs require fewer manual interventions during the development process. However, as shown in recent research, such foundation models still have limitations and suffer from hallucination [Liu24, Li23], where the models generate decisions and predictions based on non-existent and non-sensical observations. In such cases, the erroneous data need to be annotated manually, which is considered inherently costly and requires skilled human annotators to execute the task. For instance, Cordts et al. [Cor16] claim that annotating a single image for semantic segmentation takes up to 90 minutes with quality control. Moreover, data exchange across different parties for annotation is subject to data privacy regulations, e.g., the General Data Protection Regulation (GDPR), Personal Information Protection Law (PIPL), or California Consumer Privacy Act (CCPA), that may regulate such exchange process. Consequently, the reliance on costly data annotation can be significantly reduced by developing models and methodologies that require fewer human interventions and annotations, such as utilizing contextual knowledge from specific tasks.

1.3 Contributions

The contributions of this thesis to the field of understanding and mitigating corner cases with contextual knowledge are summarized as follows:

- As a preliminary study, the importance of data quality is demonstrated by comparing various postprocessing methods after data recording for image anonymization. This section is motivated by the fact that Personally Identifiable Information (PII) needs to be stored and analyzed in such a way that the identity of the person cannot be reconstructed anymore, which makes the anonymization of human faces and vehicle number plates mandatory

due to the data protection regulations prior to data exchange. Four frequently used anonymization patterns are compared for semantic segmentation in combination with diverse image encoders, training resolutions, and decoder architectures. The results demonstrate a statistically significant negative impact of image anonymization on segmentation performance. Based on the observation and analysis, an alternative way of anonymization that is able to retain the original data distribution is discussed to mitigate the performance loss. This work was published in the proceedings of the 2022 IEEE Intelligent Vehicles Symposium (IV 2022) [Zho22b].

- A systematic analysis of corner cases is conducted in the context of highly automated driving. As requested by the rigorous regulations and safety requirements for automated driving functions, it is essential not only to identify those critical driving situations, the corner cases, but also to proactively mitigate them during function development by systematic and provable processes. In this work, corner cases are classified according to the interpretation problems of neural networks that lead to errors, independent of the hardware limitations and of risks connected with certain driving decisions. The interpretation problem is related to the data points that are available during training and validation, called the *development dataset*, unveiling the risks behind each status category of data. Furthermore, this work reviews existing solutions that can mitigate the corner cases. This work was published in the proceedings of the 2023 IEEE Intelligent Vehicles Symposium (IV 2023) [Zho23c].
- Critical Error Rate (CER) is proposed as a supplement to existing semantic segmentation metrics. The CER focuses on the error rate, reflecting predictions that fall outside the defined ground truth category and potentially manifest significant implications. With experiments conducted on four mainstream datasets, this work demonstrates the prediction robustness of various image encoders and network architectures measured by the predicted superclass of the model exposed to covariate shift and novel classes not encountered during training, showing the importance of creating and utilizing a high-quality dataset for generalization ability. Last but not least,

the impact of utilizing use-case relevant class hierarchy is discussed, revealing the mechanism of the appearance-based optimization. This work was published in the proceedings of IEEE / CVF Computer Vision and Pattern Recognition Conference 2023 Workshops (CVPRW 2023) [Zho23b].

- An unknown-aware object detection approach is proposed, which incorporates class taxonomy as contextual knowledge integrated into the training of the object detection networks. With the help of a novel classification head, the unknown-aware object detector is able to detect both known classes as well as object classes that were unknown at training time but share object categories with known classes. The proposed network architecture is extensively evaluated using a synthetic dataset and a real-world automated driving dataset. In addition, the impact of different class hierarchies on the model's closed and open set detection performance is discussed. This work was published in the proceedings of the 26th IEEE Conference on Intelligent Transportation Systems (ITSC 2023) [Zho23d].
- Few-Shot Semantic Segmentation (FSSS) with prior knowledge for challenging driving scene understanding tasks is evaluated, where the main objective is to segment a novel class based on limited training samples. Compared to the previous work that merely concentrated on object-centric datasets, with a clear differentiation between foreground and background classes, this work first analyses fundamental differences in properties between commonly used FSSS datasets and the driving datasets. Based on the observations, the integration of the class hierarchy into the meta learning model during training, utilizing more effective feature extractors and support images is evaluated on driving scene understanding datasets to demonstrate the effectiveness. This work was published in the proceedings of the 2024 IEEE Intelligent Vehicles Symposium (IV 2024) [Zho24a].
- A Weakly Supervised Domain Adaptation framework for Lane Detection (WSDAL) is proposed, which requires readily available Number-of-Lanes (NoL) labels from the target domain to aid the domain adaptation process for data-driven lane detection models. The proposed framework leverages the teacher–student network to generate pseudo labels for the lane anchors,

assisted by an auxiliary segmentation head that ensures the generalization ability of the network. The loss function incorporating meta information provides beneficial guidance for arbitrary anchor-based lane detectors during domain adaptation. Validated on a wide range of lane detection datasets covering scenarios from highway driving to urban situations, WSDAL demonstrates its effectiveness over the unsupervised and fully supervised counterparts. Last but not least, extensive ablation studies varying the quality of NoL labels suggest that labeling inaccuracies at realistic scales still yield acceptable adaptation performance. This work was published in the proceedings of IEEE / CVF Computer Vision and Pattern Recognition Conference 2024 Workshops (CVPRW 2024) and won the Best Paper Award of the SAIAD workshop [Zho24b].

1.4 Thesis Outline

As shown in Fig. 1.2, this thesis strives to understand corner cases in the context of assisted and automated driving functions in the first place and mitigate such corner cases with contextual knowledge that does not require explicit manual annotations for the downstream tasks. Chapter 2 introduces the fundamentals of deep learning, data-driven computer vision, and the commonly adopted methodology used for maintaining the data-driven models, which is followed by Ch. 3, which presents the recent advancements in the downstream tasks that are related to this thesis, including the efforts that increase the learning efficiency. As a preliminary study, Ch. 4 contains an initial experiment demonstrating the role of data quality in the performance of semantic segmentation models, where the image data anonymization may adversely affect performance due to the absence of critical data. Inspired by the experiment, the relation between corner cases and available data during the development is further discussed and analyzed in Ch. 5, providing a novel perspective on classifying corner cases for automated driving from the interpretation problem of the data-driven solutions. As manual annotation of the raw data is considered expensive, this thesis emphasizes leveraging contextual knowledge from known data sources to increase learning efficiency.

Chapters 6 and 7, as a central part of this thesis, provide an in-depth analysis of how class hierarchies can be used for out-of-distribution evaluation for semantic segmentation, for unknown-aware object detection, which aims to detect previously unknown object classes within the same category and for few-shot learning to segment a novel class with a limited number of examples. In addition to class hierarchy, Ch. 8 illustrates the usage of other contextual knowledge from the driving environment, in this case, the Number-of-Lanes (NoL), to efficiently adapt the data-driven models to unseen domains.

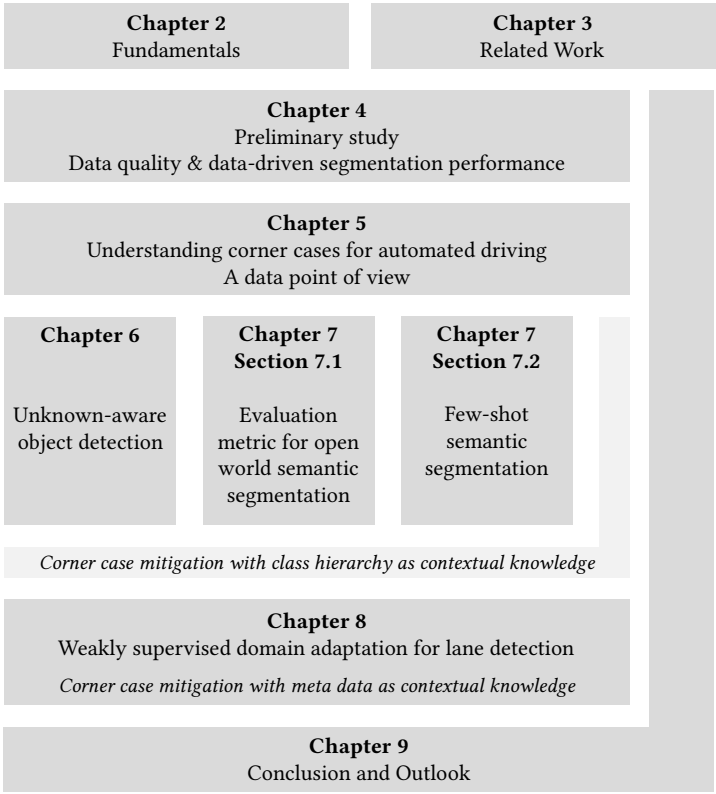


Figure 1.2: Outline of the thesis with chapter and section numbers indicated

2 Fundamentals

The following chapter provides an overview of the fundamental theoretical elements of data-driven assisted and automated driving functions. Section 2.1 briefly introduces the basic mechanism of deep learning, which makes such a learning paradigm possible with large amounts of data, while Sec. 2.2 focuses on two frequently used backbone network types that are specifically widely used in the computer vision domain that are empowered by the introduced data-driven methods. In Sec. 2.3, three major downstream tasks that are relevant for assisted and automated driving functions are introduced with the latest research advancements, and Sec. 2.4 gives an overview of the available datasets that address driving scene understanding and the metrics used for quantitative evaluation.

2.1 Deep Learning

Differentiating from classical data analytics, where handcrafted features play an important role, deep learning learns representations based from large amounts of data. Inspired by the biological nervous system, McCulloch et al. [McC43] introduced the single-layer perceptron system, which is considered as a fundamental building block of modern learning systems. The output y of the neuron is given by $f_{\text{act}}(\sum_i w_i x_i + b)$, where x_i are the inputs, w_i and b are the weights and biases. The output is propagated using a non-linear activation function f_{act}, for instance, Rectified Linear Unit (ReLU). Subsequently combining multiple hidden layers in addition to the input and output layers allows complex non-linear functions to be represented.

In a supervised learning paradigm, the annotated dataset $\mathcal{D}$ that contains N paired training examples $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ is used for the optimization of the network parameters θ. During training, methods are used to derive the estimate of the gradient of the θ to minimize the loss function $\mathcal{L}$. Depending on the learning task, loss functions $\mathcal{L}$ focus mainly on two aspects: Regression and classification. For the regression task, loss functions measure the disparity between the predicted values $\hat{y}_i$ and the ground truth y_i, presented by Mean Squared Error (MSE) and Mean Absolute Error (MAE) shown in Eq. (2.1) and Eq. (2.2).

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{2.1}$$

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{2.2}$$

For the classification task, Cross-Entropy (CE) loss is commonly used for binary and multi-class classification tasks:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log(\hat{y}_{i,c}), \tag{2.3}$$

where $y_{i,c}$ is the one-hot encoded classification vector and $\hat{y}_{i,c}$ is the predicted probability that observation i belongs to class c. The optimization follows the scheme of gradient descent, which calculates the gradient g of the entire dataset $\mathcal{D}$ to update the network weights and biases using backpropagation

$$\theta_t = \theta_{t-1} - \epsilon g, \tag{2.4}$$

where ϵ is the learning rate. Such calculation is considered to be computationally expensive and slow. In addition, the gradient descent also requires significant memory resources, especially when dealing with large datasets. Therefore, Stochastic Gradient Descent (SGD) replaces the gradient g with an expectation $\hat{g}$. The expectation may be approximately estimated using a small

set of samples with a minibatch $\mathcal{B}'$ from the annotated dataset $\mathcal{D}$,

$$\hat{g} = \frac{1}{|\mathcal{B}'|} \sum_{(x_i, y_i) \in \mathcal{B}'} \nabla_\theta \mathcal{L}\left(f(x_i; \theta), y_i\right). \tag{2.5}$$

Besides its advantages in memory usage and training efficiency, SGD introduces a degree of randomness that helps networks avoid settling into local minima. This stochastic nature of SGD enables the exploration of the solution space more thoroughly, increasing the likelihood of discovering a global minimum. Consequently, SGD enhances the model's generalization performance, leading to better generalization on unseen data.

To further accelerate gradient vectors in the direction of optimization, momentum accumulates an exponentially decaying moving average of past gradients to update the parameters using

$$v_t = \alpha v_{t-1} - \epsilon \hat{g} \tag{2.6}$$

$$\theta_t = \theta_{t-1} + v_t, \tag{2.7}$$

where the hyperparameter α determines the decay of previous gradients. Other optimizers like AdaGrad [Duc11], Adam [Kin15] and AdamW [Los19] help to dynamically adapt hyperparameters during training, like the learning rate ϵ.

Due to the fact that the network parameters θ are updated simultaneously during training, the selection of the learning rate for different layers may become non-trivial as high-order interactions need to be considered when the neural network exceeds a certain depth. Therefore, batch normalization reparameterizes activations after any input or hidden layer from the network

$$a' = \gamma \left(\frac{a - \mu}{\sqrt{\sigma^2 + \epsilon_0}} \right) + \beta, \tag{2.8}$$

where a small constant ϵ_0 is added to ensure numerical stability. During training, the mean μ and standard deviation σ are derived from each minibatch of the training dataset. At test time, they are replaced by running averages from

the inference batches. By adding two sets of learnable parameters γ and β, the original activation can be restored. Batch normalization reduces internal covariate shifts and ensures that the means and variances of the activations remain constant, promoting faster convergence and enabling higher learning rates. In addition to batch normalization, there are other normalization methods, including layer normalization [Ba16], group normalization [Wu18], and instance normalization [Uly16].

In order to mitigate the problem of overfitting, regularization methods are frequently used during training like drop-out, which aims to drop random sets of the neural units in the network during training and thereby learning more robust feature representations. Weight decay penalizes large weights in the model to achieve a simpler and more generalizable model.

2.2 Data-Driven Computer Vision

Starting with LeNet, proposed by LeCun et al. [LeC98], computer vision tasks have become increasingly data-driven. One essential milestone was the ILSVRC classification task between 2012 and 2014 [Rus15a], in which convolutional neural networks (CNNs) showed significant performance advantages over solutions based on classical handcrafted features. This section introduces convolutional neural networks, along with transformer architectures, which have shown great success in the Natural Language Processing (NLP) domain, for data-driven computer vision applications.

2.2.1 Convolutional Neural Networks

Linear convolution has been an essential part of the classical computer vision tasks for detecting edges and image smoothing. Mathematically, discrete convolution on a two-dimensional space is defined as:

$$S(u,v) = \sum_{i=1}^{W_k} \sum_{j=1}^{H_k} x(i,j) \cdot k(u-i, v-j) \tag{2.9}$$

$$= \sum_{i=1}^{W_k} \sum_{j=1}^{H_k} x(u-i, v-j) \cdot k(i,j), \tag{2.10}$$

where $x(u,v)$ represents the input image, and $k(i,j)$ denotes the convolutional kernel with spatial dimensions $W_\mathrm{k} \times H_\mathrm{k}$.

In addition, pooling is frequently combined with convolution and activation layers, which aims to condense the network's output at a particular location by summarizing the spatial patches. Pooling creates invariance to small translations in the image, ensuring that minor input shifts do not significantly alter the pooled outputs. This makes it useful for tasks where the presence of a feature is more crucial than its exact position. Such a property thereby enforces a strong prior that the learned function must be translation invariant. Examples of pooling operations include max pooling, which selects the maximum value of the patch, and average pooling, which computes the mean of the values.

Furthermore, downsampling methods, such as stride, are frequently implemented to calculate the full convolution at intervals. Stride refers to the number of elements skipped. Another critical aspect is padding, which prevents the representation width from shrinking at the boundary of each layer. Figure 2.1 illustrates an example of convolution and pooling on an image.

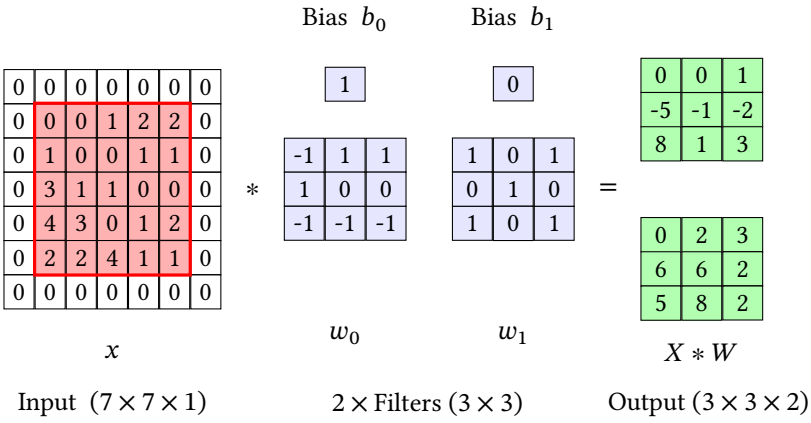


Figure 2.1: An example of convolutional layer with a input size of 5×5. The convolutional layer has two filters of size 3×3 with a stride of 2 and padding of 1.

In comparison to fully connected layers, convolution layers – encompassing core operations such as convolutions, stride, padding, and activation functions – demonstrate their advantages with three essential principles [Goo16]:

1 **Sparse interactions** reduce the number of parameters as well as the needed computational resources. Convolutional networks employ kernels that are smaller than the input images.

2 **Parameter sharing** enhances efficiency by reusing the same set of parameters across different input regions. This can be explained by the fact that some visual features like edges are universally applicable for understanding the images.

3 **Equivariant representation** is advantageous when detecting consistent features across different locations, especially, the same set of parameters applies uniformly across the entire input. Translation in the input images leads to the same output shift from the convolutional neural networks.

Starting from AlexNet [Kri12], the subsequent work of CNN concentrates on utilizing deeper convolutional layers with relatively small receptive fields to

reduce the number of parameters, like VGGNet [Sim15]. With Inception-v2 [Sze16], normalization methods like batch normalization are introduced in the network architecture. In 2015, He et al. [He16] introduced ResNet with residual blocks to address the vanishing gradient problem via skip connections. The proposed network architecture contains configurations with various numbers of layers, containing 18-layer, 50-layer, and 101-layer configurations, which are used in this thesis. Geirhos et al. [Gei19] exploited the split-transform-merge strategy for the ResNext architecture, allowing multiple sets of transformations to be aggregated based on ResNet. Research also shows that CNNs are texture-biased and have a strong inductive bias with a hierarchical structure [Zei14, Gei19].

2.2.2 Vision Transformer

Transformers, originally designed for natural language processing [Vas17, Dev19, Bro20], have been adapted for various computer vision tasks [Dos21, Liu21b, Xie21]. The architecture of CNNs shows that it is challenging to capture long-range feature relations, making them less effective for long sequences.

To overcome this limitation, the central component of transformer models is the self-attention mechanism [Vas17], which enables each element of an input sequence to attend to other elements from the sequence. Self-attention computes a set of attention weights denoting the importance of other elements. The attention mechanism is given by:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{2.11}$$

where Q, K, and V denote the query, key, and value matrices, respectively, and d_k is the dimension of the keys.

Specifically, for the Vision Transformers (ViTs), the input image is split into patches, and their relative position is fed into the self-attention modules as position embeddings [Dos21]. ViTs exhibit a weaker image-specific

inductive bias compared to CNNs. In CNNs, properties such as locality, two-dimensional neighborhood structures, and translation equivariance are inherently embedded within each layer throughout the entire architecture. Such neighborhood structure is minimally utilized in the ViT, initially when the image is segmented into patches and subsequently during fine-tuning process for modifying the position embeddings corresponding to images of differing resolutions. Conversely, in the Vision Transformer, only the MLP layers exhibit local and translationally equivariant characteristics, whereas the self-attention layers operate on a global scale. Due to the reduced inductive bias, ViT models are more data-hungry than CNNs. They show their advantages only when trained on a large-scale dataset, for instance, the JFT300M [Tou21]. Recent advancements have concentrated on integrating convolutional layers with transformer encoders for hybrid models, modifying the architecture to cope with tasks that require dense prediction, e.g., semantic segmentation, and training data-efficient image transformers [dAs21, Has23].

Recent research also suggests that transformers exhibit greater resilience than CNNs when exposed to out-of-distribution samples [Zha22a, Li22a] or adversarial attacks [Nas21, Pau22]. However, if a similar training setup is applied, transformers have not been shown to be more robust than CNNs, although their operating mechanisms are vastly dissimilar with different focuses [Bai21, Wan23c, Nas21]. There are CNN-based image encoder architectures that are heavily inspired by the vision transformers, like ConvNeXt [Liu22b] and SegNeXt [Guo22], featuring modified block setups, a patchified stem, large kernel sizes and different normalization layers. With approaches like shifted window, Swin transformer is able to perform efficient local self-attention that scales linearly with the size of the input image [Liu21b].

2.3 Perception Tasks for Automated Driving

This section delineates three pivotal perception tasks for assisted and automated driving functions: Object detection, lane detection, and semantic

segmentation. For each of these tasks, two key facets are explored: An abstract description of the task and a comprehensive summary of the state-of-the-art research relevant to that task. In this thesis, RGB images, denoted $x \in \mathbb{R}^{H \times W \times 3}$, where H and W are the height and width of the RGB image, are used as input of the neural networks. Paired images $x \in \mathcal{X}$ and their corresponding ground truth annotations $y \in \mathcal{Y}$ are used to learn the mapping functions ϕ that contain the parameters θ of each neural network.

2.3.1 Object Detection

Problem Description

Object detection models aim to localize and classify objects of interest given an RGB image input. The primary task is to learn parameters θ for a mapping function ϕ_θ that predicts bounding boxes and class labels for objects from a predefined closed set of classes in the image.

Let $y = \{b_1, b_2, \dots, b_{N_{\text{bb}}}\}$ be the set of bounding boxes with length $|y| = N_{\text{bb}}$, where each bounding box b_i is represented by two essential elements: A class label c and a geometric vector l that contains the horizontal and vertical coordinate of the bounding box center and the height and width of the object, shown as $l_i = (u_i, v_i, w_i, h_i)$. To simplify the problem, it is assumed that the prediction $\hat{y}$ and the ground truth y are associated. The learning objective can be formulated as a combination of localization that can be regarded as a regression task using L_1 or L_2 distance and classification loss that can be trained with cross-entropy loss:

$$\mathcal{L}_{\text{od}}(y, \hat{y}) = \sum_i \left[\mathcal{L}_{\text{cls}}(c_i, \hat{c}_i) + \omega_{\text{OD}} \mathcal{L}_{\text{loc}}(l_i, \hat{l}_i) \right], \tag{2.12}$$

where ω_{OD} is a hyperparameter that balances the contribution of localization and classification losses.

Current Research

Three major learning paradigms are used in the development of object detection models. Presented by the Faster R-CNN model [Ren15], the object detection task is divided into two subtasks, known as two-stage models, containing object localization using a Region Proposal Network (RPN) and the classification head. RPN outputs a series of object proposals containing objectness scores based on predefined anchor boxes, which are the input for the second step, where the localization and classification of the object instances are finalized. While two-stage approaches demonstrate strong performance, their prolonged inference time is regarded as a key limitation. To address this, one-stage models, e.g., YOLO [Red16, Red17, Red18, Boc20] and Single Shot Detector (SSD) [Liu16], are proposed to abstract such detection systems within a single convolutional neural network based on predefined grids or anchor boxes that provide tradeoffs between inference speed and network performance. With the help of compact, lightweight backbones [How17, San18, How19], the model is able to run considerably faster to ensure edge processing. It is worth noting that the single- and two-stage detectors still rely on manually crafted operations like Non-Maximum-Suppression to mitigate overlapping predictions about the same object. In addition, hyperparameters like detection thresholds and the number of anchors during training also play an essential role in the performance of the model.

In comparison, the latest transformer-based models like Detection Transformer (DETR) usually comprise a CNN backbone to extract features from the input image, a transformer encoder-decoder module with parallel decoding, and MLP-based prediction heads [Car20]. As introduced in Sec. 2.2, the extracted feature representations are fed into the transformer encoder-decoder architecture for global correlation of the image parts. The feed-forward network utilizes decoder outputs to directly produce detection of object instances that can be trained end-to-end and do not rely on manually crafted hyperparameters. Models like Deformable DETR further developed the transformer-based decoder architecture by improving spatial attention in the transformer modules [Zhu21], which improved convergence speed, computational efficiency, and performance on smaller objects.

2.3.2 Semantic Segmentation

Problem Description

For each pixel in a given image x, one class c from the predefined object class set $\mathcal{C}$ is assigned in the task of semantic segmentation. The task aims to learn a posterior probability mapping $\phi_\theta : \mathcal{X} = \mathbb{R}^{H\times W\times 3} \mapsto \mathcal{Y} = \mathbb{R}^{H\times W\times|\mathcal{C}|}$. For a total of N_p pixels in the image, where $N_p = H \times W$, a cross-entropy loss is calculated as:

$$\mathcal{L}_{\mathrm{CE}}(y,\hat{y}) = -\frac{1}{N_p}\sum_{i=1}^{N_p}\sum_{c=1}^{|\mathcal{C}|} y_{i,c}\log(\hat{y}_{i,c}). \tag{2.13}$$

Current Research

Since the introduction of the Fully Convolutional Network (FCN) [Lon15], CNNs have achieved significant success, becoming the preferred architecture for semantic segmentation. In the deep learning era, segmentation model architectures are typically bifurcated into encoder and decoder components. It is common practice for encoders to employ established classification networks, such as ResNet, ResNeXt [Xie17], and DenseNet [Hua17], rather than designing bespoke architectures, as introduced in Sec. 2.2. Nevertheless, semantic segmentation involves dense prediction tasks, distinguishing it from image classification. Therefore, improvements observed in classification networks may not directly translate to the more complex segmentation tasks. Consequently, specialized decoders have emerged, like the Atrous Spatial Pyramid Pooling (ASPP) head from DeepLabv3Plus (DLv3+) [Che18], the Unified Perceptual Parsing Network (UPerNet) decoder head [Xia18], HRNet [Wan20b], SegFormer [Xie21], and HRFormer [Yua21]. Working in tandem with encoders, the decoder component plays a crucial role in refining segmentation outcomes. Various decoders are designed to meet distinct objectives, including achieving and expanding multi-scale receptive fields, aggregating multi-scale semantic information, enhancing edge features, and capturing global context. Recently, methods based on transformers have

demonstrated substantial potential, surpassing the performance of traditional CNN-based approaches [Liu21b, Xie21, Yua21].

2.3.3 Lane Detection

Problem Description

Anchor-based lane detection aims to achieve the task of taking the input image x for the mapping ϕ_θ to the lane prediction $y \in \mathbb{R}^{N_a \times L}$, where N_a denotes the number of lane anchors, *i.e.*, the maximum number of possible lane predictions, and L the dimension of the lane description vector $l = [l^{(1)}, \ldots, l^{(L)}]$. Here, l contains $L-2$ elements $l^{(1)} \ldots l^{(L-2)}$ describing the lane position as coordinates, and two elements $l^{(L-1)}$, $l^{(L)}$ for "foreground" and "background" respectively, indicating whether the particular lane likely exists or not.

Current Research

Before deep learning methods were applied, lane detection relied heavily on handcrafted features [Low04, Can86], which contain pre-processing, feature extraction, and curve fitting [Tan21, Zha22e]. Pre-processing is used to remove undesired interference from the image [Cha14, Yan18, Yan19]. Edge detection algorithms were then applied for lane feature extraction, such as Sobel [Mu14], Canny [Wu14], SIFT [Yan19], and Hough transform [Niu16]. In the era of data-driven lane detection, the most commonly used models can be categorized into *segmentation-based* and *anchor-based* detectors. The vanilla approach treats lane detection as a segmentation task. With LaneNet [Nev18], the lane detection task was accomplished by utilizing two lane segmentation branches for foreground segmentation and identifying the lane instances. Then, a transformation matrix projected the segmentation maps into the bird's eye view. Further improvements like SCNN [Pan18b] and RESA [Zhe21] utilized spatial CNNs and added multi-stride connections, facilitating perception of continuous structures. There are also experiments [Xu20] that leveraged AutoML to explore and optimize the model structure for lane detection models. Liu et al. [Liu20a] utilized a style transfer model to enrich the training set

and introduced a second branch that predicts the existence of a lane based on pixel-level ground truth.

Line-CNN [Li20a] introduces the line anchor in lane detection. Similar to box anchors in object detection, anchors extract feature representations to constitute anchor features, which are used for final prediction. LaneATT [Tab21] extracts local feature maps to form anchor local features. With the help of an attention mechanism, a global feature vector is aggregated to deal with the sparse property of the lane detection task. SGNet [Su21] utilizes prior information like vanishing point and lane information to improve performance. As a row anchor-based method [Qin20], it predicts the probable cell for each predefined row on images. CondLaneNet [Liu21a] further improves row anchor-based lane detection by learning a heatmap of possible starting points of the lanes as prior information. Similar to many two-stage object detection models, CLRNet [Zhe22] adopts learnable anchor parameters for the regression tasks and the detection head makes predictions for each tensor. CLRerNet [Hon24] further improves the performance by optimizing the similarity cost to align the confidence score with the IoU score.

2.4 Datasets and Evaluation

2.4.1 Datasets for Driving Scene Understanding

Cityscapes: Cityscapes [Cor16] is one of the first semantic segmentation datasets for urban scene understanding. The dataset contains 25 000 images captured from European cities in clear weather conditions during the daytime. Frequently, 19 classes are included in the annotation emphasizing static and dynamic objects in the city.

ACDC: ACDC [Sak21] shares the annotation space with the Cityscapes dataset but was recorded in Switzerland under challenging conditions like fog, night, rain, and snow. The dataset with 4 006 images is typically used to evaluate the robustness of the models, domain adaptation, and generalization ability.

BDD100k: BDD100k [Yu20] contains 100 000 images with various annotations, recorded in diverse weather and lighting conditions across the United States, covering different road types. For the task of semantic segmentation, there are 10 000 annotated frames. The dataset shares the annotation policy with the Cityscapes dataset.

A2D2: A2D2 [Gey20] is a diverse dataset from Audi with semantic segmentation, object detection, and lane detection annotations that were recorded in several German cities. The dataset also provides high-quality data annotation from multiple modalities. For semantic segmentation, 41 280 frames are annotated with high label granularity, which makes the comparison between different datasets possible.

Mapillary: Mapillary [Neu17] includes 25 000 images from diverse training situations in various weather and lighting conditions around the world. According to the annotation policy (v1.2), 66 classes are available from 9 categories for semantic segmentation.

nuImages: nuImages [Cae20] contains driving scenes in diverse conditions, primarily from locations in Boston, USA, and Singapore. A total of 93 000 images are selected based on the focus on rare classes. The annotated images include rain, snow, and nighttime. In addition, the dataset provides past and future camera images of the annotated frames for the investigation of temporal dynamics. The class hierarchies of the aforementioned datasets can be found in Ch. B.

TuSimple: TuSimple [TuS17] is a lane dataset focusing on highway scenarios collected in the United States containing around 6 000 images. The dataset is relatively homogeneous as the data were collected under good and moderate weather conditions and only covers highways.

CULane: CULane [Pan18b] is a large-scale lane dataset that contains diverse driving situations in Beijing with 133 235 frames. In addition to normal scenarios, there are eight challenging subsets, including traffic jams, nights, and intersections, which account for 72.3% of the entire dataset. When lanes are

occluded, the labels are assigned manually by the human annotator according to the context.

CurveLanes: CurveLanes [Xu20] is a large-scale lane detection dataset with 150 000 images for complex driving scenarios and up to 14 lanes per image. Like the CULane dataset, it was collected in China with high urban bias. Besides that, this dataset also concentrates on curved lane markings that differ from the natural curvature distribution from other datasets.

2.4.2 Evaluation Metrics

Evaluation metrics are essential for quantitative comparison of the model performance. This section elaborates on the evaluation metrics employed in the three computer vision tasks: object detection, semantic segmentation, and lane detection.

Despite differences in the tasks, these evaluation metrics share some common properties and definitions of base elements. Starting with multi-class image classification, each element $M_{i,j}$ from the confusion matrix M, which has a dimension of $|\mathcal{C}|\times|\mathcal{C}|$, represents the number of predictions of the class-i from the neural network that belong to the ground truth of class-j. Therefore, the number of True Positive (TP) predictions for class c is given by $M_{c,c}$, and the number of False Positive (FP) predictions is defined by:

$$\mathrm{FP}_c = \sum_{i=1}^{|\mathcal{C}|} M_{c,i} - \mathrm{TP}_c. \tag{2.14}$$

Similarly, the number of False Negative (FN) predictions is defined by:

$$\mathrm{FN}_c = \sum_{i=1}^{|\mathcal{C}|} M_{i,c} - \mathrm{TP}_c. \tag{2.15}$$

To measure the proportion of true positive predictions among the total number of positive predictions made by the model, precision is frequently used

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{2.16}$$

Recall, on the other hand, measures the proportion of true positive predictions among all actual positive instances:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{2.17}$$

To combine the two metrics, in the binary classification task, the F_1-score is also frequently used:

$$\text{F}_1 = \frac{2}{\text{Precision}^{-1} + \text{Recall}^{-1}}. \tag{2.18}$$

Detection Tasks

As object detection aims first to locate and then identify object instances within an image, it involves the localization of the proposed bounding boxes in addition to the classification of the object. In order to measure the overlapping area of the ground truth and prediction bounding boxes, Intersection over Union (IoU) is frequently used:

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}. \tag{2.19}$$

Based on various IoU thresholds, an evaluation scheme from the COCO dataset for object detection is usually adopted. The average precision value of ten different IoU scores from 0.5 to 0.95 with a step size 0.05 is calculated. This is frequently noted as AP@[.5 : .95] or Average Precision (AP). Similarly, such IoU thresholds can also be applied to Recall values that lead to the Average Recall (AR). The mean Average Precision (mAP) is the averaged AP over all

the object classes, noted as

$$\text{mAP} = \frac{1}{|\mathcal{C}|} \sum_{c=1}^{|\mathcal{C}|} \text{AP}_c. \tag{2.20}$$

In other datasets, e.g., Pascal VOC, AP is calculated as the area under the interpolated precision-recall curve based on different recall levels in an ascending order.

Segmentation Tasks

Semantic segmentation can be seen as an image classification task on each pixel of the image. The evaluation scheme is similar to the image classification task that can be evaluated by accuracy, which is defined as the ratio of correctly classified pixels to the total number of pixels:

$$\text{Accuracy} = \frac{\sum_{c \in \mathcal{C}} M_{c,c}}{\sum_{i=1}^{|\mathcal{C}|} \sum_{j=1}^{|\mathcal{C}|} M_{i,j}}. \tag{2.21}$$

However, accuracy treats all classes equally and does not account for the different distributions of classes. A model might achieve high accuracy through high performance on the dominant classes in the images, e.g., *Sky*, while neglecting small objects and rare classes.

IoU further addresses the false positive predictions by calculating

$$\text{IoU}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c + \text{FN}_c} \tag{2.22}$$

of each class c. The mean IoU is the average IoU value of all the available classes

$$\text{mIoU} = \frac{1}{|\mathcal{C}|} \sum_{c=1}^{|\mathcal{C}|} \text{IoU}_c. \tag{2.23}$$

Therefore, the metric IoU is widely used to evaluate semantic segmentation models across all datasets, e.g., Cityscapes, Mapillary, and A2D2.

Lane Detection

Lane detection can be regarded both as a detection task, with true positives identified based on a predefined IoU threshold, and as a segmentation task, where lane markings are annotated at the pixel level. The concrete evaluation scheme is defined, therefore, based on the datasets.

For the TuSimple dataset, the quality of lane detection is determined by evaluating sampled points with given vertical distances from the image (v-direction). A lane point prediction is deemed accurate if the horizontal distance (u-direction) between the ground truth and predicted lane points is less than 20 pixels. A True positive is recorded for a lane prediction if over 85% of the lane points are correctly predicted. Metrics such as False Positive Rate (FPR), False Negative Rate (FNR), and accuracy are complemented by reporting the F1-score.

For the CULane dataset, a prediction qualifies as a true positive when dilated lane detections have an IoU above a predefined threshold with the ground truth. F1-scores for individual subsets, showing specific driving situations, as well as the entire dataset are reported. Notably, according to the annotation policy, the false positive count is reported for the *cross* subset as there should be no lane markings at junctions. Similarly, such an evaluation scheme is also utilized in the CurveLanes dataset, where F1-scores, Precision, and Recall values are calculated based on predefined IoU thresholds.

3 Related Work

This chapter aims to introduce related work addressing the corner cases from three major aspects. Section 3.1 gives an overview of defining and interpreting corner cases in the context of assisted and automated driving functions, followed by Sec. 3.2, where three aspects of robust scene interpretation are discussed: Disturbances from the training datasets, the appearance of novel object instances, and understanding the reasoning behind network predictions. Finally, Sec. 3.3 elaborates on data-efficient methods that are used for adapting existing data-driven methods to deal with previously unknown situations, e.g., domain adaptation, weakly-supervised learning, utilizing class hierarchy, and few-shot segmentation.

3.1 Corner Cases in assisted and automated driving functions

Due to the nature of closed set setup that was introduced in Sec. 2.3, the long-tail problem is considered to be closely related to corner cases, where the difficulty of detecting and interpreting rare or infrequent objects and scenarios is addressed. For instance, while common entities such as cars and pedestrians are abundantly represented in training datasets, many critical yet scarce cases, including atypical vehicle classes, uncommon weather conditions, and seldom encountered road signs, are underrepresented. This class imbalance hampers the performance of detection and segmentation models on those rare instances, which are essential for ensuring safe automated driving functions.

Bolte et al. [Bol19a] started the discussion for the enhancement and further development of automated driving functions through the detection of corner cases, which were defined as *a non-predictable relevant object/class in a relevant location*. It is claimed that such corner cases must be technically unpredictable for certain classes. Based on an ontology, Breitenstein et al. [Bre20] proposed a systematical definition regarding corner cases, which are classified into five main categories sorted by complexity, from *pixel level* that addresses the sensor level to *scenario level* that addresses specific patterns across multiple frames. Similarly, Bogdoll et al. [Bog21] characterized that the corner cases occur when the machine learning model reaches its limit due to epistemic uncertainty. Heidecker et al. [Hei21] further expanded the definition from Breitenstein et al. [Bre20], encompassing additional sensing modalities from RaDAR and LiDAR. They classified corner cases into four categories: Sensor layer, content layer, temporal layer, and method layer. Pfeil et al. [Pfe22] stated that the corner cases are *atypical, rare scenarios that are of high interest* and segmented corner cases based on their causes into *environmental factors*, *functional constraints*, and *system-internal conditions*. The aforementioned research emphasizes the technical relation from corner cases to outliers, anomalies, and novelties.

The interpretation of corner cases is frequently contingent upon specific use cases in the context of automated driving functions. *Probability of Occurrence* is frequently used to determine whether a particular incident is considered a corner case. For instance, Ries et al. [Rie19] categorized rare driving scenarios as corner cases. Kishore et al. [Kis21] treated corner cases similarly to long-tail problems, identifying underrepresented incidents and generating synthetic data via imitation learning methods. Similar to Heidecker et al. [Hei21], Möller et al. [Möl21] deemed corner cases as rare but predominantly highly critical. They also leveraged the detection of erroneous outputs from neural networks to indicate corner cases.

Chou et al. [Cho18] evaluated corner cases in the context of trajectory generation. They defined corner cases as events where input signals result in trajectories violating safety regulations but are avoidable beforehand. Pei et

al. [Pei17] treated the erroneous driving behavior of neural networks for end-to-end driving as corner cases. Ouyang et al. [Ouy21] defined corner cases as data points where the output of a deep learning function is easily affected by perturbations in the surrounding area.

3.2 Robust Scene Interpretation

Building on the understanding of corner cases in the context of automated driving functions, this section addresses the challenge of achieving robust scene interpretation by investigating three interrelated research areas. First, the generalization capability of deep neural networks is examined, aiming to reduce performance degradation on previously unseen scenes. Open set object detection aims to enable models to recognize and appropriately handle classes that are sparsely represented or entirely absent from the training distribution. Furthermore, model interpretability techniques are introduced to promote a more transparent decision making process, thereby enabling the assessment of the trustworthiness of model outputs.

3.2.1 Robustness of Neural Networks

The robustness of the neural networks is examined through two distinct, decoupled dimensions. First, the methodological design of the models that contributes to the system's stability under perturbations is examined. In addition, the frequently overlooked characteristics of the training data are introduced and addressed, which directly influence the model's capacity to generalize.

Methodology

It is shown that achieving high accuracy in training datasets while ensuring substantial invariance to perturbations during inference remains challenging [Tao20, Sze14, Gei18]. For computer vision tasks, the presence of input image corruptions notably degrades the performance of neural networks, subsequently revealing vulnerabilities that may be exploited for adversarial

attacks [Hen19, Kam20, Yan23a]. This aspect, however, is beyond the scope of the present thesis.

For robustness evaluation, models are often trained on a source dataset without disturbances and subsequently tested in a zero-shot manner on other datasets containing corrupted images or out-of-distribution samples without any adaptation [Hen19, Hen21]. Recent studies indicate that specific architectural properties affect network robustness [Kam20], implying optimal network performance is contingent upon noise- and blur-free input images. Similar observations have been made in contexts such as LiDAR detection [Hah22], navigation tasks [Liu21c], and semantic segmentation [Yan23a]. As introduced in Sec. 2.2, models based on transformer architectures have demonstrated state-of-the-art performance across various machine learning domains [Bai21, Pau22], coupled with enhanced robustness against sensor artifacts [Bho21].

On the other hand, the robustness of neural networks is related to learning generalized feature representations from the source dataset, which enables the handling of varying data distributions during inference, called Domain Generalization (DG). Current research of DG predominantly focuses on the task of image classification [Shu21, Ghi15], with work that concentrates on semantic segmentation [Pan18a, Cho21, Pan19, Pen21, Che20a, Hua21a]. To fortify against disturbances in the feature representation, generally, methods such as advanced data augmentation strategies [Xie20, Yan22, Yun19, Wan22a], large-scale supervised and unsupervised pretraining [Che20b, Rad21, Rus15a, Zho22a] can be leveraged. Despite the fact that there are differences in the DG methodologies for various downstream tasks, they fundamentally share core principles.

Domain generalization for semantic segmentation can be broadly categorized into three parts: learning normalized features, domain randomization, and knowledge distillation methods. Pan et al. [Pan18a] incorporated dual normalization layers to ensure feature invariance to appearance changes. The efficacy of feature normalization is further enhanced through domain-variant

frequency analysis assistance [Hua21a]. Other approaches involve normalizing global features by eliminating style-specific information [Cho21] or utilizing alternative normalization layers [Uly17]. Data augmentation strategies generate additional training samples through style transfer models to mitigate domain-specific biases [Pen21]. Chen et al. [Che20a] proposed a contrastive learning pipeline combined with knowledge distillation to achieve better generalization.

Dataset

An often overlooked aspect of the robustness of neural networks concerns training data variability [Din19]. Recent research highlights the potential to enhance the robustness and generalization capabilities when the models are trained on large-scale cross-domain composite datasets [Xia22, Kir23]. Despite the intuitive appeal of real-to-real setups for practical applications, a critical prerequisite for multi-dataset learning is the establishment of a unified label space, which, as demonstrated by MSeg [Lam20] and Kim et al. [Kim22], is complex. Compared to the sim-to-real setup, where synthetic datasets like SYNTHIA [Ros16] or GTA5 [Ric16] share the annotation policies with real-world datasets like Cityscapes, DG is rarely evaluated using datasets that are not explicitly aligned to share the label space, as the evaluation schemes primarily rely on metrics such as class-wise accuracy and Intersection over Union that assume the existence of true positive predictions.

In addition, despite the trivial hypothesis that the quality of the data should have a corresponding impact on the robustness of the trained neural networks, the impact of data anonymization on computer vision tasks was not sufficiently evaluated in the previous work. Geyer et al. [Gey20] explored the implications of image data anonymization as part of the A2D2 dataset evaluation. They showed that using Pyramid Scene Parsing Network (PSPNet) with a ResNet-101 backbone revealed only marginal performance differences caused by anonymization, which was confined to a single network architecture. Similarly, Yang et al. [Yan21b] examined the effects of face obfuscation through various Gaussian blurring configurations, demonstrating that image blurring

only slightly affects accuracy across several computer vision tasks, including object recognition and detection using the ILSVRC dataset [Rus15b]. Their findings indicated a correlation between classification difficulty and the extent of blurring in the training data, with objects having a higher overlap area with human faces leading to more errors.

Due to the limitations of the previous work, Chs. 4 and 7 will further address the challenge caused by the dataset and streamline real-to-real domain generalization assessment.

3.2.2 Open Set Object Detection

Dhamija et al. [Dha20] introduced the first open set object detection protocol, where the training class set $\mathcal{C}_{\text{train}}$ is a subset of the validation set $\mathcal{C}_{\text{val}}$, highlighting that models trained under a closed set assumption tend to erroneously classify unknown objects as known class instances with high confidence. Hosoya et al. [Hos22] demonstrated that simple modifications after the softmax function of an existing closed set object detection model can effectively transform it into an open set object detector. To address the open world object detection challenge, Joseph et al. [Jos21] introduced the Open World Object Detector (ORE) framework [Jos21], which differentiates between known and unknown objects by employing contrastive clustering in the feature space and subsequently utilizing an energy-based method for unknown identification. In order to differentiate from the unknown classes, ORE necessitates a held-out validation set of unknown objects to estimate the distribution by leveraging weak supervision. Additionally, representing all possible unknown objects using one single prototype vector fails to adequately represent the diverse intra-class variations from the unknown classes.

Gupta et al. [Gup22] proposed the Vision Transformer-based multi-scale context-aware detection model, Open World Detection Transformer (OW-DETR). This model incorporates an attention-driven pseudo-labeling mechanism, an objectness branch, and a novelty classifier to detect both known and unknown classes. In addition, Du et al. [Du22b, Du22a] explored various model regularization methods to enhance out-of-distribution object detection.

The evaluation of the open set object detection is two-fold, concentrating on detecting the known (*closed set*) and unknown (*open set*) object classes. For the known classes, noted as $\mathcal{C}_{\text{train}}$, as introduced in Sec. 2.4.2, the commonly utilized metrics of object detection, mAP, and AR are reported [Dha20, Jos21, Gup22].

Given that the open set definition prescribes the unknown class set, noted as $\mathcal{U}$, as infinite, Hosoya et al. [Hos22] introduced the OSOD-III setting, where unknown classes share at least one superclass with a known class considered during evaluation. Consequently, the unknown mean Average Precision and unknown Average Recall can be assessed over a subset of unknown classes that conform to the OSOD-III alignment. Additionally, two major metrics are reported to provide a comprehensive analysis of the performance of those open set object detection models: The Wilderness Impact (WI) [Dha20] and the Absolute Open Set Error (A-OSE) [Mil18].

WI is defined as the ratio between the precision (P) when the detection model is evaluated on the closed set (P_{cs}) and on the open set (P_{os}):

$$\text{WI} = \frac{\text{P}_{\text{cs}}}{\text{P}_{\text{os}}} - 1. \tag{3.1}$$

For an ideal open set detection model, WI should be 0, indicating that the model's performance on the known class set remains unaffected by the inclusion of objects from the unknown class set during evaluation. Following the implementation of Dhamija et al. [Dha20], this thesis adopts the Wilderness Impact $\text{WI}_{\mathcal{R}_l}$ computed across recall levels $\mathcal{R} = \{0.1, 0.2, .., 0.9\}$ with a step size of 0.1. The overall Wilderness Impact is then calculated as the mean value across all recall levels:

$$\text{WI} = \frac{1}{|\mathcal{R}|} \sum_{l=1}^{|\mathcal{R}|} \text{WI}_l. \tag{3.2}$$

In addition to the WI, A-OSE quantifies the absolute number of detections where the ground-truth class belongs to $\mathcal{U}$ but is misclassified as a known

class in $\mathcal{C}_{\text{train}}$. This situation arises when a predicted detection has a bounding box with an IoU greater than 0.5 compared to a ground-truth bounding box.

3.2.3 GradCAM and Uncertainty Estimation

To increase the interpretability of deep learning models, Zhou et al. [Zho16] presented a methodology for computing class-discriminative activation maps by introducing a novel set of weights to provide insight into which regions of an input image significantly influence the predictions. The proposed method requires the weights to be individually trained for each feature map. The Class Activation Map (CAM) is defined by:

$$M_{ij}^{c} = \sum_{k} w_{k}^{c} A_{ij}^{k}, \tag{3.3}$$

where A^k is the k-th feature map and the weights w_k^c are independently trained using global average pooling over individual feature maps. The class score S_c is obtained by summing all the class activation maps:

$$S_c = \sum_{i} \sum_{j} M_{ij}^{c}. \tag{3.4}$$

Selvaraju et al. [Sel17] extended this work with the introduction of GradCAM, which generalizes the original CAM implementation by eliminating the need for training a new set of weights. Instead, the weight w_k^c for a specific class c can be calculated as:

$$w_{k}^{c} = \sum_{i} \sum_{j} \frac{\partial S_c}{\partial A_{ij}^{k}}. \tag{3.5}$$

To assess the uncertainty in network outputs, the entropy maps of the neural network's output class probability vectors $\hat{y}_\theta$ based on the set of parameters θ are visualized. The entropy, combined with the softmax function at the output

layer, which serves as a measure of uncertainty, can be computed using:

$$H(\hat{y}_\theta) = -\sum_{i=1}^{|\mathcal{C}|} \hat{y}_{\theta,i} \log \hat{y}_{\theta,i}. \tag{3.6}$$

The summation encompasses all classes from $\mathcal{C}$ to provide insights into the overall uncertainty in classification predictions.

3.3 Data-Efficient Perception Models

This section introduces four major aspects of leveraging easy-to-obtain or limited data for the training of perception models. These approaches aim to reduce the necessary effort associated with manual annotation, which is prevalent in the supervised learning paradigm.

3.3.1 Weakly-Supervised Learning

Deviated from the fully supervised learning, where the training dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ is available, weakly-supervised learning utilizes $\mathcal{D}_{\text{weak}} = \{(x_i, y_i^{\text{weak}})\}_{i=1}^N$ for training. y^{weak} represents a more abstract form of y, which can be incomplete, inexact, or inaccurate, for instance, it may be generated from heuristic rules or external resources that are easier to obtain compared with y.

Weakly supervised learning is used in several applications including object localization [Gao18, Sin18], object detection [Zha22b, Wan18], and semantic segmentation [Joo17, Wei17, Dai15, Lin16]. One of the major contributing supervision cues for the weak supervision is the Class Activation Map (CAM) [Zho16, Sel17], as these CAM-generated discriminative regions for prediction reasoning can provide additional information for the images, especially within the task of semantic segmentation [Joo17, Wei17]. Generally, sparse annotations are also frequently employed as weak labels, including image-level labels in object detection tasks, or non-exhaustive annotations,

e.g., by annotating a small amount of pixels in segmentation tasks [Bea16]. Gao et al. [Gao18] proposed a method that leverages the number of objects as a supervision signal for object localization. Similarly, in segmentation tasks, leveraging the given bounding boxes to refine and create unsupervised region proposals can enhance the segmentation performance without pixel-level annotations [Dai15, Kho17, Son19]. Joon Oh et al. [Joo17], on the other hand, combined external saliency masks with image-level labels for training, while Scribble [Lin16] is utilized to propagate category information to unlabeled pixels, serving as a cost-effective annotation solution that is beneficial for segmentation tasks. It is worth mentioning that the aforementioned weak supervision can complement one another, which makes the integration of diverse weak supervision methodologies during training possible.

3.3.2 Domain Adaptation

For domain adaptation tasks, the trained model ϕ_θ^{S} is learned from the annotated source domain dataset $\mathcal{D}_{\mathrm{S}} = \{(x_s, y_s)\}$. When the data distribution in the target domain differs, such discrepancy can lead to performance degradation of the model ϕ_θ^{S}. One research direction is presented by Unsupervised Domain Adaptation (UDA), where the raw data without annotation from the target domain $\mathcal{D}_{\mathrm{T}}^{\mathrm{UDA}} = \{(x_t)\}$ are available. By leveraging the knowledge extracted from the source domain, ϕ_θ^{S} can be adapted to a new target domain, forming ϕ_θ^{T} without any annotation on the target domain.

UDA for the segmentation task is predominantly classified into three categories: Domain alignment through statistical divergence [Liu22a, Tze14, Kan19b, Sun16], domain adversarial learning [Tze17, Vu19, Che19], and normalization statistics alignment [Li18, Mar17], alongside self-training approaches [Shi20, Lia19, Hoy23]. Several studies have also explored the use of weak labels for domain adaptation [Ino18, Wan19b, Lv21, Wan23b, Das23].

Significant attention has been paid to UDA for the lane detection task, including viewpoint transformation alignment across domains, adversarial training, and self-training. Yu et al. [Yu19] introduced a method to project images into a standard viewpoint by estimating a transformation matrix. Adversarial

learning-based methods are closely aligned with general unsupervised learning strategies. For instance, Garnett et al. [Gar20] employed a discriminator to differentiate between the model's predictions from the source and target domain, utilizing the gradient image of the input to constrain deep features extracted by the model. Furthermore, Hu et al. [Hu22] explored both generative and discriminative adversarial methods for sim-to-real domain adaptation. The MLDA proposed by Li et al. [Li22b] employs pseudo labels for lane domain adaptation, encompassing pixel-wise, instance-wise, and category-wise tasks, which aims to enhance the model's global understanding by predicting lane types within images from the target domain.

3.3.3 Utilizing Class Hierarchy

A tree-structured class hierarchy $\mathcal{T} = (\mathcal{V}, \mathcal{E})$, in which each node $v \in \mathcal{V}$ represents a semantic class and each edge $(u, v) \in \mathcal{E}$, represents class dependencies between the two classes. Specifically, a parent node v represents a more generalized class compared to a child node u. The root node of the hierarchy $\mathcal{T}$, written as v^r, represents the most generalized class, while leaf nodes $\mathcal{V}_\mu \subseteq \mathcal{V}$ correspond to the most specific classes.

Ranging from classical machine learning that utilizes support vector machines [Che04] to the latest data-driven neural networks [Li22c], class hierarchies are incorporated to increase the performance of the models. In the current computer vision tasks, single-label hierarchical classification is frequently used, while hierarchical multi-label classification, where each data point can be assigned multiple paths within the hierarchy, is also used in other machine learning tasks [Bi11b, Weh18, Giu20]. Leveraging the class hierarchies in the computer vision tasks can be categorized into three major groups [Ber20]: Label-embedding methods that project class labels into vectors capturing semantic relationships [Ben10, Aka15, Xia16]. In addition to that, hierarchies are integrated into the loss functions in order to enforce prediction consistency [Den10, Zha11, Ver12, Bil17, Ber20]. Furthermore, the network architectures are adapted to align with the class hierarchies [Zwe07, Yan15, Ahm16, Zhu17].

More specifically, for the unknown object detection leveraging class hierarchy, Lee et al. [Lee18a] proposed a hierarchical novelty detection framework for object recognition, offering top-down and flattened approaches for classification within the taxonomy. Graaff et al. [Gra22] utilized capsule networks with class hierarchy for novelty detection in object recognition. Zwemer et al. [Zwe20] enhanced an SSD with hierarchical classification, thereby improving object localization performance and evaluating hierarchical binary classification through a novel training loss function [Zwe22]. Li et al. [Li22c] developed a semantic segmentation model with integrated class hierarchy, incorporating hierarchical constraints and tree-triplet loss based on a predefined class hierarchy for urban driving scenes and for human parsing tasks.

3.3.4 Few-Shot Semantic Segmentation

Few-shot learning is often employed to mitigate the epistemic uncertainty inherent in neural networks when learning an underrepresented class with a limited number of training examples. The previously known classes are noted as base classes by $\mathcal{C}_{\text{base}}$, and the novel class set $\mathcal{C}_{\text{novel}}$, where $\mathcal{C}_{\text{base}} \cap \mathcal{C}_{\text{novel}} = \emptyset$. During inference, the model receives a support image set $\mathcal{X}_{\text{S}} = \{(x_i, m_i)\}_{i=1}^{K}$, where $K = |\mathcal{X}_{\text{S}}|$ denotes the number of support image–label pairs and m represents the corresponding binary label for the novel class. Given a query image set $\mathcal{X}_{\text{Q}} = \{x_i\}_{i=1}^{N}$ that contains the novel class $c \in \mathcal{C}_{\text{novel}}$, the model predicts a binary mask for each class c with $m_c = \{0,1\}^{H \times W}$ indicating the presence and location of the novel class c in the query image as described by the support set. During training, the FSSS models typically follow a meta learning paradigm, where it involves a random selection of classes from the base set $\mathcal{C}_{\text{base}}$, with corresponding information extracted to segment base classes, ensuring the learned model generalizes to novel classes in the novel set $\mathcal{C}_{\text{novel}}$.

Similar to other few-shot learning tasks, e.g., few-shot classification [Vin16, Fin17] and few-shot object detection [Kan19a], the FSSS paradigm typically involves comparing feature representations from the support set with those from the query image. Shaban et al. [Sha17] proposed utilizing a feature

vector as support information, facilitating the segmentation branch's prediction of the query image mask. Subsequent work adopted Masked Average Pooling (MAP) to concentrate on and enrich the feature representations of novel classes [Zha20]. Some studies [Wan19a, Zha19, Li21] leveraged support prototype vectors to reinforce segmentation predictions, as seen in PFENet [Tia20], where expanded prototype vectors and query features are concatenated with a prior mask generated by calculating cosine similarity between the extracted query and support features. Other research aims to enhance performance through multi-scale feature processing [Min21, Iqb22, Shi22] and self-attention modules [Shi22, Zha22c], improving novel class feature representation. Singular Value Fine-tuning (SVF) employs singular value decomposition to decompose the pre-trained backbone network, aiming to reduce overfitting and enhance generalization [Sun22]. Similar to open set problems introduced in Sec. 3.2, Generalized Few-Shot Semantic Segmentation (GFS-Seg) needs to segment both novel and base classes without prior knowledge of which novel classes the query image contains [Tia22, Liu23, Haj23, Mye22]. Lang et al. [Lan22] further sought to suppress false positive novel class predictions by integrating base class segmentation results into the novel class prediction branch.

Despite the advancements in the generic FSSS research, limited knowledge dealing with complex driving scenes has been obtained. Hu et al. [Hu23c] evaluated their proposed method on the Cityscapes dataset and revealed significant performance degradation compared to traditional few-shot segmentation datasets like Pascal VOC [Eve12], COCO-Stuff [Cae18], and FSS-1000 [Li20b]. This highlights the heterogeneity of the segmentation problem due to varying complexities when different image regions are segmented as novel classes, resulting in false positive predictions.

4 Preliminary Study

As shown in Ch. 3, the current evaluation scheme mainly emphasizes the performance improvements given a certain training set. In real applications, the training data, characterized by their quantity and quality, significantly affect the generalization ability of neural networks. Therefore, this chapter serves as a preliminary study and concentrates on the potential impact of training data quality on data-driven perception tasks. The presented results are partly based on the author's publication in the proceedings of the 34th Intelligent Vehicle Symposium (IV) [Zho22b].

4.1 Motivation

It is shown that large-scale datasets that accurately represent real driving scenarios ensure the performance and robustness of data-driven perception functions. Therefore, data collection is a fundamental task for continuously improving such functions. However, in several global regions, especially Europe, data containing personally identifiable information must adhere to specific legal conditions outlined in regulations during data exchange. With the implementation of the GDPR, the exchange and processing of such data must ensure that individuals can no longer be identified from the collected data. To safeguard such a process, an identified natural person during data collection is endowed with numerous rights to be informed about the data handling and processing steps, including the rights to rectification, erasure, and restriction of processing. Consequently, for data exchange, it is imperative to anonymize visible license plates and faces of individuals from the recorded data, ensuring that privacy-sensitive areas are rendered non-reconstructable. However, the

impact of such data anonymization methods on the performance of semantic segmentation models has not been sufficiently investigated.

Typically, such an anonymization pipeline incorporates an object detector for personal identity-sensitive areas alongside a module that applies a designated anonymization pattern to augment the collected images. However, such modifications of the recorded data, e.g., the removal of certain visual features and the blurring of specific regions in an image, can result in performance limitations for data-driven algorithms when performing inference on raw camera images, where such artificial modifications of visual properties do not exist, potentially leading to less efficient training processes or substantially altered training outcomes that emphasize anonymization patterns that are used during training rather than the actual content of the scenes.

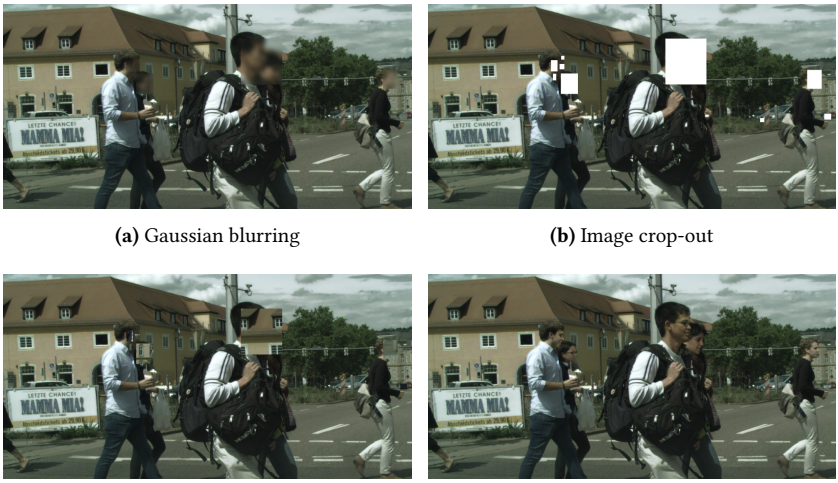
Given that the computational capacity of mass-production vehicles is significantly lower than that of server clusters, compact network architectures are usually utilized during inference in vehicles equipped with such driving functions. The objective of this chapter is to evaluate the impact of anonymization in training datasets on semantic segmentation, employing various input image resolutions and network architectures.

4.2 Experimental Setup

The primary aim of this preliminary study is to assess the potential impact of image anonymization and the statistical significance of the changes induced by anonymization within each de-identified training set for semantic segmentation. This can be achieved by evaluating whether the changes caused by anonymization within each de-identified image set $\mathcal{X}_i$, alongside with the fine annotations $\mathcal{Y}$ for the trained mapping function ϕ_θ^i are statistically significant. The applied anonymization methods, the training setup, and the network architectures employed are introduced in this section. For the experiments, the widely adopted Cityscapes dataset [Cor16] is utilized for evaluation.

Anonymization Methods

This section concentrates on four major image anonymization methods: Gaussian blurring, image crop-out, random crop, and image resynthesis. The first three methods are regarded as classical ways of anonymization: Crop-out removes specific areas of the input image that are detected as human faces and number plates, while random crop replaces them with other slices of the input images. Gaussian blur, which applies convolution to an image based on a Gaussian filter, is a simple yet effective anonymization method widely used in pratice. For image resynthesis, a generative model proposed by Hukkelås et al. [Huk19] is applied to the images by taking key points from the detected faces and replacing them with realistic generated images using conditional Generative Adversarial Network (GAN) [Kar19]. The dataset is denoted as $\mathcal{X}_{\text{IR}}$. Examples of the anonymized images can be found in Fig. 4.1.



(a) Gaussian blurring

(b) Image crop-out

(c) Random crop

(d) Image resynthesis

Figure 4.1: Examples of anonymized images from Cityscapes dataset.

To address the potential issue of false positive and false negative predictions from the face and number plate detector, the Regions of Interest (RoIs) that

need to be anonymized are generated by comparing an official anonymized image set with the original dataset. The quantitative comparison for the RoI generation is based on the Structural Similarity Index (SSIM), proposed by Wang et al. [Wan04], where the luminance, contrast, and structural differences of the images are compared. Based on the extracted RoIs and the original raw dataset $\mathcal{X}_0$, the anonymized image sets $\mathcal{X}_{\text{GB}}$, $\mathcal{X}_{\text{IC}}$, $\mathcal{X}_{\text{RC}}$ are generated for the upcoming experiments.

Network Architectures

For the comparison of network architectures, three primary decoders are utilized, FCN [Lon15], PSPNet [Zha17] and UPerNet [Xia18]. Compared with the setup with the FCN decoder, the latter two image decoders leverage intermediate convolutional layers from the image encoders, structuring multi-scale feature embeddings. In the pyramid pooling module of PSPNet, feature embeddings from the backbone are pooled at multiple scales before being processed through a convolutional layer. By leveraging the PSP modules and combining them with an FPN network, UPerNet demonstrates its ability to handle multiple computer vision tasks. It is used by the current evaluations with Swin Transformer and ConvNeXt encoders.

Therefore, in addition to the classical ResNet family, ranging from ResNet-18 to ResNet-101, further network architectures that are heavily inspired by the transformer models are included in the evaluation and comparison, including ConvNeXt-Pico and ConvNeXt-Tiny, which are comparable to ResNet-18 and ResNet-50 in their model sizes. During training, a fixed batch size of 6 is utilized. The models are trained using the default optimizer – stochastic gradient descent for the models with ResNet backbones and AdamW for the models with ConvNeXt backbones.

Evaluation Setup

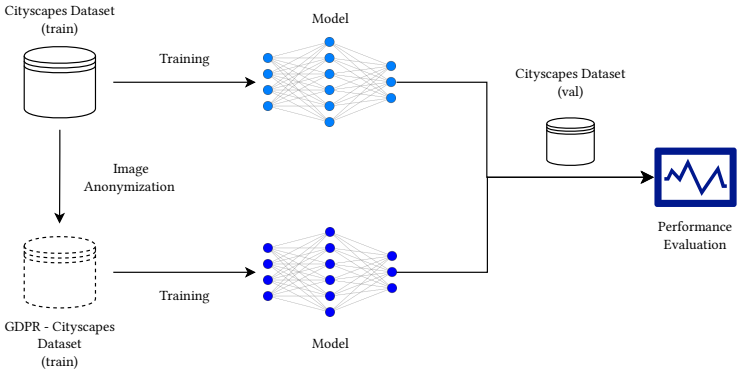


Figure 4.2: Training and evaluation setup for the preliminary study.

As shown in Fig. 4.2, the evaluation and comparison of the models that share the same hyperparameters with different data sources are summarized as follows:

1 The baseline semantic segmentation models are trained with the original Cityscapes dataset that contains raw camera images and fine annotations $\mathcal{D}_0 = (\mathcal{X}_0, \mathcal{Y})$;

2 The anonymized image sets are generated based on extracted RoIs or the GAN model from Hukkelås et al. [Huk19]. By combining the anonymized image sets with the fine annotations from the original dataset, $\mathcal{D}_{\text{GB}}$, $\mathcal{D}_{\text{IC}}$, $\mathcal{D}_{\text{RC}}$, and $\mathcal{D}_{\text{IR}}$ are created with $\mathcal{D}_i = (\mathcal{X}_i, \mathcal{Y})$;

3 The segmentation models ϕ_θ^i are trained with identical hyperparameter settings as their respective baseline models with training dataset $\mathcal{D}_i$;

4 Aligned with real driving scenarios, in which the camera stream is directed fed to the computing unit, the trained models ϕ_θ^i are applied for validation utilizing the original Cityscapes validation dataset $\mathcal{D}_0$.

In addition to the anonymization pattern used, the experiments further ablate the impact of image data anonymization with various image encoders, decoders, and training image crop sizes. The image backbones are pre-trained on ImageNet-1k. Two distinct image resolutions are used during training, with 1024×512 representing a higher input resolution and 768×384 a lower resolution. Each model is trained for up to 10^6 iterations. A series of data augmentation methods, including random cropping, random flipping, and photometric distortions, are applied during the training of semantic segmentation models. The mean Intersection-over-Union (mIoU) metric is utilized for the evaluation of the trained semantic segmentation models based on the Cityscapes validation dataset. In addition, the experiments consider classes related to instances of the category *human* and *vehicle* that are directly affected by image anonymization. Similar to the evaluation scheme suggested by Cordts et al. [Cor16], two distinct mean performance scores, $\text{IoU}_{\text{category}}$ and $\text{IoU}_{\text{class}}$, based on two semantic granularities, are reported.

Furthermore, non-parametric statistical tests assess the statistical significance of image anonymization on segmentation models' performance. The Wilcoxon test evaluates whether the IoU of related categories changes significantly in response to a specific anonymization pattern [Wil45], offering an averaged analysis across all training input resolutions, neural network architectures, and backbones. As a supplement to the averaged analysis across all the models, the Mann-Whitney U test examines whether the distribution of segmentation performance from a single model varies when trained with anonymized data [Man47].

4.3 Results and Analysis

As introduced in Sec. 4.2, the quantitative evaluation of the impact of anonymization comprises multiple experiments involving diverse image crop sizes during training, network architectures, and anonymization patterns. However, to compare the impact caused by certain variables, evaluation results from other possible training configurations are combined and averaged to minimize variability and ensure readability despite the differences

Table 4.1: Relative comparison of the models trained on different image resolutions

	Hi-Res	Lo-Res
Class		
Person	**-0.84**%	-0.49%
Rider	**-1.76**%	-0.57%
Car	**-0.27**%	-0.19%
Bicycle	**-0.58**%	-0.55%
Category		
Human	**-0.76**%	-0.42%
Vehicle	-0.20%	**-0.23**%
mIoU	**-0.63**%	-0.15%

Table 4.2: Relative comparison of the decoder architectures

	PSPNet	FCN
Class		
Person	-0.6%	**-0.96**%
Rider	-1.10%	**-2.70**%
Car	-0.13%	**-0.27**%
Bicycle	**-0.40**%	-0.38%
Category		
Human	-0.62%	**-0.80**%
Vehicle	-0.10%	**-0.13**%
mIoU	0.05%	**-0.86**%

in image backbones and anonymization patterns. For example, to compare the impact of diverse training image sizes, Tab. 4.1 presents the results from experiments in which the crop size of the input image is the only variable. In this scenario, results across three different image encoders, two decoders, and four anonymization patterns are averaged, resulting in 24 experiments for each image size. Given the relatively small variations in the absolute IoU differences, unless otherwise stated, the relative changes of the class IoU are calculated and reported with $d_{\text{IoU}} = (\text{IoU}_{\text{ano}} - \text{IoU}_{\text{orig}})/\text{IoU}_{\text{orig}}$. In addition, statistical analyses are performed for selected anonymization patterns.

Table 4.1 demonstrates that models trained with high-resolution input images suffer from performance degradation due to anonymization across all the *human* and *vehicle* classes compared to those trained with low-resolution images from $\mathcal{D}_0$. One possible explanation for that is that the segmentation networks are encouraged to concentrate on global textures and visual appearances. In addition to that, local features are diminished in their priority among the downsampled training examples compared to their representation in the original resolution. Similarly, within the decoder architectures, as observed in Tab. 4.2, the architectures that leverage multi-scale inputs from the image encoder demonstrate higher robustness in mitigating the negative impact brought by the anonymized training images.

Table 4.3 illustrates the average relative IoU changes caused by applying various anonymization patterns in the semantic segmentation training dataset. It can be observed that the crop-out method results in the highest IoU loss, which is attributed to the complete deletion of the detected privacy-sensitive areas containing number plates and human faces. Similarly, the random crop method replaces essential image regions containing personal information with random patches from the training images. Despite the fact that the segmentation performance for the categories *human* and *vehicle* decreases, no significant negative impact on the mean IoU can be observed, indicating better segmentation of other categories like *flat* that contains classes *road*, *sidewalk*, and category *construction* with classes *building* and *wall*. Utilizing Gaussian filters for anonymization, it can be seen that the relevant class IoUs are negatively affected by the moderate changes. In contrast, image resynthesis preserves the segmentation performance in the category *human*, with classes such as *person* and *rider* unaffected despite the image anonymization actions.

Image anonymization has a comparatively greater impact on the *person* and *rider* classes within the *human* category than on the *car* and *truck* classes within the *vehicle* category. Such discrepancy arises due to the fact that human faces often cover the boundary of one person. In contrast, vehicle contours are frequently defined by body panels instead of centrally located number plates from front- and rear-view. Additionally, a decrease in the segmentation performance is also observed for classes not directly relevant to image anonymization, such as *bicycle*. This decrease is attributed to the interdependence of recognizing the object instances, e.g., bicycles and their riders. As anonymization complicates the segmentation performance of the class *rider*, it thereby also increases the segmentation failures in the relevant classes simultaneously.

Table 4.3: Relative comparison of the models trained on various anonymization patterns

	Crop Out	Random	Gaussian	Resynthesis
Class				
Person	-0.99%	-0.93%	-0.43%	0.08%
Rider	-2.87%	-2.00%	-0.82%	0.09%
Car	-0.29%	-0.14%	-0.17%	-0.13%
Bicycle	-0.22%	-0.56%	-0.40%	-0.09%
Category				
Human	-0.89%	-0.83%	-0.40%	0.07%
Vehicle	-0.07%	-0.18%	-0.10%	-0.01%
mIoU	-0.87%	0.02%	-0.31%	-0.05%

Table 4.4 ablates the impact of employing different ResNet-based backbone networks. Performance degradation induced by image data anonymization decreases with the increasing capacity of the backbone network, as the segmentation models with the ResNet-101 backbone are generally less affected by anonymization. Such observations align with the findings from Geyer et al. [Gey20], who evaluated image data anonymization using PSPNet decoder with ResNet-101 backbone and concluded that anonymization does not negatively impact image segmentation. However, it is worth mentioning that more complex backbones also impose higher computational resources. Edge applications, e.g., robotics and automated driving functions, are often considered to have limited resources. Therefore, the trade-off between network capacity and the feasibility of online implementation should be considered when designing such perception functions.

Table 4.4: Relative comparison of the models with backbones from the ResNet family trained on anonymized data.

	ResNet-18	ResNet-50	ResNet-101
Class			
Person	-1.24%	-0.80%	-0.32%
Rider	-1.83%	-1.82%	-1.05%
Car	-0.25%	-0.26%	-0.15%
Bicycle	-1.25%	-0.03%	-0.26%
Category			
Human	-1.25%	-0.58%	-0.30%
Vehicle	-0.31%	0.05%	-0.10%
mIoU	-0.82%	-0.72%	0.37%

In addition to the ResNet backbones, in Tab. 4.5, further image encoders from the ConvNeXt family, which are heavily inspired by the recent development of vision transformers, are ablated. The image encoders are paired with the UPerNet image decoder, and the experiments are repeated five times to eliminate the possible perturbations caused by random seeds. Consistent with the findings presented in Tab. 4.4, despite demonstrating superior performance on the Cityscapes dataset as measured by mIoU, the backbone with lower learning capacity is still more adversely affected by data anonymization. Such negative impact can be observed by the underrepresented classes like *rider*, where the absolute difference of the single class decreases by up to 2% with higher standard deviations, while the models with more parameters can cope with such disturbances. Furthermore, modern CNNs are also more robust compared to the ResNet family, as shown by the comparison between ConvNeXt-Tiny and ResNet-50, which have similar model sizes.

Table 4.5: Comparison of segmentation performance utilizing various modernized image encoders and training datasets

	Original	Crop Out	Random	Gaussian	Resynthesis
			mIoU /%		
ConvNeXt-Pico	$77.1_{\pm 0.3}$	$76.5_{\pm 0.4}$	$76.7_{\pm 0.2}$	$76.7_{\pm 0.4}$	$77.2_{\pm 0.4}$
ConvNeXt-Tiny	$80.5_{\pm 0.2}$	$80.2_{\pm 0.1}$	$80.3_{\pm 0.4}$	$80.6_{\pm 0.2}$	$80.5_{\pm 0.2}$
UPerNet-R50	$75.7_{\pm 0.1}$	$75.4_{\pm 0.2}$	$75.1_{\pm 0.3}$	$75.7_{\pm 0.4}$	$75.7_{\pm 0.2}$
			Person /%		
ConvNeXt-Pico	$81.3_{\pm 0.1}$	$80.4_{\pm 0.1}$	$80.0_{\pm 0.3}$	$80.4_{\pm 0.2}$	$81.2_{\pm 0.2}$
ConvNeXt-Tiny	$83.6_{\pm 0.1}$	$83.6_{\pm 0.1}$	$83.4_{\pm 0.2}$	$83.7_{\pm 0.1}$	$83.8_{\pm 0.2}$
UPerNet-R50	$80.7_{\pm 0.1}$	$80.0_{\pm 0.1}$	$79.9_{\pm 0.1}$	$80.1_{\pm 0.1}$	$80.3_{\pm 0.3}$
			Rider /%		
ConvNeXt-Pico	$60.6_{\pm 0.6}$	$59.2_{\pm 0.7}$	$58.6_{\pm 0.8}$	$59.6_{\pm 0.7}$	$60.2_{\pm 0.9}$
ConvNeXt-Tiny	$64.2_{\pm 0.5}$	$64.1_{\pm 0.6}$	$64.1_{\pm 0.4}$	$64.5_{\pm 0.4}$	$64.4_{\pm 0.8}$
UPerNet-R50	$58.6_{\pm 0.7}$	$56.9_{\pm 0.8}$	$57.3_{\pm 0.7}$	$57.5_{\pm 0.7}$	$57.3_{\pm 1.2}$

Figure 4.3 compares several segmentation images produced by an FCN network with a ResNet-18 backbone that was previously trained on the Gaussian anonymized Cityscapes dataset. In this study, GradCAM, as implemented in Captum [Kok20], is utilized to evaluate pixel contributions to the class *human*. The analysis focuses on the gradients of the last convolutional layer within the decode head of the neural network. The first row illustrates a scene where the entire body of a pedestrian is visible to the camera. In this setup, pedestrians can be detected and segmented by the semantic segmentation network despite notable uncertainty in the human faces shown by the entropy map. There are instances, depicted in rows b to d of the qualitative results, where the network fails to segment the input images accurately. A substantial decline in segmentation performance is noted when only the faces of pedestrians, regarded as privacy-sensitive regions during data anonymization, are visible from the vehicle's camera. The entropy of class *person* in row d is markedly higher compared to rows e and f, indicating greater uncertainty in the neural network trained with the anonymized Cityscapes dataset versus the original and re-synthesized datasets.

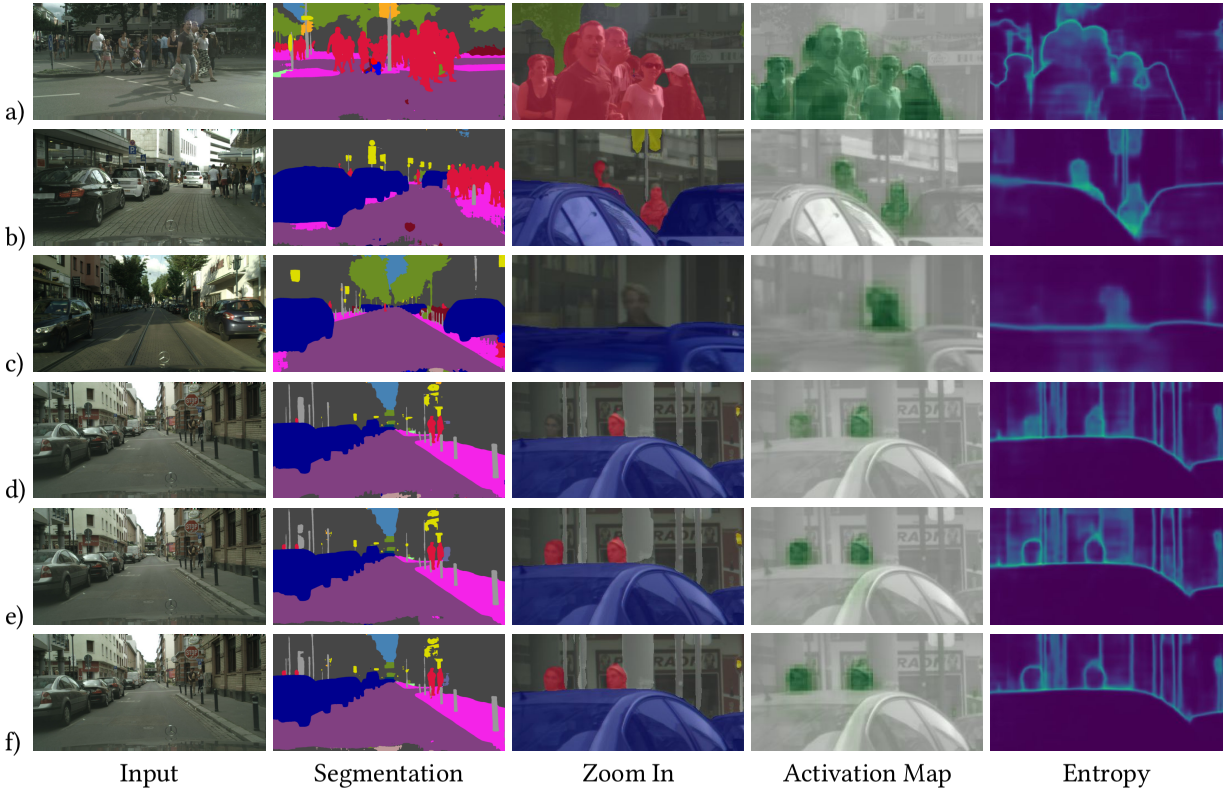


Figure 4.3: Qualitative results on Cityscapes dataset from FCN network with ResNet-18 backbone. The neural network was previously trained on the dataset with Gaussian anonymization pattern. The examples illustrate the impacts of data anonymization on the class *Person*.

Statistical Analysis

Given the widespread use of the Gaussian filter in practice due to its simplicity, the statistical analysis concentrates on the impact of utilizing this method for anonymization. For comparison, the FCN model is trained 42 times on the raw Cityscapes dataset with higher resolution and 34 times on its anonymized variant, maintaining identical training configurations. As introduced in Sec. 4.2, the Wilcoxon signed-rank test and Mann–Whitney U test were applied to evaluate the segmentation precision from the experiments. In the test setups, the null hypothesis $H_0 : \mathrm{IoU}_{\mathrm{orig}} \leq \mathrm{IoU}_{\mathrm{ano}}$ is tested against the alternative hypothesis $H_1 : \mathrm{IoU}_{\mathrm{orig}} > \mathrm{IoU}_{\mathrm{ano}}$, with the significance level $\alpha = 0.01$ chosen to ensure a higher confidence level. If the p-value of the test falls below the significance level α, the null hypothesis H_0 is rejected. Table 4.6 reveals a significant impact of traditional anonymization patterns on the *human* category, while the impact on the *vehicle* category and mIoU is not substantial. Table 4.7 also indicates statistically significant changes caused by the Gaussian filter on the ResNet-18 based FCN segmentation network. Notably, segmentation results for the *rider* class and associated *motorcycle* and *bicycle* classes exhibited significant differences in segmentation precision. The statistical analysis corroborates the quantitative and qualitative findings, demonstrating that image anonymization significantly affects semantic segmentation, particularly for neural networks with a smaller backbone.

Table 4.6: p-values of Wilcoxon signed-rank test on the models' performance trained on different anonymization patterns.

	Crop Out	Random	Gaussian	Resynthesis
Category				
Human	$<10^{-3}$	$<10^{-3}$	<0.01	0.689
Vehicle	0.190	0.0757	0.102	0.396
mIoU	0.046	0.017	0.425	0.311

Table 4.7: p-values of Mann-Whitney U Test on segmentation performance when trained with Gaussian anonymized dataset.

Person	Rider	Car	Bicycle	Motorcycle	mIoU
$<10^{-14}$	$<10^{-8}$	0.103	$<10^{-4}$	<0.01	0.037

4.4 Conclusion

In this chapter, a systematical analysis of the impact of image data anonymization on semantic segmentation as a preliminary study is proposed. The performance loss has been investigated qualitatively as well as quantitatively. It is shown that neural networks with more complex backbones and multi-scale features are better at generalizing from anonymized training data than smaller ones. While face anonymization degrades the detection performance on partially obstructed pedestrians, number plate blurring has only a marginal effect on the detection of vehicles. However, using generative models to regenerate biometric features can mitigate the side effects of data anonymization, while ensuring privacy compliance.

Recently, transformer models have also been utilized for computer vision tasks like object detection and semantic segmentation in research. However, as it is shown in Tab. 4.5, backbones that are inspired by the latest advancements that have a smaller learning capacity are still statistically affected by the image anonymization process, showing its high data relevance. In the following chapters, starting with Ch. 5, the aspect of data-related corner cases will be further discussed.

5 Corner Cases for Automated Driving

The goal of this chapter is to propose an alternative definition and classification of safety-critical driving situations related to ML-based environment perception, reducing ambiguity from the perspective of data-driven tasks.

Consequently, the following research question is therefore raised and discussed:

> **Research Question 1**: What are the core properties of corner cases in the context of real-world ADAS and HAD applications?

- Hypothesis 1.1: In addition to the current methodologies, corner cases can also be classified according to the availability of the data points during development and validation.

This chapter is mainly based on the author's publication in the proceedings of the 35th IEEE Intelligent Vehicles Symposium [Zho23c].

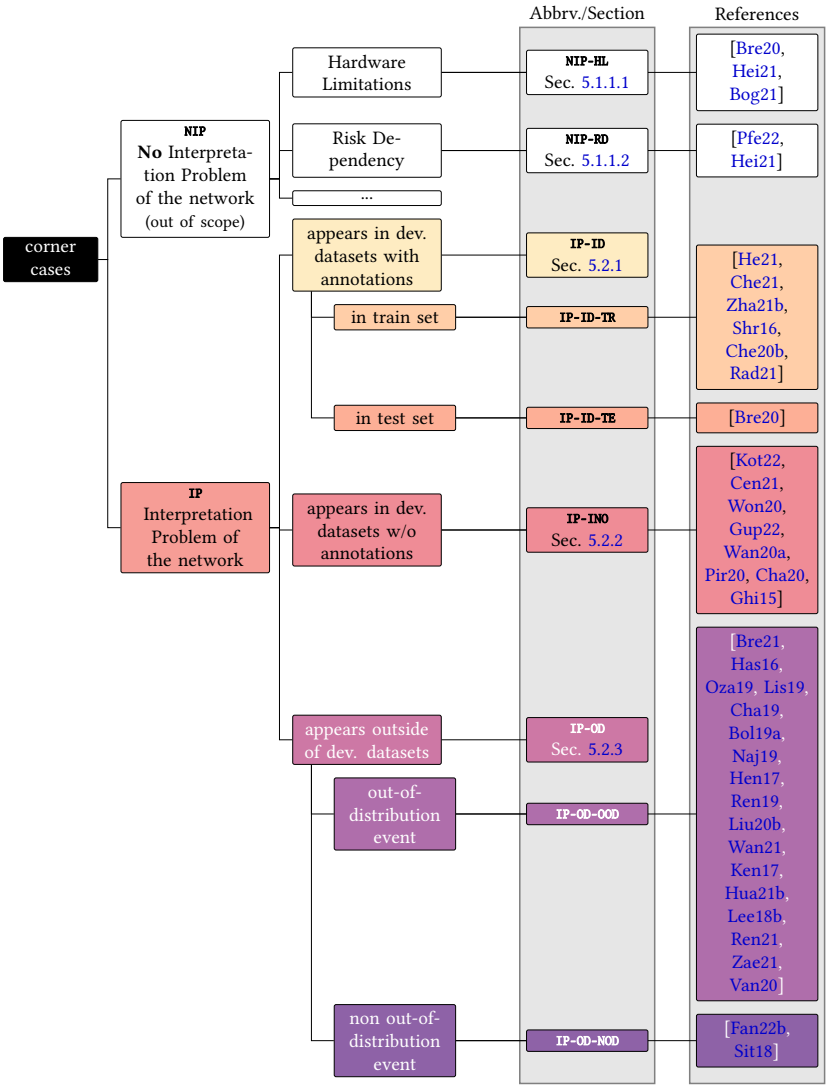


Figure 5.1: Classification of *corner cases* from the perspective of interpretation problems in the neural networks.

5.1 Interpretation-Error-Based Criteria

As outlined in Sec. 3.1, prior research has primarily concentrated on the classification and taxonomy of corner cases within the context of highly automated driving functions that rely on visual systems [Bre20, Bog21] or other modalities [Hei21].

Nevertheless, given that the majority of the sub-modules involved in automated driving functions – such as perception and motion planning – are either directly based on neural networks or indirectly reliant on their outputs, it is pertinent to consider corner cases from the perspective of network interpretation, especially since there are network setups that streamline end-to-end automated driving functions. This involves assessing whether any interpretation issues within the networks lead to corner cases, thereby linking these cases to the properties of the training and testing datasets, as necessitated by the EU AI Act [Eur21]. The subsequent sections will systematically discuss the taxonomy of corner cases as depicted in Fig. 5.1, referencing relevant existing literature where applicable.

5.1.1 No Interpretation Problem (NIP)

Even in scenarios where the neural networks exhibit no interpretation issues, various factors can still pose risks and lead to hazardous situations. These factors range from technical failures in the vehicle, e.g., from the sensors, as addressed by ISO 26262, or from the unpredictable and risky behaviors of other traffic participants, as covered by ISO 21448. Although this is not the primary focus of the thesis, such corner cases are still illustrated through two potential influences, namely hardware limitations (*NIP-HL*) and risk dependency (*NIP-RD*), which are distinct from neural interpretation problems and will be expounded in the upcoming Sec. 5.2.

5.1.1.1 Hardware Limitations (NIP-HL)

Given that sensors generate input data for neural networks, significant distortions or noise can affect the quality of the model's output. For instance, adverse weather conditions may severely impact the functionality of automated driving functions [Hah22, Cae20], as vehicles equipped with both cameras and LiDAR sensors face visibility constraints in foggy conditions due to water droplets that scatter, reflect, and absorb LiDAR rays while limiting passive camera visibility [Gey20]. Under- and over-exposure can be frequently observed upon entering and exiting tunnels. These effects are already notable without considering any data processing with data-driven methods and may induce risks in conventional, non-ML systems.

In addition to that, multiple sensors and sensor modalities are often deployed along the vehicle's primary axis to ensure redundancy and maintain a robust sensor field of view around the driving environment. Nevertheless, the perception of the driving environment is still affected by blind spots created by other traffic participants or static objects, such as an obscured bicyclist behind a truck due to limited visibility. Consequently, effectively detecting these objects necessitates Vehicle-to-Everything (V2X) communications [Che17], specifically Vehicle-to-Infrastructure (V2I) or Vehicle-to-Vehicle (V2V). These concepts, however, are not primarily associated with neural network interpretations.

5.1.1.2 Risk Dependency (NIP-RD)

The inherent risks posed by the driving behavior of other traffic participants are a consistent key factor for automated driving functions. The primary motivation for detecting and eliminating corner cases is to mitigate any potential collisions. The impact of a certain object on the automated vehicle is influenced by several factors: Its position and speed relative to the ego vehicle, its mass (or mass difference), and its vulnerability.

Typically, neural networks encounter challenges with interpretation problems that are intricately linked with these risk dependencies. Nevertheless, the

presence of risks is concurrently a fundamental aspect of decision-making in traffic and is not exclusively tied to the interpretation challenges with data-driven neural networks, especially when the network can accurately depict such driving scenes, e.g., overtaking bicycle riders or unusual driving behavior of other vehicles.

5.2 Causes of Interpretation Errors

This section focuses primarily on corner cases caused by interpretation problems in neural networks, emphasizing the critical role of data utilized for training and validation. To reduce complexity, distinctions among the data available to various groups responsible for automated driving functions, such as function development teams and their testing groups, suppliers, OEMs, certification bodies, and approval agencies, are omitted. Despite the fact that such distinctions are practically significant and relevant, they are considered fundamentally organizational within a part of the collaborative process to ensure the release of "safe" products for public roads. Consequently, all data accessible up to the vehicle's certification are aggregated into the *development dataset*. Such a dataset can be further divided into three subsets: The annotated *training set* used directly for model training, the annotated *test set* used for performance evaluation on unseen data, and an *un-annotated set* representing real-world driving scenarios that were collected during the entire development process that are considered useful for development purposes without exhaustive annotation.

Furthermore, any data not included in this *development dataset* is assumed to occur during field operations, likely on public roads used by customers, which is considered to entail a higher risk.

5.2.1 In Annotated Development Dataset (IP-ID)

For any issue arising within the *development dataset*, a differentiation is made between occurrences within the direct training data (*train set*) and those only

within the data used for validation and potentially iterating with different parameters (*test set*).

In Train Set (IP-ID-TR)

In cases where the neural networks show inadequate performance even on the training dataset, as indicated by poor training metrics, multiple factors may be responsible. These may include insufficient learning capability of the neural network, such as inadequate size or inconclusive training data. An often-overlooked but essential aspect of training pertains to the training procedure of neural networks [He21]. Improper configuration of learning rates or data augmentations applied during training can strongly affect performance and inhibit the required generalization on different domains.

Another significant factor is the pretraining phase. Image encoders are typically pretrained in a supervised manner with large-scale datasets like ImageNet [Den09]. However, constraints related to cost, proprietary considerations, and specific requirements from downstream tasks require more efficient and effective pretraining methods. Existing studies demonstrate the advantages of improved training organizations, such as utilizing alternative forms of supervision [Che21], leveraging self-supervised contrastive learning methods [Zha21b], and mining challenging training examples [Shr16]. These approaches have improved upon random initialization methods and achieved performance comparable to supervised pretraining [Che20b, Rad21].

In Test Set (IP-ID-TE)

In this case, the interpretation problem of the networks only appears in the *test set*, which is a subset of the *development dataset*. The steps outlined in the previous section (*IP-ID-TR*) should be revisited to exclude a fundamental lack of generalization from the neural networks, specifically, compare the network's learning capacity with the size of the training dataset to rule out simple overfitting. If neither of these explanations accounts for the observed interpretation problem, a high-priority task is determining whether the data

points in the development dataset are independent and identically distributed (i.i.d.). A performance disparity between training and test data suggests that their distributions may differ. Additional data collection aims at extending the annotated *development dataset*, thus supplementing the training set and test set, which can mitigate epistemic uncertainty arising from a lack of diversity from the datasets.

5.2.2 In Raw Development Dataset (IP-INO)

Challenges associated with unannotated raw datasets are not immediately identifiable by methodologies mentioned in previous subsections. Exhaustive annotation is often impractical, as it would ideally have been completed with the dataset already included in the annotated subset. This necessitates alternative approaches to identify pertinent scenes or scenarios, which poses significant challenges leveraging diverse detection methods with minimal or no supervision from manually generated labels.

Current research in this direction predominantly targets data with clear distinctions between objects and backgrounds [Vo21], while investigations into complex scenarios, such as automated driving and other sensor modalities like LiDAR, remain limited [Cen21, Naj19]. Nevertheless, image retrieval methodologies, such as deep template matching [Kot22], that seek to identify comparable images based on the distance between features generated from extensive unlabeled datasets, given a small subset of query images, are beneficial for identifying previously unknown critical driving situations. Additionally, methods aiming to detect previously unknown object instances that share common semantic properties based on annotated datasets also exhibit potential [Won20, Gup22]. Beyond image-level techniques, there are approaches grounded in LiDAR inputs, combined with unsupervised scene flow estimation and automatic meta-labeling, which have shown promising results in open set 3D detection tasks [Naj19].

5.2.3 No Data Available (IP-OD)

If such an interpretation problem of the neural network does not arise within the *development dataset*, it indicates that there is a misalignment between the learned distribution and the domain of the system during operations. This misalignment can occur due to an incomplete dataset or an inadequate operational environment that differs from the learned data distribution. Similar effects arise from mismatches between training and test datasets (*IP-ID*) with overlapping causes. However, the consequences are potentially more severe as these issues are discovered during product operation on public roads, not during development or certification.

Domain shifts may occur, for instance, when networks trained with European data are validated with data from the USA, or due to variations in driving conditions, as extensively discussed in Sec. 3.3. Networks trained solely under fair weather conditions exhibit significant performance drops when real driving situations involve more challenging conditions [Hah22, Cae20]. Domain shifts in the temporal context are particularly noteworthy: As the domain evolves over time, any static *development dataset* inevitably loses its i.i.d. property relative to the operational domain. Based on whether an event represents an out-of-distribution (OOD) event – meaning it is not reflected in a broadly sampled training dataset – this section is split into two sub-parts: Out-of-Distribution events (*IP-OD-OOD*) and non Out-of-Distribution events (*IP-OD-NOD*).

Out-of-Distribution Events (IP-OD-OOD)

In this category, the issue emerges when sensor data as input during inference significantly deviates from the learned distribution represented in the *development dataset*. Such deviations may manifest as high uncertainty outputs from the data-driven models. To mitigate such an issue, it is feasible to implement a separate OOD detection mechanism without altering the *development data*, the sensor setup, or the operational domain. This auxiliary mechanism can identify and potentially mitigate these OOD events as a contingency measure.

Extensive research has been done on OOD detection [Yan21a, Bre21], which can be broadly classified into reconstruction-based, prediction-based, post-processing score-based, and density-based methods.

Reconstruction-based methods involve employing an ensemble of data encoders and decoders to verify if the feature representations generated from the image encoder can be used to reconstruct the input data accurately [Has16, Oza19, Lis19]. Significant differences between the reconstructed and the original input data suggest a drift from the known data distribution. This methodology can also be adapted to generative adversarial networks [Cha19]. Prediction-based methods focus on comparing predicted data based on previous data sequences to the actual measurement data. Similar to reconstruction-based methods, significant disparities between the predicted and real data indicate an inconsistency between the input and the training dataset [Bol19a, Naj19]. Postprocessing score-based methods utilize maximum softmax score [Hen17], temperature-based score [Ren19], and energy-based score [Liu20b, Wan21]. The gradients from the network are employed to estimate uncertainty from the model outputs [Hua21b], or multiple forward passes are executed to gather this information [Ken17]. Recent advances in uncertainty quantification have examined the rejection of input examples that exhibit unreliable network uncertainties [Fis22].

Additionally, density and probabilistic models assume that low-density areas represent underrepresented situations. Feature representations from the backbone are compared to feature embeddings derived from previously known data. Various distance metrics, including Mahalanobis distance [Lee18b, Ren21], cosine similarity [Zae21], kernel similarity [Van20], and normalizing flow models [Jia22, Nal19], are used to predict corner cases.

Non Out-of-Distribution Events (IP-OD-NOD)

In this category, the issue arises even though the input data falls entirely within the distribution scope of the development data as perceived by the system. Notably, this implies that the joint distribution between raw data and

their true interpretations changes, even when the distribution of the input remains constant. Consequently, the generated outputs from a learned mapping may deviate from the true interpretation. From this perspective, the learning distribution must vary to effectively distinguish the perceived (likely: "safe") interpretation from the true (likely: "unsafe") ground truth. However, this distinction cannot be reliably established based on the given sensor, processing structure, and development dataset.

Since the systematic detection of such events at runtime is not feasible [Fan22b], the primary countermeasure involves minimizing the probability of introducing such vulnerabilities during the development phase. This may manifest as limited development data that overlooks critical distinctions, insufficient system perspective, e.g., lack of awareness of geographic changes or perception at inappropriate wavelengths or resolutions, or adversarial attacks [Sit18]. Countermeasures involve conducting exhaustive completeness evaluations on the *development dataset*, incorporating synthetic data to implicitly introduce assumptions about valid and invalid variations.

It is important to note that occurrences in this category represent the most severe type of problems, as they arise at runtime without any straightforward means for in-situ resolution.

5.3 Solutions to Corner Cases

Active Learning

Since data annotation is regarded as labor-intensive due to its nature, research is currently exploring how such resources can be efficiently leveraged for annotating raw data. Active learning has the main objective of iteratively extracting a subset of an unlabeled dataset for annotation in such a way that annotated data optimally benefits the training of the given neural networks. These newly acquired data points may come from previously known data (*IP-ID*), from raw datasets that have been collected from the real world (*IP-INO*), or from an OOD detector operating online within a vehicle fleet (*IP-OD*). Such

acquisition functions responsible for sample selection usually rely on network uncertainty [Gud20] or data diversity metrics [Jia22].

Class-Incremental Learning

When new classes emerge within the *development dataset*, the goal is to incorporate them into the neural networks with minimal effort compared to complete retraining with the whole dataset. Current research in class-incremental learning predominantly addresses the image domain. Here, the central assumption is the non-reappearance of previously known classes in the sequential training steps, as normally only the novel classes are annotated, emphasizing the need to mitigate catastrophic forgetting to ensure the network retains the ability to recognize all previous classes. Methods such as replay [Ach20], which preserves a fraction of data from previous tasks, and regularization [Kir17], which constrains model changes or employs soft labels from the latest model [Li17], have shown efficacy. However, in practical applications, it is more realistic to adopt a framework where the *no repetition constraint* is relaxed [Cos22], as previously known classes do not vanish entirely in the new dataset.

Domain Adaptation and Generalization

Domain adaptation becomes crucial when the training data in the *development dataset* do not satisfy the i.i.d. assumption across operational domains. The objective is to adapt a model trained on the *development dataset*, also called the source domain, to perform well on the datasets that exhibiting semantic difference from the training dataset, called the target domain. Various methods emphasize transferring the style of images and normalizing feature representations [Wan20a]. In addition, domain generalization methods aim to develop networks capable of learning domain-invariant features from the *development dataset*, thereby enabling a generalized model to handle multiple target domains [Pir20, Cha20, Ghi15].

Advanced Data Augmentation

In addition to conventional augmentation strategies such as cropping, image flipping, and color shifting applied to image datasets, advanced data augmentation methods have demonstrated substantial efficacy across numerous machine learning applications, including computer vision [Cub18], LiDAR object detection [Che20c], or behavior generation [Lia20]. These novel data augmentation methods enhance model performance without necessitating additional training examples and can be employed in scenarios where certain driving scenes or scenarios are present within the development dataset, e.g., from *IP-ID* and *IP-INO*.

5.4 Conclusion and Research Questions

This section presents a solution-oriented approach to defining corner cases based on interpretation problems in neural networks. The potential causes of these corner cases are examined based on data availability in the *development dataset* utilized during development, certification, and type approval, alongside factors that are independent of interpretation issues. It is considered that the interpretation problems caused by the data points that are not represented in the *development dataset* pose high risk during real-world operation for potential public users. Therefore, this section also extensively explores potential technical solutions to enhance the capabilities of data-driven driving functions comprehensively.

As large-scale data annotation for computer vision tasks is highly cost- and labor-intensive, the following chapters of the thesis concentrate on leveraging existing contextual knowledge from downstream tasks to mitigate certain interpretation problems of the neural networks that do not require additional annotated data with a priori knowledge, which motivates the subsequent research questions:

- **Research Question 2**: How can hierarchical taxonomies be leveraged in downstream computer vision tasks to evaluate and improve the generalization ability of the trained neural networks?
- **Research Question 3**: Does the class hierarchy reduce the required effort for training in computer vision tasks since the training process becomes more effective and the amount of data needed for training decreases?
- **Research Question 4**: Beyond class taxonomies, how can other contextual knowledge be leveraged to mitigate corner cases?

6 Hierarchical Class Taxonomies as Contextual Knowledge for Object Detection

Chapter 5 analyzed the roots of corner cases according to the availability of the data during development. Beginning with this chapter, the thesis focuses on the mitigating corner cases, namely, how to leverage a priori knowledge from the driving world so that the trained networks achieve better performance. Specifically, this chapter emphasizes object detection, while Ch. 7 also investigates the task of semantic segmentation.

This chapter addresses the following research question:

> **Research Question 2**: How can hierarchical taxonomies be leveraged in downstream computer vision tasks to evaluate and improve the generalization ability of the trained neural networks?

- Hypothesis 2.1: A hierarchical taxonomy based on visual appearance can be exploited to distinguish unknown objects from a known category.

This chapter is based on the author's publication in the proceedings of the International Conference on Intelligent Transportation Systems (ITSC) 2023 [Zho23d].

6.1 Dataset Setup

In addition to the large-scale real-world dataset nuImages, which has been introduced in Sec. 2.4, the evaluation of unknown-aware hierarchical object detection models is conducted using a synthetic dataset generated by the open-source autonomous driving simulation engine CARLA [Dos17]. The simulation engine offers comprehensive 3D representations of previously defined urban and highway traffic scenarios and supports a diverse array of vehicle models, buildings, pedestrians, street signs, and other elements. Additionally, CARLA allows for the specification of various environmental conditions, such as weather and time of day, and accommodates customized sensor configurations, including multiple cameras positioned at different locations. A significant advantage of utilizing synthetic datasets is the ability of the simulation engine to generate fully disjoint training and testing class distributions. The controlled driving conditions within the simulator ensure that the test set contains classes excluded from the training set, representing novel classes.

For this study, the front camera is employed to capture urban traffic scenarios across multiple predefined towns, incorporating various weather conditions and times of day. The training and evaluation processes utilize the provided ground-truth 2D bounding boxes for objects classified as *person*, *car*, *bicycle*, *motorcycle*, *truck*, and *van*. In total, the dataset comprises over 10 000 images with 25 000 bounding box annotations across the specified classes. The dataset is divided into training and validation sets in a 70%:30% ratio. Importantly, all images containing at least one object from the classes *truck* and *van* are excluded from the training set. This exclusion ensures that these classes are classified as *unknown unknowns*, as defined in Sec. 3.2, thereby facilitating the evaluation of the object detection models' capabilities in open set scenarios.

In contrast to the synthetic dataset, the nuImages dataset represents a large-scale, real-world collection of driving scenarios. It encompasses diverse urban and highway traffic scenarios from cities in the USA and Singapore. Given that the number of instances for certain open set classes in the validation set is insufficient to draw meaningful conclusions regarding their open set performance, images from the training set that contain at least one open set

object are added to the validation set. The resulting label distributions for both datasets are provided in Ch. A in the Appendix. Unlike the synthetic dataset, where open set performance is evaluated based on the classes *truck* and *van* – both of which closely resemble the closed set class *car* – the open set classes proposed for the nuImages dataset are selected to encompass a broader range of unknown classes, such as *wheelchair*, *stroller* and *personal mobility*. Overall, this dataset enables the evaluation of detection models in significantly more complex and diverse traffic scenarios.

The nuImages dataset inherently features a hierarchical class structure as introduced in the publication [Cae20]. However, minor modifications are implemented to enhance comparability with the synthetic dataset hierarchy. Specifically, the classes *adult*, *child*, and *construction worker* are consolidated into the category *pedestrian*. Additionally, the superclasses *living*, *commercial*, and *two-wheeled* are incorporated, and the root class object is introduced to satisfy the criteria for a valid hierarchy, as outlined in Sec. 3.3. The final hierarchical structures are depicted in Figs. B.1 and B.2 in the Appendix. Similar to the synthetic dataset, both fine-grained and coarse-grained hierarchical class taxonomies are proposed, with the fine-grained hierarchy serving as the default for all experiments unless otherwise indicated. The primary objective is to assess the impact of the hierarchy on the classifier performance, as detailed in Sec. 6.4.

6.2 Network Architectures

Basic object detection models are not capable of explicitly classifying objects from unknown classes alongside known classes. In order to provide further information about each detected object, an unknown-aware classification head is proposed in this section to classify the detected objects into a predefined hierarchical taxonomy, as shown in Fig. 6.1.

Inspired by the network architectures presented in Sec. 2.3, an unknown-aware hierarchical object detection loss is introduced:

$$\mathcal{L}_{\text{od}}(y, \hat{y}) = \sum_i \left[\mathcal{L}_{\text{h}}(c_i, \hat{c}_i) + \omega_{\text{OD}} \mathcal{L}_{\text{loc}}(l_i, \hat{l}_i) \right] \tag{6.1}$$

where $\mathcal{L}_{\text{h}}$ represents the unknown-aware hierarchical classification loss between the one-hot encoded ground-truth label c_i and the predicted class scores $\hat{c}_i$. $\mathcal{L}_{\text{loc}}$ is the regression loss between the ground-truth bounding box l_i and the predicted bounding box $\hat{l}_i$. The two losses are weighted by ω_{OD}.

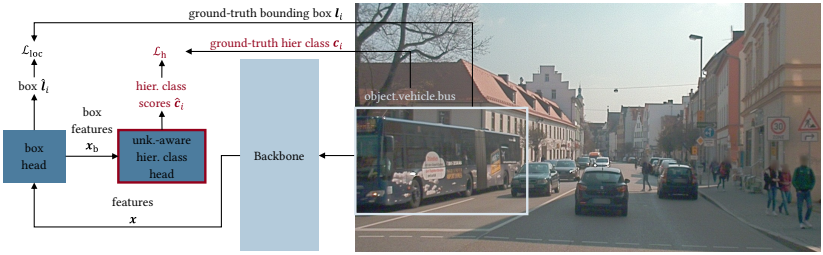


Figure 6.1: The proposed architecture for unknown-aware object detection in this section with the modifications to general object detection models highlighted in red. The hierarchical classification head is trained with hierarchical class labels, e.g., *object.vehicle.bus*.

Hence, the unknown-aware hierarchical object detection is split into two separate tasks, following a two-stage object detector fashion: Localization and classification.

6.2.1 Localization

As introduced in Sec. 2.3, mainstream object detection architectures, e.g., Faster-RCNN and Detection Transformer (DETR), demonstrate their generalization ability and performance on closed set tasks with high precision and recall. In order to achieve unknown-aware object detection, the previously unknown objects need to be localized first, where the existing RPN from

the Faster R-CNN models is required to fulfill the requirements of class-agnostic object detection. It is shown that the two-stage models perform better in an open set setup compared with their one-stage counterparts like YOLO [Dha20]. Despite the absence of the RPN, DETR generates a fixed number of object proposals given a certain input image. During the training of the DETR model, the network implicitly learns to provide such class-agnostic matching by using the Hungarian algorithm to match the ground truth and predicted bounding boxes, encouraging the model to focus on various parts of the image. The generalization ability of classical RPN and transformer models is further investigated empirically in Sec. 6.3.

6.2.2 Classification

As conventional object detection only concentrates on selecting one class from a predefined closed set, new training and evaluation schemes for hierarchical object detection need to be introduced. In this part, three major methods are implemented and compared for the evaluation. Inspired by the work of Li et al. [Li22c], a multi-label binary classification scheme representing the leaf classes and their corresponding categories that incorporate the class taxonomy is chosen. During inference, a confidence-based stop criterion for the prediction is utilized in order to obtain the relevant superclass as prediction from the localized Region-of-Interest of the unknown object. In addition, frequently used relabeling and leave-one-out methods from Lee et al. [Lee18a] are also evaluated. As a baseline, the method with the modifications on the output logits proposed by Hosoya et al. [Hos22] is utilized. This method leverages a simple ratio of top-1/top-2 between the top two class scores from the output logits $\hat{y}$, which is compared with a predefined threshold λ_{ow}.

Hierarchical Constraints (HC)

Following Li et al. [Li22c], hierarchy incorporated classification f_{obj} can be formulated as a multi-label classification problem, where the root-to-leaf path in the class hierarchy $\mathcal{T}$ is uniquely defined. The predicted scores from $\hat{y}_{\text{h}}$

follow the hierarchical constraints for their positive and negative properties [Bi11a, Li22c] concerning their class hierarchy and, therefore, predict the scores $\hat{y} = \text{sigmoid}(f_{\text{obj}}(x))$ based on the N_{bb} detected objects $x \in \mathbb{R}^{N_{\text{bb}} \times D}$, where D is the feature space dimension.

In addition, to further incorporate the class hierarchy, a hierarchical representation learning strategy is utilized as an additional regularization term to ensure that the feature encoder efficiently distinguishes feature representations from various superclasses.

Therefore, the classification head is trained on a combined loss from a binary cross-entropy loss $\mathcal{L}_{\text{hc}}$ that is related to the hierarchy-conform prediction scores and the tree-triplet loss $\mathcal{L}_{\text{tt}}$ on the generated features of the corresponding class i, the positive samples from the positive superclasses i^+ and the negative samples from deviating superclasses i^- that can be formulated as:

$$\mathcal{L}_{\text{h}} = \mathcal{L}_{\text{hc}}(\hat{y}_{\text{hc}}, y_{\text{h}}) + \omega_{\text{tt}} \mathcal{L}_{\text{tt}}(i, i^+, i^-). \tag{6.2}$$

To ensure classes can be appropriately identified as either congruent or incongruent with respect to the extracted feature representations, the inverse similarity ψ of the classes within the hierarchy $\mathcal{T}$ is utilized for calculating the tree-triplet loss:

$$\psi(y,\cdot) = \frac{1}{f_{\mathcal{T}}(y,\cdot)}, \tag{6.3}$$

where the function $f_{\mathcal{T}}$ returns the number of shared superclasses y^+ or y^- with the anchor class y. All classes share at least one *root* class v^r.

During the inference, it is also essential to distinguish if an object should belong to the superclass or its related child classes, as the confidence-based classification condition is defined to predict the class label $\bar{y}_{\text{hc}}$:

$$\bar{y}_{\text{hc}} = \begin{cases} \arg\max_{y' \in \chi(s)} \hat{y}_{\text{hc},y'} & \text{if confident,} \\ s & \text{otherwise,} \end{cases} \tag{6.4}$$

where $\hat{y}_{\text{hc}}$ are the predicted hierarchical class scores, and $\chi(s)$ are the child classes of the current superclass s. This process is repeated until the predicted class label $\bar{y}_{\text{hc}}$ reaches the leaf node or the prediction is not confident enough so that the prediction keeps at the corresponding superclass [Lee18a], where the confidence is measured by the Kullback-Leibler (KL) divergence between the softmax scores and a uniform distribution $U(\cdot|s)$ compared to a defined confidence threshold λ_{U} to determine the confidence:

$$D_{\text{KL}}\Big(U(\cdot|s) \,\Big\|\, \text{softmax}(\hat{y}_{hc}(\chi(s))) \Big) \geq \lambda_{\text{U}}. \tag{6.5}$$

Relabeling (RL)

Object detection models that are trained on a closed set often misclassify previously unknown objects as a known class with high confidence [Dha20]. To mitigate this, relabeling a subset of known leaf class instances to their ancestors during the training phase could help shape the feature representations of the known ancestor classes. This approach should have a negligible effect on the detection performance of the previously known classes, considering that the superclass representations derive from all of its leaf classes. Consequently, these representations inherently increase the robustness of the representations from the superclass as a greater degree of dissimilarity from the known leaf classes is mixed during training.

During training, the known leaf classes in the hierarchy $v \in \mathcal{T}$ are relabeled to their superclasses $s(v)$ in the hierarchy:

$$y_{\text{hr}} = \begin{cases} s(v) & \text{if } b_v = 1, \\ v & \text{otherwise,} \end{cases} \tag{6.6}$$

here b_v is drawn from a binomial distribution Binomial(p), where p defines the relabel probability to the corresponding superclasses. Then, the one hot-encoded label vector $y_{\text{hr}} \in \{0,1\}^{|\mathcal{T}|}$ over all nodes in the hierarchy $\mathcal{T}$ is generated.

Leave-One-Out (LOO)

For comparison, a leave-one-out strategy [Lee18a] is employed where, during the training phase, each superclass $s \in \mathcal{C}_{\text{A}}$ from the set of ancestor classes $\mathcal{C}_{\text{A}}$ within the hierarchy $\mathcal{T}$ is selected. This strategy involves removing the related descendants $\mathcal{C}_{\text{D}}(s)$ from the class taxonomy. Consequently, all labels from the removed descendants are reclassified to the corresponding superclass s. In this situation, the hierarchical classification loss is formulated as:

$$\mathcal{L}_{\text{h}} = \mathcal{L}_{\mathcal{T}} + \sum_{s \in \mathcal{C}_{\text{A}}} \mathcal{L}_{\mathcal{T} \setminus \mathcal{C}_{\text{D}}(s)}, \tag{6.7}$$

where the hierarchical loss consists of the normal cross-entropy loss with all nodes in the hierarchy and the sum of the losses with certain descendant relabels as their ancestor class $\mathcal{L}_{\mathcal{T} \setminus \mathcal{C}_{\text{D}}(s)}$. With the help of the Relabeling and Leave-One-Out strategy, the model is able to learn the general visual properties of the superclasses due to the composed class taxonomies during training.

6.3 Experimental Setup and Results

As introduced in the Sec. 6.2.1, the closed set and open set performance of the two baseline object detection models Faster R-CNN [Ren15] and Deformable DETR [Zhu21] is evaluated. The second step, the evaluation of

hierarchical classification, is based on the model architecture with better generalization ability. The evaluation strictly follows the datasets that are created in Sec. 6.1. During training, a full-disjoint setting is utilized where the unknown classes are not represented. The frames that contain unknown classes are used for evaluation. To avoid data leakage, the image encoder used for the experiments is pre-trained using a self-supervised scheme on ImageNet [Car21], as some of the unknown classes in the setup are given in the vanilla pretraining setup with ImageNet-1k.

The output of the Deformable DETR model has the size $(N_{\text{bb}} \cdot |\mathcal{Y}|)$, where N_{bb} is the number of bounding box predictions and $|\mathcal{Y}|$ is the number of classes. To mitigate the potential issue that one bounding box may be selected several times with different class labels, the sigmoid function at the prediction head is replaced with a softmax function, ensuring that only the class prediction with the highest score is selected. To avoid manual selection of the best score threshold for each model, e.g., based on the Precision-Recall curve computed over several thresholds, a score threshold based on the number of bounding box predictions is implemented with $\text{softmax}(\hat{y}) > 1/N_{\text{bb}}$. Open world object detection models OW- DETR [Gup22] and the modifications introduced in Hosoya et al. [Hos22] for the baseline Deformable DETR model, noted as Deformable DETR*, are utilized for the comparison with other models.

Object Localization

As introduced in Sec. 6.2, the generalization ability of the object detection architectures is evaluated first, as it is essential to generate corresponding bounding boxes for the following classification step from the Faster R-CNN and Deformable DETR. The metrics used for evaluation are the mAP, AR of the known classes, WI, and A-OSE that are introduced in Sec. 3.2. To address the ability that the models have for previously unknown classes, the Average Recall of the unknown classes, noted as AR_{u} is also reported. The evaluation results on the two datasets are shown in Tab. 6.1.

Table 6.1: Localization performance of the two main stream object detection models, Faster R-CNN and Deformable DETR, on known and unknown classes in %. The models are trained in a disjoint setting, where only the known classes are available in the training set.

Model	mAP (↑)	AR (↑)	WI (↓)	A-OSE (↓)	AR_u(↑)
Synthetic Dataset					
F. R-CNN	49.51	64.68	**3.73**	**2 226**	56.58
De. DETR	**60.68**	**70.86**	7.65	2 396	**68.07**
nuImages					
F. R-CNN	57.20	76.30	1.71	11 741	**55.15**
De. DETR	**61.61**	**76.97**	**0.85**	**5 805**	52.06

It can be observed that the Deformable DETR outperforms Faster R-CNN in the closed set setting with a significant advantage in dealing with the synthetic dataset. On the real dataset, nuImages, the Deformable DETR model is less affected by the previously unknown objects indicated by lower WI and fewer absolute error detections. Those observations reinforce the widely held view that the vision transformers only generalize well when trained on large, diverse datasets, whereas the CARLA-based synthetic dataset is limited in both size and variety [Bai21]. However, when data leakage is systematically prevented by dataset design, the Deformable DETR model shows better generalization ability on detecting unknown objects measured by AR_u with more than 10% advantage. Based on the observations, Deformable DETR is chosen as the base architecture for further experiments in the hierarchical classification task.

Hierarchical Classification

As described in Sec. 6.2.2, three strategies for considering class hierarchies are evaluated in this subsection. If not stated otherwise, two empirically selected hyperparameters for the classification head are used for the evaluation. The confidence threshold $\lambda_U = 0.05$ guarantees the balance of the known and unknown class performance on the synthetic dataset and the nuImages dataset, and the relabeling rate p of 15% is selected. In Tab. 6.2, the comparison of the proposed models and the baseline open set/world models is shown. It can be

observed that the OW-DETR model is more robust dealing with the differentiation of the unknown and known classes in unknown-aware object detection, as evidenced by its lower WI, A-OSE scores and the highest closed set mAP on both synthetic and real-world datasets. The model that applies only a simple uncertainty thresholding method achieves a high average recall of the unknown classes in both datasets. Relabeling all uncertain predictions to the unknown also decreases the closed set mAP and AR. With the three proposed hierarchical classification methods, a balance of known and unknown classes can be found without the prerequisite of pretraining on data that include unknown classes in advance. It can be seen that both relabeling-based methods achieve noticeably better performance compared to the confidence-based hierarchical constraint method on the synthetic dataset. This can be explained by the fact that the unknown classes are visually similar to the known classes in this setup. LOO demonstrates the best mAP_u of 7.3%, which is almost double that of the next best model from the hierarchical methods. Compared to Deformable DETR*, LOO shows a better tradeoff between recall and precision evaluated on the unknown classes. In addition, the closed set performance of the two relabeling methods also outperforms the hierarchical constraint method. A detailed analysis can be found in the upcoming subsection, where the hierarchical classification behavior is discussed in detail. Nevertheless, the proposed HC shows its strength when the scenes become more complex, as demonstrated with the nuImages dataset, achieving balanced closed and open set performance.

Table 6.2: Overall evaluation of the open set object detection models based on Deformable DETR model with the synthetic and nuImages dataset in %. **U. in Tr.** denotes unknown classes in training and **Hier.** denotes utilizing class hierarchy.

Model	U. in Tr.	Hier.	mAP (↑)	AR (↑)	mAP_u(↑)	AR_u(↑)	WI (↓)	A-OSE (↓)
Synthetic Dataset								
Def. DETR*			51.25	56.50	0.36	**47.19**	9.75	2 036
OW-DETR	✓		**64.72**	**67.41**	0.12	10.35	**0.30**	**375**
HC		✓	53.09	61.81	0.08	1.39	6.98	2 216
RL		✓	53.18	67.03	4.38	31.00	7.47	2 425
LOO		✓	53.20	65.88	**7.34**	36.91	7.82	2 641
nuImages								
Def. DETR*			44.09	45.83	**1.18**	**63.10**	0.92	775
OW-DETR	✓		**51.98**	56.13	0.005	9.42	**0.15**	**196**
HC		✓	42.20	50.24	1.14	10.54	1.56	2 431
RL		✓	45.65	**68.37**	0.22	10.90	2.82	6 762
LOO		✓	43.94	64.86	0.32	8.69	2.41	6 574

Hierarchical Classification Behavior

By examining the confusion matrices presented in Figs. 6.2 and 6.3, a detailed analysis of the classification error on various hierarchy levels is conducted. In the confusion matrices, all rectangles in dark blue depict the predictions made in the known classes, whereas the red rectangles denote those predictions classified into their corresponding superclasses.

For the synthetic dataset, the hierarchical constraint (HC) is prone to misclassifications into the root class *object* shown by the orange box. In contrast, relabeling (RL) and leave-one-out (LOO) approaches show far less confusion for the *object* class, instead, a higher rate of confusion appears between the two-wheeled vehicles, namely *motorcycle*, and *bicycle*, as indicated by the green rectangle. Moreover, RL and LOO exhibit a more significant number of misclassifications between the known classes and the related superclasses, which are depicted in orange. This discrepancy arises as a byproduct of the relabeling strategy implemented in these models. Nevertheless, RL and LOO achieve a higher number of correctly classified unknown objects, specifically *truck* and *van*, into the most relevant superclass *vehicle*. Across all models, there is a notable confusion between the unknown classes – *truck* and *van* – with the known class *car*, as highlighted in grey at the lower left corner of the confusion matrices.

Turning to the nuImages dataset, similar to the observations from the synthetic dataset results, HC frequently misclassifies known class detections as unknown, as shown in orange. Moreover, the relabeling-based model exhibits confusion within the known classes *truck*, *trailer*, and *bus*, as seen in the bright blue box, while HC erroneously classifies these as the direct superclass *commercial*. In addition to that, HC often misclassifies the unknown class *construction vehicles* into the known class *car* and *truck*. This misclassification trend is observed to extend to the known classes *trailer* and *bus* in RL and LOO, shown in grey. Notably, in all models, most detections from the unknown classes within the *vehicle* category are misclassified into the known classes. In RL and LOO, these misclassifications extend to the superclasses *two-wheeled* and *living*, which can be observed in the orange rectangles at the lower right corner. This phenomenon can also be attributed to specific labeling policies within the nuImages dataset, which will be further elaborated upon in the subsequent section. Consequently, only a limited number of unknowns are accurately classified into their corresponding most related superclasses, as shown in the yellow boxes. Additionally, due to the relabeling strategy in RL and LOO, there is significant confusion between the known classes – *bicycle*, *motorcycle*, and *pedestrian* – and their direct superclasses, *two-wheeled* and *living*, respectively.

In addition, the qualitative results of the proposed unknown-aware hierarchical object detection models are shown in Fig. 6.4 compared with the baseline Deformable DETR, where the unknown class ground truth and predicted bounding boxes are visualized for clarity. The baseline Deformable DETR gives positive predictions for many unknown class objects that reconfirmed the quantitative results above and from Dhamija et al. [Dha20]. In contrast, the proposed hierarchical models are capable of correctly detecting and classifying the unknown class objects, despite some models still misclassifying known classes into the wrong hierarchy level, which is usually considered less critical.

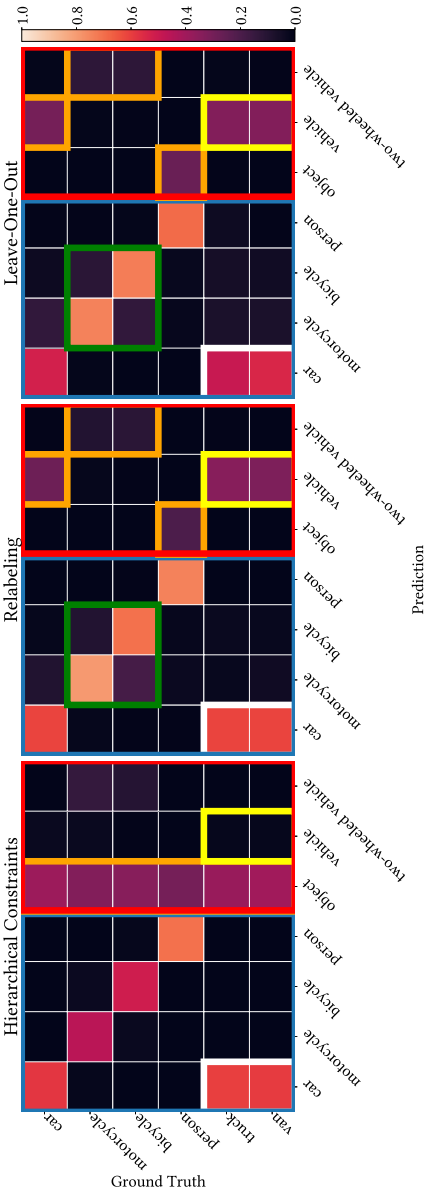


Figure 6.2: Confusion matrices for various unknown-aware training strategies on synthetic dataset. All dark blue rectangles represent predictions associated with previously known classes, while the red rectangles indicate predictions categorized under their respective superclasses.

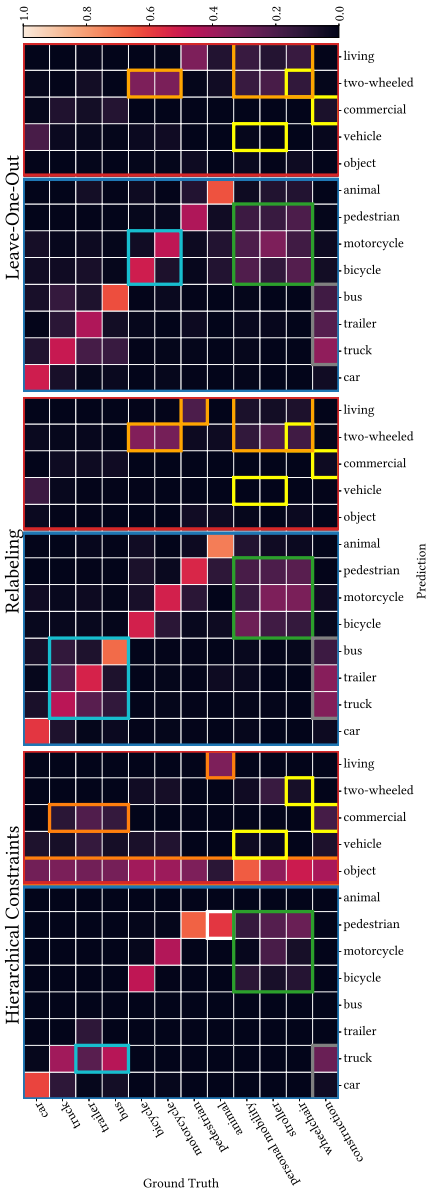


Figure 6.3: Confusion matrices for various unknown-aware training strategies on nuImages dataset. All dark blue rectangles represent predictions associated with previously known classes, while the red rectangles indicate predictions categorized under their respective superclasses.

Figure 6.4: Qualitative results from the validation sets that contain previously unknown classes. The baseline Deformable DETR model predicts only known classes in white bounding boxes, while the unknown-aware hierarchical object detection models show the predicted unknown class instances in red and the ground truth bounding boxes are in green.

6.4 Ablation Study

Influence of Hierarchy

The definition of the class hierarchy is a pivotal element in hierarchical classification, as it directly determines how the classes interact during training and inference. Therefore, the ablation study examines the effect of utilizing different class hierarchies – in addition to the fine-grained hierarchy introduced in Sec. 6.1 – also a coarse-grained hierarchy that eliminates all the superclasses beyond level 2, as depicted in Fig. B.2 in the Appendix. For the ablation study, LOO is selected as it demonstrates superior performance across all tasks without necessitating manual stop criteria selection, which is necessary for the hierarchical constraint method.

The comparison of the models trained on different hierarchical granularities is displayed in Tab. 6.3. With the synthetic dataset, where the diversity of the recorded driving scenes is limited by the available simulation assets and render engine, the fine-grained class hierarchy of known classes enhances the model's ability to detect known and unknown classes, with the exception of differentiating open and closed set classes indicated by slightly worse WI. In the large-scale nuImages dataset, although the inclusion of additional superclasses marginally impacts the closed set mAP and AR, utilizing a fine-grained hierarchy significantly increases the open set performance shown by mAP_u and AR_u and in addition balances the detection of all the classes. Such findings suggest that employing various class hierarchies has a considerable impact on the model's open and closed set performance. A use-case specific class hierarchy is essential to guarantee optimal performance, as the utilized hierarchy is intricately linked to the data characteristics pertinent to concrete driving scenarios, which may vary from urban to highway environments.

Table 6.3: Evaluation of leveraging LOO strategy for unknown-aware object detection using the fine- and coarse-grained class hierarchies in %

Hierarchy	mAP	AR	mAP_u	AR_u	WI	A-OSE
Synthetic Dataset						
Fine	**53.20**	**65.88**	**7.34**	**36.91**	7.82	**2 641**
Coarse	51.70	65.61	7.08	34.57	**7.49**	2 785
nuImages						
Fine	43.94	64.86	**0.32**	**8.69**	**2.41**	**6 574**
Coarse	**45.87**	**67.03**	0.007	1.40	2.47	7 387

Level-1 Comparison

In the classical evaluation scheme with class-based mAP and AR, the advantage of leveraging class hierarchy with the corresponding superclasses is not fully addressed. The models' performance is analyzed based on their first hierarchical level from the root, as shown in Sec. 6.1. For the synthetic dataset, the superclasses are *vehicle* and *person*, while the two superclasses *vehicle* and *living* in nuImages are reported. Since Average Precision and Average Recall are calculated for these superclasses, predictions belonging to descendant classes are mapped to their ancestors. Table 6.4 represents the classification performance together with the confusion matrices in Figs. 6.2 and 6.3, which shows that the HC method exhibits the best average precision based on high hierarchy level, especially with large-scale real datasets like nuImages. However, as observed from the confusion matrices from Figs. 6.2 and 6.3, HC tends to classify objects conservatively, resulting in confusion with the root classes and consequently lower performance when evaluated on leaf nodes $\mathcal{V}_\mu$. Within the nuImages dataset, all models misclassify certain unknown classes, specifically *personal mobility*, *stroller*, and *wheelchair*, into the *living* category. This can be explained by the fact that the nuImages dataset has a labeling policy to treat the vehicles and their rider as a single object.

Table 6.4: Level 1 comparison of the hierarchical object detection models on the synthetic and nuImages dataset.

Model	$\mathbf{mAP_{L1}}(\uparrow)$	$\mathbf{AR_{L1}}(\uparrow)$	AP(↑)	
			vehicle	**person/living**
Synthetic Dataset				
HC	**63.74**	71.32	**67.44**	**60.04**
RL	58.84	72.23	64.18	53.49
LOO	61.11	**73.18**	66.87	55.34
nuImages				
HC	**73.66**	80.90	**78.29**	**69.04**
RL	60.59	83.40	65.63	55.55
LOO	61.97	**84.04**	67.77	56.17

6.5 Conclusion

This chapter addressed the challenge of classifying unknown objects within a predefined hierarchy while being able to detect previously known classes. The proposed hierarchical open set object detection approach aims to identify novel classes that are not encountered during the training process by categorizing them into the most relevant superclasses, where an unknown-aware hierarchical object detection model is introduced. The model comprises two primary tasks similar to the two-stage object detection models: Localization and hierarchical classification. Various training strategies complemented with class hierarchies are examined to assess their efficacy in detecting unknown objects. Experimental results indicated that the proposed method achieves promising performance in both open and closed set scenarios with prior knowledge. Additionally, the experiments underscored the significance of selecting a class hierarchy tailored to the specific use case and the visual proximity of objects.

7 Hierarchical Class Taxonomies as Contextual Knowledge for Semantic Segmentation

While Ch. 6 focused on the detection task, this chapter elaborates on leveraging hierarchical taxonomies for semantic segmentation tasks. In the previous work for evaluating semantic segmentation models, the distinction between a classification error that assigns a similar class compared to its ground truth class with respect to the specified use case and one that assigns a significantly different class is often overlooked. Similar to the previous section, the class hierarchy is utilized in this section to evaluate the open world performance of semantic segmentation models, which leads to the following hypotheses in Research Question 2 discussed in Sec. 7.1.

> **Research Question 2**: How can hierarchical taxonomies be leveraged in downstream computer vision tasks to evaluate and improve the generalization ability of the trained neural networks?

- Hypothesis 2.2: A use case-relevant class taxonomy can facilitate the evaluation of domain generalization and the performance of encountering novel classes.
- Hypothesis 2.3: Appearance-based class hierarchies can be learned implicitly from the given datasets, which helps the neural networks to understand the semantic difference between objects better. However, the class hierarchy for the criticality evaluation is highly use-case relevant.

- Hypothesis 2.4: The usage of hierarchical taxonomy is task-agnostic with respect to the aforementioned computer vision tasks.

In Sec. 7.2, it is further investigated how the class hierarchy can then be leveraged to mitigate corner cases and improve learning efficiency when only a limited amount of data from the novel classes is available, using FSSS methods. Research Question 3 and its hypothesis are formulated as follows:

Research Question 3: Does the class hierarchy reduce the required effort for training in computer vision tasks since the training process becomes more effective and the amount of data needed for training decreases?

- Hypothesis 3: A new class that shares some similar properties can be learned with fewer training samples with consideration of hierarchical class taxonomies.

The content of this chapter is based on the publications of the author in the proceedings of the 2023 IEEE / CVF Computer Vision and Pattern Recognition Workshops [Zho23b] and the 36th IEEE Intelligent Vehicles Symposium [Zho24a].

7.1 CER as a Category-Aware Metric for Domain Generalization

This section introduces a novel evaluation metric that specifically emphasizes inter-category errors. For instance, misclassification between *truck* and *bus* is less critical compared to confusion with a class that is from Vulnerable Road Users (VRUs) category, such as *pedestrian*. As introduced in Sec. 2.4, current evaluation metrics presuppose that the predicted class should exactly match the ground truth, neglecting the distinction between classification errors that are inside and outside of the given category. Although the classical evaluation metrics, e.g., IoU and accuracy, can be calculated based on the object categories instead of the classes, this approach does not provide any insights

into whether a specific class within a certain category exhibits a higher error rate in comparison to other classes. Such property inhibits the evaluation of underrepresented classes in the datasets. To address this limitation, the proposed evaluation metric, termed the *Critical Error Rate* (CER), focuses on the error rates of the predictions that fall outside the ground truth category of class c:

$$\mathrm{CER}_c = \frac{\mathrm{FP}_{\mathrm{out},c} + \mathrm{FN}_{\mathrm{out},c}}{\mathrm{TP}_c + \mathrm{FP}_c + \mathrm{FN}_c} \tag{7.1}$$

$$\text{where } \mathrm{FP}_{\mathrm{out},c} = \sum_{c' \in \mathcal{C} \setminus \mathcal{K}_c} M_{c,c'} \text{ and } \mathrm{FN}_{\mathrm{out},c} = \sum_{c' \in \mathcal{C} \setminus \mathcal{K}_c} M_{c',c} \,,$$

given the confusion matrix M, which has a dimension of $|\mathcal{C}| \times |\mathcal{C}|$ (predictions$\times$ ground truth) and $\mathcal{K}_c$, the set of classes in the same category as c. The numerator in CER includes the sum of false positive and false negative predictions that are outside the ground truth category of class c ($\mathrm{FP}_{\mathrm{out},c} + \mathrm{FN}_{\mathrm{out},c}$), while the denominators remain consistent with the IoU metric to ensure comparability. It can be observed that $\mathrm{CER}_c \in [0,1]$ and $\mathrm{IoU}_c + \mathrm{CER}_c \leq 1, \forall c \in \mathcal{C}$.

In scenarios involving novel classes, where a class has not been encountered during training, the network does not generate predictions for that novel class, so $\mathrm{FN}_c > 0$ and $\mathrm{TP}_c = \mathrm{FP}_c = 0$. In such cases, CER simplifies to represent the proportion of segmentation predictions that do not belong to the correct object category calculated as follows:

$$\mathrm{CER}_c^{\text{novel class}} = \frac{\mathrm{FN}_{\mathrm{out},c}}{\mathrm{FN}_c} = \frac{\sum_{c' \in \mathcal{C} \setminus \mathcal{K}_c} M_{c',c}}{\mathrm{FN}_c} \tag{7.2}$$

Compared to the IoU metric, where the prediction errors are treated equally across the classes, the error rate outside of the category highlights errors deemed more critical due to shifts in object categories in comparison with the ground truth. Furthermore, the CER metric serves as an indicator of the network's generalization capability, as the novel classes are classified into the

same object category given a use-case specific class hierarchy. This evaluation does not require extensive annotation of a novel class set as a prerequisite, facilitating the evaluation of domain generalization.

This section focuses primarily on appearance-driven class taxonomies that are predefined by the semantic segmentation datasets, which can be found in Ch. B in the Appendix. However, specific tasks, such as motion behavior prediction, represent divergent challenges in the definition of class hierarchies, as confusion between *motorcycle* and *bicycle* is more critical due to their differing motion patterns despite their similar semantic properties. Consequently, as shown in the previous section, the class taxonomy used for evaluation should be adjusted to specific use cases rather than being considered as a natural constant. In Sec. 7.1.3, various class hierarchies are examined for robustness, since appearance-based hierarchies may not always represent safety considerations.

7.1.1 Experimental Setups

The aim of the experiments is to assess the impact of employing various image encoders and decoders on domain generalization performance. Three different setups are utilized to quantify differences:

- On **Source Domain:** At the first step, domain generalization performance is evaluated solely on the source domain to exhibit the inherent properties of intra- and inter-category confusion without any domain shifts. In this setup, models are trained and evaluated independently using two mainstream datasets, Cityscapes or Mapillary, focusing on the comparison of IoU and CER for the categories *human* and *vehicle*.
- Under **Domain Shifts:** The visual appearance of objects changes but the list of the available classes remains consistent. In this context, Cityscapes and Mapillary serve as the source datasets, with evaluations performed on class instances that share definitions across the A2D2, ACDC, and BDD100k datasets.
- With **Novel Classes:** In this setup, an open world heterogeneous setup is considered, where the domain changes and the appearance of new

classes emerge simultaneously. Similar to the domain shift setup, networks are trained on the two source datasets and evaluated on the A2D2 and Mapillary datasets, with particular emphasis on the novel classes.

Unless stated otherwise, experiments are repeated three times to calculate average values and standard deviations. For the image encoders, three different sizes from the ResNet family that are frequently used in computer vision research are investigated. In addition, due to the rapid advancement of current transformers or heavily transformer-inspired models, ConvNeXt [Liu22b], Swin Transformer [Liu21b] and SegNeXt [Guo22] encoders with corresponding network sizes paired with their ResNet counterparts are analyzed. For the image decoders for segmentation, the frequently-used ASPP head from DLv3+ [Che18] and the UPerNet decoder head [Xia18], which is frequently paired with current state-of-the-Art encoder architectures, are compared as benchmarks. Additionally, simple and frequently used decoder heads such as FCN [Lon15] and the pyramid pooling module [Zha17] are considered in the upcoming experiments. The training sessions employ the same number of iterations and network parameters across the datasets to ensure comparability. The SGD optimizer with a base learning rate of 1×10^{-2} is utilized for all the networks with ResNet backbones and the AdamW optimizer with a base learning rate of 1×10^{-4} for other backbones. The training is scheduled with a linear warm-up policy with 1500 iterations. The backbones are pretrained with ImageNet-1k [Rus15a]. A batch size of 8 is used for the training across the datasets to train the network with 80k iterations with crop size of 1024×512. During training, common data augmentation methods like random flip and photometric distortion [MMS20] are applied to increase the data variety. For simplification, methods like auxiliary head or stage-wise learning rate decay are not applied during training, although they may contribute to better network generalization or training stability. Since there exists higher uncertainty in the networks when the input data distribution drifts away from the source, the training sessions are repeated three times and the average values of each class are reported.

7.1.2 Experiments

Source Domain

Table 7.1 depicts the results of directly evaluating on the Cityscapes dataset, noted as C, after training with various backbones and segmentation heads. It can be observed that the CER negatively correlates with the IoU, which is inferred from the fact that the absolute error made by a network decreases as the segmentation performance improves as illustrated in Eq. (7.1). Due to this characteristic, most models exhibit comparable CER values below 15% on the Cityscapes dataset, except models with a small backbone and with FCN decoders, where the absence of high-quality multi-scale features also leads to lower IoU on the source domain. In addition, it can be observed that semantic segmentation models with smaller backbones bias towards out-of-category classification when a prediction error is made, which suggests that such errors are significantly influenced by the quality of the feature representations generated by the image encoder.

In contrast, the SegNeXt-S model, although achieving IoU performance comparable to the ResNet-101 backbone combined with the ASPP decoder, exhibits a higher rate of out-of-category errors. This indicates that when a classification error occurs, the SegNeXt-S model is more likely to miscategorize a pixel to a different object category, posing a greater safety risk.

Table 7.1: Source (C)→Target (C): Evaluation of semantic segmentation models trained on Cityscapes without domain shift. The class IoUs and class CERs are reported over three runs.

Backbone	Decoder	IoU% ↑									CER% ↓								
		person	rider	car	truck	bus	train	m-bike	bicycle	mean	person	rider	car	truck	bus	train	m-bike	bicycle	mean
ResNet-18	FCN	80.1	57.3	93.5	46.4	72.7	46.3	53.9	76.2	70.7	16.4	22.7	4.3	22.5	8.9	25.4	23.2	20.4	16.3
	PSPNet	79.2	54.5	94.2	66.9	82.2	70.8	56.5	76.3	74.0	16.8	22.3	4.5	19.1	10.8	19.7	22.5	20.3	15.8
	ASPP	81.4	60.6	94.7	73.7	84.8	74.2	61.6	76.9	76.0	15.6	22.7	4.2	11.8	7.4	15.6	21.7	19.8	14.3
SegNeXt-T	HamHead	81.5	60.7	95	81.4	88.6	77.8	64.1	77.0	78.2	15.6	23.6	4.1	9.1	6.8	15.0	20.4	19.9	13.8
SegNeXt-S		83.1	64.1	95.4	82.5	91.4	82.4	69.0	78.7	80.0	14.4	22.0	3.8	7.8	5.8	14.0	18.7	18.6	12.9
ResNet-50	FCN	82.8	62.4	94.3	54.2	73.3	47.5	61.6	79.2	73.0	14.1	20.2	3.7	18.5	8.0	22.9	19.5	18.1	14.7
	PSPNet	82.5	61.8	95.3	75.7	87.6	80.9	66.7	79.0	77.5	14.4	20.3	3.9	14.8	8.2	14.7	19.2	18.3	13.9
	ASPP	84.0	66.0	95.7	79.9	88.9	80.6	69.3	79.5	79.3	13.6	20.0	3.5	10.8	5.7	12.6	18.3	17.8	12.7
ResNet-50	UPerNet	82.6	61.2	95.3	78.7	87.7	76.8	67.8	78.9	77.9	14.1	20.8	3.8	12.5	7.6	16.1	18.2	18.3	13.5
Swin-T		82.9	62.5	95.4	80.7	88.8	79.3	68.2	78.7	79.0	14.2	21.6	3.7	9.4	6.1	13.8	18.4	18.3	12.9
ConvNeXt-T		84.0	64.7	95.8	86.0	91.8	83.5	70.3	80.3	80.8	13.2	19.9	3.5	7.0	5.3	12.2	17.0	17.0	11.9
SegNeXt-B	HamHead	84.6	67.2	95.9	88.1	93.0	86.8	72.7	80.3	81.9	13.2	20.3	3.4	5.6	4.9	10.5	16.2	17.2	11.6
ResNet-101	FCN	83.3	64.4	94.8	61.9	81.8	60.3	62.7	79.1	75.7	13.9	20.2	3.6	13.1	6.1	12.4	18.6	18.2	13.3
	PSPNet	83.6	64.5	95.6	80.9	89.6	79.5	69.3	79.6	78.8	13.7	20.2	3.7	11.3	5.9	12.9	17.8	17.8	13.0
	ASPP	84.4	67.0	95.8	79.4	89.8	82.8	70.3	80.2	79.9	13.2	19.6	3.5	9.1	5.4	12.4	17.4	17.4	12.3
ResNet-101	UPerNet	83.3	63.5	95.4	77.7	88.2	79.4	67.8	79.2	78.6	13.9	21.1	3.7	11.4	7.3	15.7	18.9	18.1	13.3
Swin-S		84.2	65.0	95.9	83.4	91.6	85.8	71.1	80.1	81.0	13.2	20.1	3.4	9.6	5.2	10.9	17.2	17.3	12.0
ConvNeXt-S		84.8	66.6	95.9	84.3	92.2	86.3	72.5	81.1	81.8	12.6	18.9	3.3	5.8	5.1	11.6	16.0	16.4	11.2
SegNeXt-L	HamHead	85.2	68.1	96.1	88.5	93.4	88.4	73.6	80.8	82.5	12.6	19.5	3.3	5.5	4.7	10.0	15.6	16.8	11.2

Table 7.2 depicts the evaluation of three mainstream models with various backbone and decoder architectures on the Mapillary dataset, noted as M. These models represent classical CNN-driven semantic segmentation networks based on ResNet backbones, modernized CNN models like ConvNeXt, and the current state-of-the-art segmentation models on Cityscapes (SegNeXt). Compared to the Cityscapes dataset, the Mapillary dataset has a significantly greater variety of available classes. Despite its complexity, it can be observed that the mean IoU and mean CER are still negatively correlated. For those classes that are underrepresented in the training set, such as *other rider*, *trailer*, and *rails*, the models achieve relatively low IoUs, especially for the class *other rider*, even the model with the highest learning capacity yields an IoU of 0% on this class. Similarly, for the class *trailer*, only models with modern architectures are able to segment this class correctly, showing an IoU value up to 10%. It is posited that the class decision boundary is influenced not only by the class itself but also by other classes sharing similar semantic or visual properties within the same object category. An increase in the network's learning capacity also correlates with a reduction in the Critical Error Rate.

For the *trailer* class, as reflected in the CER values, only a small fraction of the predictions deviate from the *vehicle* category. Furthermore, although the *other rider* class shows low IoUs, this does not necessarily mean that the predictions are completely unreliable, as a large amount of detections still land in the *human* category with other classes like *pedestrian*, *bicyclist*, and *motorcyclist*. Similar patterns emerge with the classes, e.g., *other vehicles* and *rails*.

The analysis of results across Cityscapes and Mapillary indicates that the IoU metric reliably reflects network performance when object instances are well distributed across the classes and sufficient training data are available. In datasets that confront the long-tail distribution problem with underrepresented classes, the CER metric emerges as an important supplementary metric for evaluating the performance of semantic segmentation models.

Table 7.2: Source (M)→Target (M): Evaluation of semantic segmentation models trained on Mapillary Vista without domain shift. The class IoUs and class CERs are reported over three runs.

Backbone	Decoder	IoU% ↑									CER% ↓								
		person	bicyclist	o. rider	car	o. vehicle	trailer	truck	rails	mean	person	bicyclist	o. rider	car	o. vehicle	trailer	truck	rails	mean
ResNet-18	ASPP	70.2	35.6	0.0	90.3	0.9	0.0	64.3	0.0	36.9	26.0	27.3	46.8	6.8	29.2	7.5	14.9	30.7	33.4
ResNet-50		75.4	57.6	0.0	92.1	15.8	0.0	73.6	17.4	44.2	21.8	21.6	45.6	5.9	24.7	6.8	10.9	31.1	29.6
ResNet-101		76.3	57.2	0.0	92.5	21.5	0.0	73.7	39.1	45.5	21.1	21.1	44.1	5.6	24.6	7.6	11.1	24.2	29.0
ConvNeXt-T	UPerNet	75.3	61.0	0.0	91.8	23.5	0.0	73.8	48.8	48.0	21.9	20.4	44.2	6.1	24.8	9.0	11.8	23.4	28.4
ConvNeXt-S		76.8	63.9	0.0	92.4	27.1	9.4	75.9	59.8	50.8	20.7	19.0	47.5	5.7	22.7	10.1	9.6	21.6	26.9
ConvNeXt-B		78.0	67.4	0.0	92.6	34.4	10.1	76.4	64.4	51.5	20.0	18.9	37.4	5.6	21.9	11.0	10.0	23.2	26.6
SegNeXt-S	HamHead	71.2	58.7	0.0	90.8	26.0	0.1	73.7	49.1	46.1	26.4	23.1	46.3	7.2	27.3	8.3	12.0	21.8	30.8
SegNeXt-B		73.6	63.0	0.0	91.6	31.6	9.6	77.2	61.8	49.5	24.2	21.5	43.6	6.8	23.6	14.3	11.0	22.5	28.2
SegNeXt-L		75.1	65.4	0.0	92.1	31.0	8.2	77.9	67.9	51.5	22.9	20.1	39.2	6.4	23.4	15.7	10.1	20.3	27.5

Domain Shift

In this setup, the generalization ability of the networks is assessed using known classes across three distinct target datasets: ACDC, which is recorded in Switzerland with diverse weather conditions (noted as AC), A2D2, which is collected in Germany with higher class granularity (noted as A2), and BDD100k from the United States (noted as B). Given that ACDC and BDD100k share the same label space with Cityscapes, models that are trained on Cityscapes can directly infer on the images from two target domains without any adaptation. The corresponding class- and category-specific metrics can be calculated subsequently. In order to eliminate the potential impact of network uncertainties, if not stated otherwise, the training sessions for each network and dataset configuration are repeated three times with specified seeds from the identical ImageNet-pretrained backbones. Average metrics and their standard deviations are reported in Tab. 7.3 and Tab. 7.4.

Tab. 7.3 presents the zero-shot semantic segmentation performance of eight distinct network architectures, containing widely-used classical CNNs, transformer models, and transformer-inspired CNN models individually paired with two decoder heads: ASPP head and UPerNet head, with both heads supporting multi-level feature inputs. For the selection of evaluated classes, this section concentrates on underrepresented classes, in addition to the frequently observed class *person*. It can be seen that the performance of the models degrades from their source domain, as shown in Tab. 7.1. For certain classes, for instance, *train* shows high standard deviations, indicating the ambiguity of the learned class representations from the source dataset.

On the ACDC dataset, transformer and transformer-inspired backbones outperform the classical CNN models, which demonstrates their capability to learn more robust and generalized features as classes like *person* by outperforming the equivalent architectures with 15% IoU and 10% CER, where sufficient training samples are available in the training set. Besides that, although ConvNeXt-Tiny and DeepLabv3 models with ResNet-50 backbone have similar performance on their source dataset, the Cityscapes, the

ConvNeXt model shows 14% absolute mIoU and 12% absolute mCER improvement on the ACDC dataset. Furthermore, such models with modern image encoders also eliminate the detection errors that are located out of the ground-truth category, observed by class *truck* that has 93.6% CER from the ResNet-50 image encoder to 32% CER with ConvNeXt-Small. Similar results are observed on the BDD100k dataset. Due to the divergent data distribution of the class *train*, most models show poor performance for this class, as indicated by low IoU scores in the dataset. However, most of the segmentation predictions are still in the same object category, *vehicle*, rather than other categories. A similar phenomenon is observed for the other vehicle classes such as *truck* and *bus*: Despite the IoU values remain low compared with those on ACDC dataset, where false predictions predominantly fall into other object categories, the majority of predictions on BDD100k dataset still belong to the *vehicle* category, as indicated by the CER values. The different appearances of certain classes and deviated labeling strategies may also be attributed to this discrepancy, as pickups are labeled as *car* in Cityscapes but annotated as *truck* in BDD100k. In addition, as observed in the evaluation on Mapillary dataset without domain shift, increased learning capacity leads to lower critical error rates of the underrepresented classes. With a similar number of parameters, modern model architectures using vanilla training pipelines also outperform the classical CNN models with dedicated domain generalization optimizations, as reported by Choi et al. [Cho21].

Given that the A2D2 dataset has a completely different label space compared with the aforementioned datasets, a subset of overlapping classes with common semantics is selected for comparison and remapped to the Cityscapes evaluation scheme. Domain shift experiments emphasize two major classes, namely, *car* and *truck*, as both datasets are recorded in Germany, leading to a similar semantic representation of both classes.

For the evaluation, the models are pretrained on the Cityscapes or Mapillary dataset with identical training setups, including the number of training iterations, batch size, and learning rate, for a fair comparison. Table 7.4 presents the evaluation of the generalization ability of those models that are pretrained

on either dataset and evaluated on the A2D2 dataset. It is evident that networks using a small image encoder (ResNet-18) trained on the Mapillary dataset outperform larger models (ResNet-101) trained on Cityscapes. Similarly, the smaller ConvNeXt-Tiny backbone models trained on the Mapillary dataset outperform the much more sophisticated ConvNeXt-Base models with higher IoUs, lower error rates, and reduced standard deviations during evaluation. These findings are consistent with recent work by Piva et al. [Piv23], showing the necessity of advanced pretraining for domain generalization performance. Additionally, the superior generalization capabilities of transformer-inspired models are confirmed [Wan23c, Bai21], as ConvNeXt-based models outperform classic ResNet-based models regardless of the training dataset employed for pretraining.

Qualitative results of domain shift cases can be found in the rows a) and b) of Fig. 7.1. Row a) illustrates a case where some models fail to segment an instance of the class *bus* correctly. Nevertheless, the associated CER value indicates that the majority of the pixels are still assigned to the appropriate category, namely *vehicle*. The four models exhibit comparable behavior and achieve similar performance from the aspect of out-of-category error. Row b) depicts a scenario in which the class *truck* is interpreted differently in the BDD100k dataset compared with the Cityscapes dataset. Although the IoU for this class is very low, most of the pixels belonging to the instance are nonetheless classified as *car*, which belongs to the same class category as *truck*. This outcome stems from a shift in the class definition of the two datasets, i.e., a concept shift.

Table 7.3: Source (C)→Target (AC, B): Zero-shot evaluation of models trained on Cityscapes, evaluated on ACDC and BDD100k. Color gradients are utilized to visualize the IoU shifts within each class across the two datasets. The reported values are based on three runs.

		ACDC					BDD100k				
	Architecture	person	truck	bus	train	mean	person	truck	bus	train	mean
IoU%	DLv3+ R50	$38.1_{\pm 4.4}$	$4.9_{\pm 1.5}$	$31.4_{\pm 11.9}$	$30.0_{\pm 8.4}$	$37.6_{\pm 1.7}$	$45.2_{\pm 21.3}$	$21.8_{\pm 2.9}$	$15.8_{\pm 4.7}$	$0.2_{\pm 0.2}$	$42.9_{\pm 0.9}$
	UPerNet R50	$43.2_{\pm 1.8}$	$2.5_{\pm 2.1}$	$40.3_{\pm 7.7}$	$32.2_{\pm 1.8}$	$35.1_{\pm 0.2}$	$56.3_{\pm 1.6}$	$16.4_{\pm 2.0}$	$15.7_{\pm 6.6}$	$0.0_{\pm 0.0}$	$42.1_{\pm 1.2}$
	DLv3+ R101	$47.7_{\pm 5.7}$	$11.6_{\pm 9.6}$	$38.5_{\pm 25.6}$	$28.9_{\pm 6.5}$	$40.3_{\pm 2.9}$	$59.1_{\pm 0.5}$	$28.2_{\pm 1.3}$	$24.5_{\pm 6.4}$	$0.2_{\pm 0.2}$	$45.9_{\pm 0.9}$
	UPerNet R101	$46.4_{\pm 0.2}$	$2.3_{\pm 2.1}$	$33.6_{\pm 4.0}$	$24.8_{\pm 15.3}$	$35.8_{\pm 2.8}$	$57.4_{\pm 1.0}$	$22.6_{\pm 1.0}$	$22.1_{\pm 2.7}$	$0.1_{\pm 0.1}$	$45.2_{\pm 0.7}$
	UPerNet Swin-T	$39.6_{\pm 3.9}$	$18.5_{\pm 5.2}$	$44.4_{\pm 1.7}$	$36.7_{\pm 3.7}$	$39.4_{\pm 0.9}$	$54.8_{\pm 2.2}$	$26.7_{\pm 1.6}$	$31.8_{\pm 5.1}$	$0.0_{\pm 0.0}$	$43.7_{\pm 0.8}$
	UPerNet Conv-T	$52.1_{\pm 0.9}$	$41.4_{\pm 4.0}$	$44.0_{\pm 2.1}$	$57.7_{\pm 1.7}$	$49.8_{\pm 0.4}$	$63.1_{\pm 0.4}$	$38.6_{\pm 0.6}$	$31.0_{\pm 3.4}$	$0.1_{\pm 0.1}$	$51.2_{\pm 0.3}$
	UPerNet Swin-S	$45.6_{\pm 2.8}$	$41.5_{\pm 7.2}$	$60.9_{\pm 8.2}$	$40.5_{\pm 1.9}$	$47.1_{\pm 1.3}$	$59.9_{\pm 0.8}$	$30.5_{\pm 2.4}$	$34.5_{\pm 9.7}$	$0.2_{\pm 0.2}$	$48.0_{\pm 0.9}$
	UPerNet Conv-S	$60.3_{\pm 0.5}$	$52.1_{\pm 11.0}$	$52.3_{\pm 3.2}$	$57.3_{\pm 5.5}$	$54.5_{\pm 1.1}$	$66.7_{\pm 0.4}$	$39.6_{\pm 1.0}$	$38.6_{\pm 3.6}$	$0.4_{\pm 0.1}$	$55.4_{\pm 0.3}$
CER%	DLv3+ R50	$60.4_{\pm 5.3}$	$90.8_{\pm 2.0}$	$49.3_{\pm 12.4}$	$48.8_{\pm 11.8}$	$47.4_{\pm 1.2}$	$53.6_{\pm 22.1}$	$46.7_{\pm 8.7}$	$22.2_{\pm 5.9}$	$30.7_{\pm 13.7}$	$36.2_{\pm 1.0}$
	UPerNet R50	$56.1_{\pm 1.7}$	$93.6_{\pm 4.7}$	$42.1_{\pm 12.3}$	$33.6_{\pm 14.6}$	$49.3_{\pm 0.6}$	$43.2_{\pm 1.6}$	$50.9_{\pm 3.2}$	$23.0_{\pm 6.4}$	$14.8_{\pm 2.4}$	$34.9_{\pm 1.3}$
	DLv3+ R101	$51.1_{\pm 6.0}$	$79.1_{\pm 14.7}$	$46.3_{\pm 31.1}$	$43.9_{\pm 3.4}$	$44.9_{\pm 2.4}$	$40.3_{\pm 0.5}$	$32.9_{\pm 3.4}$	$25.1_{\pm 12.4}$	$18.9_{\pm 8.1}$	$33.0_{\pm 0.6}$
	UPerNet R101	$52.9_{\pm 0.2}$	$94.0_{\pm 4.8}$	$58.1_{\pm 3.9}$	$51.9_{\pm 32.1}$	$50.4_{\pm 3.7}$	$41.9_{\pm 1.2}$	$42.5_{\pm 2.4}$	$28.2_{\pm 5.6}$	$27.4_{\pm 13.4}$	$34.2_{\pm 1.5}$
	UPerNet Swin-T	$58.7_{\pm 4.3}$	$73.4_{\pm 6.6}$	$41.4_{\pm 5.1}$	$47.1_{\pm 4.6}$	$45.6_{\pm 0.5}$	$41.0_{\pm 0.5}$	$40.4_{\pm 3.2}$	$16.4_{\pm 3.0}$	$27.3_{\pm 6.4}$	$32.4_{\pm 1.3}$
	UPerNet Conv-T	$45.8_{\pm 1.4}$	$43.6_{\pm 11.2}$	$47.3_{\pm 5.2}$	$34.1_{\pm 2.8}$	$36.8_{\pm 1.0}$	$35.4_{\pm 0.3}$	$22.0_{\pm 0.3}$	$10.3_{\pm 0.5}$	$19.2_{\pm 4.8}$	$27.1_{\pm 0.4}$
	UPerNet Swin-S	$53.1_{\pm 2.1}$	$45.2_{\pm 4.7}$	$30.8_{\pm 10.3}$	$49.7_{\pm 4.5}$	$38.6_{\pm 1.1}$	$38.9_{\pm 0.3}$	$30.1_{\pm 5.9}$	$14.2_{\pm 6.5}$	$25.5_{\pm 13.3}$	$30.0_{\pm 1.3}$
	UPerNet Conv-S	$38.5_{\pm 0.4}$	$32.0_{\pm 11.0}$	$30.7_{\pm 3.9}$	$33.8_{\pm 6.9}$	$31.5_{\pm 0.9}$	$32.1_{\pm 0.3}$	$22.1_{\pm 2.3}$	$7.9_{\pm 1.6}$	$11.9_{\pm 1.7}$	$23.5_{\pm 0.2}$

Table 7.4: Source (C/M)→Target (A2): Evaluation of models trained on Cityscapes and Mapillary, evaluated on two classes from A2D2 dataset that share the same label space. The reported values are based on three runs.

Setup		Car		Truck	
		IoU%	CER%	IoU%	CER%
Cityscapes	R18	$88.8_{\pm 1.3}$	$5.0_{\pm 0.1}$	$49.8_{\pm 8.1}$	$24.1_{\pm 7.0}$
	R50	$89.0_{\pm 1.1}$	$5.7_{\pm 0.7}$	$54.2_{\pm 9.9}$	$24.5_{\pm 6.0}$
	R101	$88.8_{\pm 2.0}$	$5.0_{\pm 1.3}$	$53.1_{\pm 9.6}$	$17.4_{\pm 4.1}$
	Conv-T	$91.8_{\pm 0.4}$	$4.0_{\pm 0.2}$	$62.5_{\pm 1.0}$	$14.9_{\pm 1.1}$
	Conv-S	$92.5_{\pm 0.4}$	$3.9_{\pm 0.3}$	$69.3_{\pm 0.3}$	$9.9_{\pm 0.7}$
	Conv-B	$93.6_{\pm 0.1}$	$3.6_{\pm 0.2}$	$69.4_{\pm 1.8}$	$10.6_{\pm 1.8}$
Mapillary	R18	$92.5_{\pm 0.3}$	$4.1_{\pm 0.2}$	$62.8_{\pm 2.0}$	$14.0_{\pm 2.8}$
	R50	$93.9_{\pm 0.2}$	$3.5_{\pm 0.1}$	$72.0_{\pm 1.7}$	$7.3_{\pm 0.8}$
	R101	$94.3_{\pm 0.2}$	$3.4_{\pm 0.2}$	$73.7_{\pm 0.3}$	$6.2_{\pm 0.6}$
	Conv-T	$94.2_{\pm 0.0}$	$3.4_{\pm 0.1}$	$74.0_{\pm 0.3}$	$5.3_{\pm 0.3}$
	Conv-S	$94.7_{\pm 0.1}$	$3.3_{\pm 0.1}$	$76.0_{\pm 0.3}$	$5.0_{\pm 0.6}$
	Conv-B	$94.6_{\pm 0.1}$	$3.2_{\pm 0.1}$	$75.1_{\pm 0.2}$	$4.6_{\pm 0.3}$

Novel Class

The most challenging step – evaluating the performance in previously unknown classes – is further elaborated in this part. Note that such evaluation is hardly feasible in the classical evaluation scheme utilizing traditional metrics like Accuracy and IoU, as the presence of true positive predictions is set as a prerequisite. The evaluation of CER aims to determine whether the model correctly infers the class category of an unseen object that belongs to a known object category that was included during training on the source dataset.

Zero-shot evaluation was conducted using the A2D2 and Mapillary datasets under a novel class setup, employing semantic segmentation networks previously trained on either the Cityscapes or Mapillary datasets. The CER values for the novel classes *utility vehicles* and *tractors* from the A2D2 dataset, as well as *other rider*, *caravan*, *other vehicle*, and *wheeled slow* from the Mapillary dataset, are reported. All the aforementioned classes, except for *other rider*, which is classified under the *human* category, fall under the *vehicle* category in both the A2D2 and Mapillary datasets. For reference, the mIoU values are noted as performance indicators on the source domain. The average CER and standard deviation of the relevant classes are reported to demonstrate the variability

and stability across different seeds. Despite differing and inconsistent labeling policies across various datasets, the CER metric makes the evaluation and comparison of semantic segmentation models possible. Additionally, based on the observations from Tab. 7.4, networks previously trained on the Mapillary dataset are also evaluated on the A2D2 dataset to observe differences in model performance arising from variations in the source datasets.

Table 7.5 shows the comparison of the semantic segmentation models dealing with previously unknown classes. Similar to the domain shift setup, the models exhibited different levels of generalization ability despite their comparable performance on the source domains, Cityscapes, and Mapillary. Aligned with the observations from Tab. 7.4, models trained on large-scale datasets with high variety demonstrate impressive generalization capabilities, despite achieving lower mean IoU compared with the Cityscapes dataset observed by their IoU on the source domain, the Mapillary dataset. With identical network architecture, segmentation networks that are previously trained on Mapillary have significantly lower prediction errors that are located outside of the ground-truth category shown by the two novel classes *utility vehicles* and *tractor*.

In addition, consistent with previous observations on domain shift and on the source domain, modern CNN and transformer models surpass ResNet-based architectures by achieving lower CER and exhibiting smaller standard deviations. With increased learning capacity, these networks are able to implicitly differentiate objects across various categories based on visual appearance, as such differentiation is not explicitly designated as a learning objective during training.

Furthermore, it can be observed from the segmentation networks with UPerNet decoder that incorporating effective feature extraction from the images is beneficial for domain generalization. However, ResNet-based models exhibit greater performance variations, whereas modern backbones are less affected. One possible assumption is that the introduction of new normalization layers can mitigate such negative effects. ConvNeXt and Swin Transformer exhibit different behaviors when encountering novel objects. Swin models have a significantly higher error rate in the *vehicle* classes compared to the ConvNeXt

counterparts despite both networks being trained under identical configurations. Although the SegNeXt models achieve state-of-the-art performance on the Cityscapes dataset, there is no guarantee that the learned knowledge can be directly transferred to previously unknown domains and classes. Finally, as observed in the experiments, the proposed CER can also be used as an indicator of dataset variability, even when based on the CER scores for a single semantic segmentation network.

Figure 7.1 also presents qualitative results for previously unknown classes, illustrating images that contain object classes entirely absent from the training set. As discussed in this section, the IoU for the novel classes is zero because no true positive predictions exist. Therefore, only the corresponding CER scores for these novel classes are reported beneath each image. In row c), although the class *caravan* does not appear in the training data, the DeepLabV3+ and SegNeXt models correctly assign it to the *vehicle* category, whereas the Swin-Tiny model segments it as *building* from the category *background*. When the segmentation results still correspond to the same object category, such segmentation errors are deemed less critical. A comparable phenomenon can be observed for the *utility vehicle* instances in row d).

Table 7.5: Source (C/M)→Target (A2/M): Zero-shot evaluation on six novel classes in an open-world semantic segmentation setup, class CER values and their standard deviations over three runs are reported. As reference, their mIoUs in source domain are provided for comparison.

Backbone	Decoder	Dataset	Source mIoU%	A2D2 CER% ↓		Mapillary CER% ↓			
				Util. Vehicles	Tractor	Other Rider	Caravan	Other Vehicle	Wheeled Slow
ResNet-18	FCN	CS	70.7	$62.3_{\pm 3.6}$	$63.7_{\pm 12.0}$	$30.6_{\pm 2.9}$	$38.8_{\pm 8.9}$	$35.2_{\pm 2.2}$	$63.1_{\pm 2.6}$
	ASPP	CS	76.0	$56.7_{\pm 5.6}$	$56.7_{\pm 8.7}$	$33.2_{\pm 2.3}$	$20.4_{\pm 7.4}$	$31.5_{\pm 2.8}$	$66.5_{\pm 5.6}$
		MV	36.9	$48.4_{\pm 3.5}$	$18.7_{\pm 6.6}$	-	-	-	-
ResNet-50	FCN	CS	73.0	$71.7_{\pm 3.1}$	$83.8_{\pm 5.1}$	$32.3_{\pm 0.9}$	$40.0_{\pm 5.3}$	$36.4_{\pm 2.7}$	$66.2_{\pm 0.7}$
	ASPP	CS	79.3	$53.1_{\pm 3.2}$	$76.7_{\pm 2.3}$	$34.5_{\pm 3.2}$	$34.5_{\pm 30.2}$	$31.6_{\pm 2.3}$	$61.2_{\pm 1.7}$
		MV	44.2	$35.8_{\pm 2.1}$	$9.5_{\pm 0.0}$	-	-	-	-
ResNet-50	UPerNet	CS	77.9	$57.3_{\pm 4.1}$	$59.0_{\pm 19.2}$	$44.8_{\pm 3.5}$	$52.0_{\pm 10.1}$	$32.1_{\pm 1.5}$	$67.1_{\pm 4.7}$
Swin-T		CS	79.0	$51.6_{\pm 9.8}$	$73.5_{\pm 6.9}$	$29.9_{\pm 0.5}$	$70.6_{\pm 16.5}$	$28.3_{\pm 3.6}$	$63.7_{\pm 2.0}$
ConvNeXt-T		CS	80.8	$35.6_{\pm 6.1}$	$45.3_{\pm 4.2}$	$23.0_{\pm 3.5}$	$59.1_{\pm 13.3}$	$28.7_{\pm 1.5}$	$65.2_{\pm 1.6}$
		MV	48.0	$24.8_{\pm 2.1}$	$7.4_{\pm 0.3}$	-	-	-	-
SegNeXt-B	HamHead	CS	81.9	$31.1_{\pm 1.7}$	$28.4_{\pm 3.5}$	$30.5_{\pm 3.5}$	$27.0_{\pm 14.0}$	$34.3_{\pm 1.3}$	$64.8_{\pm 2.5}$
ResNet-101	FCN	CS	75.7	$55.3_{\pm 4.1}$	$65.5_{\pm 11.5}$	$27.9_{\pm 2.0}$	$30.9_{\pm 23.2}$	$32.3_{\pm 1.8}$	$67.1_{\pm 4.9}$
	ASPP	CS	79.9	$46.3_{\pm 2.7}$	$58.4_{\pm 18.2}$	$31.4_{\pm 7.6}$	$32.7_{\pm 34.1}$	$33.6_{\pm 1.2}$	$68.2_{\pm 5.4}$
ResNet-101	UPerNet	CS	78.6	$52.5_{\pm 17.1}$	$57.5_{\pm 28.2}$	$32.7_{\pm 5.2}$	$54.7_{\pm 30.0}$	$35.4_{\pm 6.1}$	$73.4_{\pm 3.0}$
Swin-S		CS	81.0	$47.4_{\pm 4.5}$	$59.4_{\pm 3.3}$	$21.7_{\pm 2.6}$	$78.5_{\pm 0.5}$	$29.7_{\pm 3.5}$	$67.1_{\pm 2.7}$
ConvNeXt-S		CS	81.8	$27.8_{\pm 3.1}$	$25.0_{\pm 4.0}$	$25.7_{\pm 1.3}$	$45.0_{\pm 17.3}$	$27.9_{\pm 0.8}$	$70.8_{\pm 1.7}$
		MV	50.8	$20.7_{\pm 3.3}$	$6.8_{\pm 0.1}$	-	-	-	-
SegNeXt-L	HamHead	CS	82.5	$33.5_{\pm 4.1}$	$35.3_{\pm 7.0}$	$29.8_{\pm 6.5}$	$31.5_{\pm 34.4}$	$30.5_{\pm 0.9}$	$65.4_{\pm 1.9}$

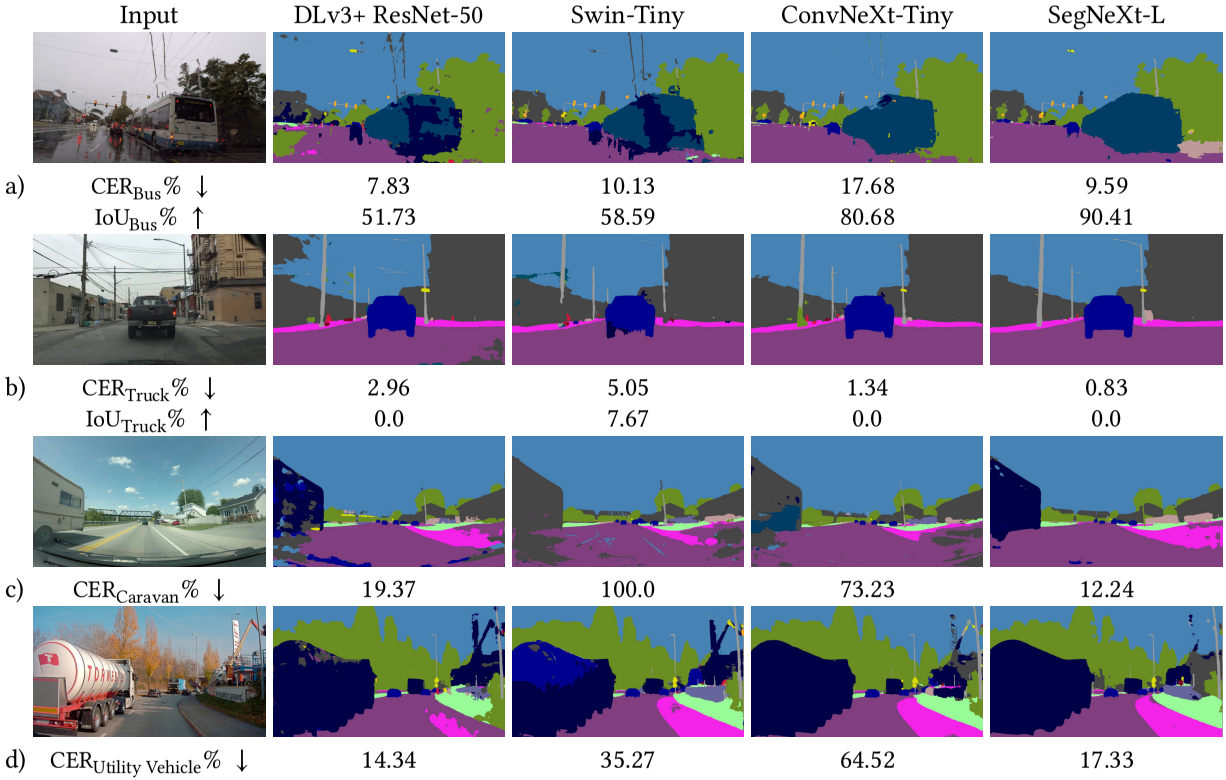


Figure 7.1: Qualitative results on various driving datasets. The semantic segmentation networks are trained on Cityscapes dataset and evaluated in a zero-shot fashion facing domain shift and unknown objects.

7.1.3 Ablation Study

The class hierarchies that are used for computer vision research are generally derived from the visual distinctions between the classes. Two alternative hierarchical concepts are proposed in this section to explore the potential safety implications of utilizing different class taxonomies. Unlike traditional hierarchy definitions, these concepts categorize classes either by their motion behavior, e.g., variations in the velocities of the class *riders* and *pedestrians*, or by whether they possess sufficient impact protection in a collision, e.g., if the class belongs to VRUs. For simplicity, the study focuses exclusively on the leaf classes $\mathcal{V}_\mu$ within *human* and *vehicle* categories.

This ablation study aims to provide the insights of what a flexible and alternative class taxonomy could yield. Therefore, the Mapillary dataset, which offers greater variety of scenes and classes, is selected for the evaluation. Two main aspects are considered in the ablation study:

- **Behavior-based Class Taxonomy**: The proposed class hierarchy is created based on the potential velocity of the classes. One possible use case here is motion prediction tasks. The available leaf classes are divided into two categories: one with considerably low velocities, such as *person* and *wheeled slow*, and the other comprises motorized objects and their riders, which can have higher speeds over a given time interval.
- **VRU-based Class Taxonomy**: This hierarchy bifurcates the leaf classes from those two categories into VRUs, which usually do not have sufficient protection in case of traffic accidents and non-VRUs.

The full class taxonomies used for the evaluation are available in Ch. B from the Appendix. The ablation study concentrates on three major leaf classes: ***M**otorcyclist*, *other **R**ider*, and ***W**heeled slow* under various class taxonomies as detailed in Tab. 7.6. The results indicate a significant variation in the reported CER for the two classes *motorcyclist* and *other rider* after the taxonomy has changed.

Evaluation under the VRU scheme demonstrates, in general, lower out-of-category error predictions. One possible explanation for that is that there

is frequent misclassification among the VRU categories. For instance, it is nontrivial to differentiate between the rider and the vehicle at the objects' boundary, as for instance, *wheeled slow* and *other rider* are considered in the same VRU superclass during evaluation, while they belong to two different categories based on their visual properties.

The other hierarchical classification scheme, as shown in Tab. 7.6, presents a challenge for the semantic segmentation networks as the number of critical errors increases. This is attributed to the reliance on visual differences during training optimization, while the evaluation metric categorizes visually similar classes into semantically distinct categories. For instance, the two visually similar classes, *bicyclist* and *motorcyclist*, are not considered within the same object category due to their motion differences, as it is nontrivial to differentiate them merely based on visual appearance. In order to mitigate this issue, specific training strategies to address such properties should be incorporated during training.

Table 7.6: Ablation study on different class taxonomies with class hierarchy from Mapillary dataset, its variant CER_B based on behavior of the classes and CER_V according to VRU property. CER values are reported over three runs based on classes ***M**otorcyclist*, *other **R**ider*, and ***W**heeled slow*.

	CER% ↓			CER_B% ↓			CER_V% ↓		
Setup	**M**	**R**	**W**	**M**	**R**	**W**	**M**	**R**	**W**
R18	20.9	33.2	66.5	86.2	32.9	48.1	11.4	24.6	34.4
R50	12.4	34.5	61.2	92.8	34.2	56.2	7.7	27.4	39.2
R101	13.4	31.4	68.2	92.3	30.3	49.8	8.9	26.3	39.2
C-T	11.7	23.0	65.2	93.3	21.3	44.6	7.9	19.3	36.2
C-S	11.5	25.7	70.8	92.6	25.3	45.0	7.5	24.3	37.7
C-B	11.6	21.0	72.3	92.9	20.0	42.2	8.5	18.9	33.8

7.1.4 Conclusion

This section proposes a novel evaluation metric for semantic segmentation that explicitly addresses the distinction between intra- and inter-category errors. An initial evaluation of out-of-category errors on source datasets, including Cityscapes and Mapillary, reveals that the optimization process of underrepresented classes is heavily influenced by appearance-similar classes. In a domain-shift setup, the generalization ability of modern architectures,

particularly ConvNeXt, has been re-confirmed. Additionally, the importance of data variety is demonstrated, which affects the performance of neural networks more significantly than the learning capacity of the models. Furthermore, the behavior of models when encountering a novel object is evaluated, illustrating the differences between the models trained on different datasets. In the ablation study, the impact of utilizing class taxonomies is evaluated that are independent of appearance for the category-related semantic segmentation evaluation.

This section posits that a relevant class taxonomy can facilitate the evaluation of domain generalization and the performance of neural networks when encountering novel classes. However, the class hierarchy's effectiveness for criticality evaluation is closely tied to specific use cases. The application of hierarchical taxonomy is considered task-agnostic across various computer vision tasks.

7.2 Few-Shot Semantic Segmentation

After demonstrating that a class hierarchy can facilitate the evaluation of out-of-distribution classes in the task of semantic segmentation, addressing the interpretation problem of the semantic segmentation networks, this section further leverages such class hierarchies to mitigate such corner cases. Due to the nature of the long-tail distribution problem, some classes are underrepresented in the training set. Therefore, one would utilize the limited amount of data addressed in Sec. 5.2.2 with *IP-INO* by annotating the novel classes in the raw datasets with FSSS methods as a preparatory step.

7.2.1 Prerequisites

Current FSSS research mainly emphasizes datasets with object-centric properties, such as Pascal VOC and COCO-Stuff, where the images contain one dominant object class representing the essential visual properties. However, scene understanding for the assisted and automated driving functions poses

greater challenges compared with the traditional datasets, as the properties of the data may shift due to its specific use case. Figure 7.2 shows a histogram of the proportion distribution of foreground classes in the images from the frequently used object-centric and driving scene datasets.

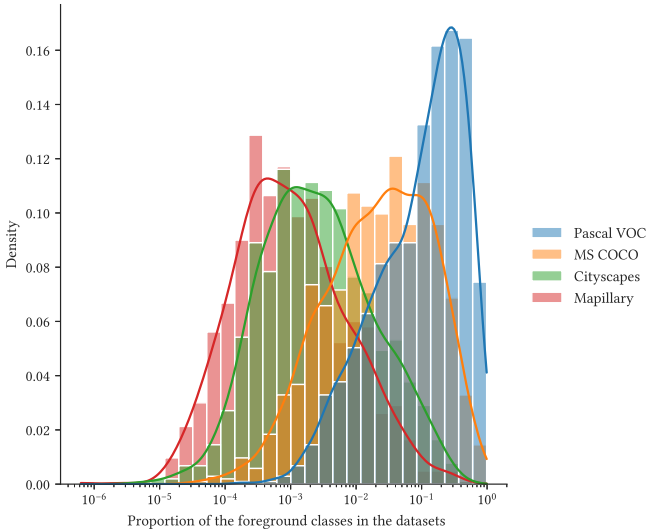


Figure 7.2: The distribution of foreground class proportions across four commonly used datasets for Few-Shot Semantic Segmentation (FSSS).

For those datasets that emphasize driving tasks, due to the lack of *background* class and according to the real use cases, dynamic classes from the categories *vehicle* and *human* are chosen as foreground classes. The distribution of the foreground classes reveals that, compared to the street scene datasets, the foreground objects in datasets that are traditionally employed in FSSS are more prominently featured. Specifically, for Pascal VOC, the foreground objects often occupy more than 10% of the image area. In contrast, COCO-Stuff, where the scenes are becoming more challenging, has a foreground class ratio generally between 1% and 10%. However, for datasets that contain driving scenes, such as Cityscapes and Mapillary, the foreground classes tend to be significantly smaller, representing only around 0.1% and 0.01% of the total

area, respectively. Such discrepancy can be anticipated to result in notable variations in the learning efficiency of FSSS models, partially attributable to the insufficiently representative features generated for novel classes from the support set.

However, in traditional FSSS research, novel classes within datasets are normally selected without considering the similarity between the novel classes and the previously known classes. For instance, in the Pascal VOC-5^i dataset, 20 classes are evenly split into four folds with $i \in \{0, 1, 2, 3\}$, each containing five classes based on their class index. As introduced in Sec. 3.3, previous research frequently used the learning paradigm of utilizing all images and all foreground classes from the training set. During training, the previously known classes that are present in the image are utilized as base classes $\mathcal{C}_{\text{base}}$. One of the base classes, referred to as the *targeted class*, is selected to be re-annotated as a binary segmentation map with the *foreground* as positive while the rest of the pixels are annotated as the *background*. This aims to train the meta learner so that it can use the learned features and generalize to further novel classes. During evaluation, $\mathcal{C}_{\text{novel}}$ instead of $\mathcal{C}_{\text{base}}$ is selected to perform few-shot segmentation.

This section specifically concentrates on novel classes from the *vehicle* category as a continuation of the previous Sec. 7.1. As shown in the previous experiments, this setup is posited to be realistic for practical perception applications for assisted and automated driving functions due to the high variation in the appearance within the *vehicle* category, which also has inherently higher deviation compared with other dynamic object classes from the different categories, e.g., *human*.

In summary, this section will focus on the major aspects highlighted by the blue dashed boxes in Fig. 7.3, aiming to obtain accurate feature representations of novel classes despite variations among object instances in driving scenes. The proposed and examined methodology is category- and architecture-agnostic, which can also be easily applied to other object categories beyond *vehicle* and not bounded to one certain network structure.

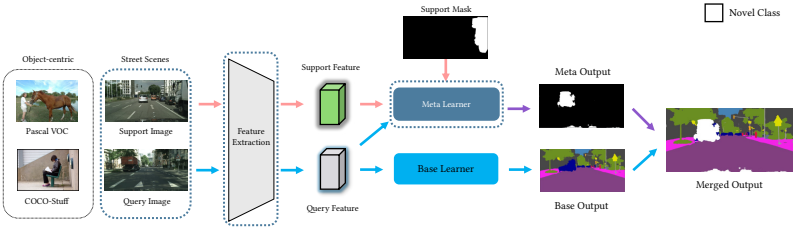


Figure 7.3: Overview of a FSSS architecture [Lan22]. This section focuses on the application of FSSS in complex driving scenarios, utilizing datasets containing street scenes illustrated in the left dashed box. Additionally, experiments are conducted to evaluate the performance of the optimized feature extractors and the training organization of the meta learner.

7.2.2 Proposed Methodology

Knowledge Integration

According to the prerequisites, it is assumed that the targeted novel classes share common visual properties with some known base classes from the same object category. In addition, the novel class set $\mathcal{C}_{\text{novel}}$ is considered to belong to a single category k^c, specifically in this work with the category *vehicle*, noted as $\mathcal{C}_{\text{novel}}^{k^c}$. During training, rather than employing a model that selects any class from the entire base set $\mathcal{C}_{\text{base}}^{\text{all}}$ as the *targeted class* to train the meta learner, the method leverages only those target classes that belong to the same category k^c, noted as $\mathcal{C}_{\text{base}}^{k^c}$. This pre-selection of the base classes encourages the model to focus on the most relevant category-specific features, avoiding interference from unrelated classes and categories. Especially for some classical *background* classes, e.g., *sky*, features that focus on other visual structures are required. Consequently, this enhances the performance in segmenting unseen classes within category k^c.

The concept can be formulated as follows:

$$\mathcal{K} = \{k^i\}_{i=1}^{N_\mathrm{K}} \tag{7.3}$$

$$\mathcal{C}_{\mathrm{base}}^{all} = \{c_{\mathrm{base}}^{i}\}_{i=1}^{N_{\mathrm{base}}} \tag{7.4}$$

$$\mathcal{C}_{\mathrm{base}}^{k^c} \subset \chi(k^c) \tag{7.5}$$

$$\mathcal{C}_{\mathrm{novel}}^{k^c} \subset \chi(k^c) \tag{7.6}$$

$$\mathcal{C}_{\mathrm{base}}^{k^c} \cap \mathcal{C}_{\mathrm{novel}}^{k^c} = \emptyset \tag{7.7}$$

where χ returns a set of node v's child nodes, $\mathcal{K}$ represents the complete set of N_K categories, and k^c denotes a constant single category.

Effective Feature Extraction

The significance of representative feature extraction from support set images for the performance of FSSS has been extensively discussed in existing research [Hu23c, Zho23a, Zha22c]. As introduced in Sec. 3.3, the meta learner utilizes features of novel classes from the image encoder by comparing the similarity between the support and query images' extracted features. Therefore, effective and representative feature representations are bound to increase the performance of the FSSS model.

In addition to the ResNet-50 model that is frequently used as a baseline in current FSSS research, this study also extends the feature encoder with the ConvNeXt encoder. For fairness, ConvNeXt-Tiny is selected due to its network size, which closely matches that of the baseline feature encoder ResNet-50. This selection facilitates an assessment of ConvNeXt's feature extraction capabilities, and eliminates model scale differences as a confounding factor. Furthermore, calculating the similarity between the features constitutes the primary factor in computational resource demands. As the final feature maps from ConvNeXt-Tiny exhibit reduced feature dimensions and fewer channels

compared to those extracted by ResNet-50, ConvNeXt-Tiny holds a significant advantage as the image encoder for FSSS models, as the similarity computation requires less time. Such effects are especially evident with high-resolution images from the Mapillary dataset. However, this may adversely affect the accurate representation of smaller objects.

Support Image Selection

FSSS is dependent on information about novel classes derived from a limited number of support images to segment novel classes accurately. Compared with the conventional FSSS dealing with object-centric datasets, as shown in Fig. 7.2, there exists a substantial discrepancy in the ratio of the occupied area by foreground objects between object-centric datasets and those primarily comprising driving scenes. Consequently, the performance of FSSS models is significantly affected by the quality of the support set in addition to the image encoders. From the size distribution, it can be observed that a majority of foreground classes in street scenes exhibit size ratios ranging from 10^{-4} and 10^{-2}. In such circumstances, the feature representations that are generated from the support image set might be limited and insufficient to accurately represent the novel class due to the pose and texture of the object. To address this, all support images for each novel class are sorted by the number of pixels a certain novel class contains. As a result, multiple subsets are constructed for each novel class, each comprising the top p_i percent of the sorted support image set, where $p = \{10 \cdot i\}_{i=1}^{10}$. Based on the predefined subsets, the inference is conducted using a subset that contains novel class instances with various sizes as a support set to evaluate the impact of support image quality on the segmentation performance of FSSS models.

7.2.3 Experimental Setup

For the experiments, two mainstream datasets are used for evaluation, the Cityscapes dataset and Mapillary dataset, which are introduced in Sec. 2.4. Concentrating on the category *vehicle*, the main emphasis in the Cityscapes

dataset is the three novel classes, *train*, *motorcycle* and *truck*, and in the Mapillary dataset, the classes *caravan*, *trailer*, *train*, *boat*, and *wheeled slow* are used according to the annotation policy (v1.2). These classes are selected based on their frequency of occurrence in the datasets, as this section aims to simulate the semantic segmentation of underrepresented classes by leveraging a limited amount of data.

For evaluation, the metric IoU is utilized based on shuffled support image sets for each novel class. The mean IoU is reported by calculating the average of the IoU scores of all the previously unknown classes.

Implementation

In previous work, classical FSSS research focuses on segmenting novel classes in the form of binary segmentation, often neglecting the model's capability to segment previously known classes. Moreover, such capability is essential for unknown discoveries in the context of data mining for assisted and automated driving functions, so that known classes can also be segmented. Lang et al. [Lan22] addressed this issue by introducing a new branch specifically for base class segmentation. Therefore, their network architecture is adopted as the foundation for the evaluation in this section. The training process is divided into two distinct phases: Training for the base learner and the meta learner. The proposed scheme is therefore called Base and the Meta learners (BAM). During the base learner phase, the model is trained exclusively for the base classes. In this section, two types of base learner network architectures are employed: PSPNet with a ResNet-50 backbone representing classical CNN in the field of semantic segmentation and UPerNet with a ConvNeXt-Tiny backbone. It is important to note that the methods proposed in Sec. 7.2.2 are model-agnostic, permitting implementation in any FSSS model without being limited to BAM. Deviating from the original implementation of BAM, the images from both training sets are entirely included, with novel classes annotated as *ignore*, so that the gradients from the novel classes are not included in the backpropagation to prevent data leakage. Contrarily, BAM originally omitted images containing novel classes from the training dataset. Regarding the

meta learner phase, the same meta learner model from BAM, PFENet [Tia20], is utilized. During the training of the meta learner, novel classes in the training images are also annotated as *ignore* instead of *background*, which is frequently adopted in traditional FSSS research. This approach is able to reflect performance more accurately, as novel classes are not considered during the training phase of the meta learner. For consistency, all subsequent experiments in this section are conducted using this training methodology. For evaluation, the weights yielding the best mean Intersection over Union score on the novel classes from the validation set are selected as the best weights.

7.2.4 Results and Discussions

Support Image Selection

As shown in Fig. 7.2, the size of the class instances in the image varies across datasets. The quality of meta learner outputs heavily relies on the generated features from the support image set. The prediction for the same query image can differ if the support features are not adequately representative. Therefore, this section begins with an investigation of the potential impact of leveraging different support image sets. Fig. 7.4 illustrates how the mean Intersection over Union (mIoU) varies with the size of the available support image set during inference, where the images are sorted in descending order based on the pixel occupancy of the novel objects. The experiments are conducted with the ResNet-based model architecture on the Cityscapes dataset, where the three classes, *train*, *motorcycle*, and *truck* are considered as novel classes in the *vehicle* category.

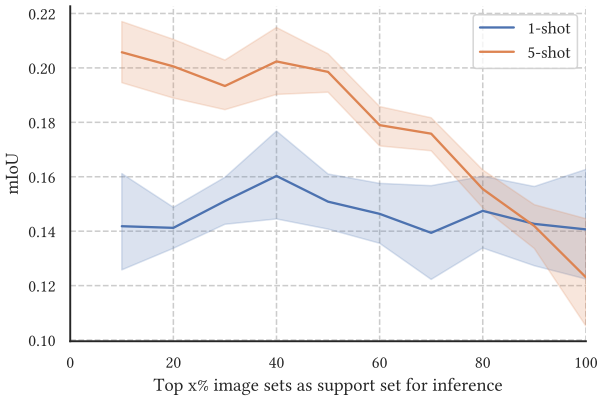


Figure 7.4: Validation mean IoU of the baseline FSSS models using various subsets of the top x% of support images, sorted by object instance size.

In a 1-shot setting, the models show increasing performance as more diverse images are included in the support image set, up to 40% of the entire set. The underlying reason is that sorting support images by the size of the novel objects is suboptimal for complex scenes, especially when the amount of data is limited. Given the high likelihood of outliers and varying poses of objects, the extracted features may not be representative for the novel classes. For instance, a rear view of a truck towing a trailer provides limited assistance for detecting another truck from other perspectives, even if the object occupies a significant portion of the support image. After the 40% threshold, the segmentation performance on novel classes declines and plateaus, indicating that a mixture of small object instances in the support image set does not positively contribute to the few-shot segmentation performance. Additionally, it can be observed that the standard deviation of performance increases as more support images with small novel objects are included.

In the 5-shot setting, the performance of novel class segmentation depicts a decreasing trend when more support images containing only a small area of the novel class are utilized. The best performance in segmenting the novel class is observed when only the top 10% of images are chosen as support

sets. By comparing the best-performing models, the mean IoU shows a relative increase of 30%, from 16% to about 21%. However, utilizing the entire dataset generates insufficient features that negatively impact model performance, sometimes resulting in 5-shot setups being outperformed by their 1-shot counterparts, as the smaller object instances are not able to summarize the fundamental visual properties representatively. Benefiting from more robust support features, the 5-shot FSSS setup demonstrates lower standard deviations compared to the 1-shot variants, further emphasizing the importance of generated support features for novel class segmentation.

To ensure reliable and consistent results in subsequent experiments, an optimum ratio of support images that yield higher performance and lower standard deviation in both 1-shot and 5-shot settings is selected. As Fig. 7.4 shows, employing the top 20% setup achieves a balance between performance and stability in the 5-shot setting, while the 1-shot setting exhibits the lowest overall standard deviation. For the following experiments, the top 20% of images containing the novel classes sorted by size are selected as the support set during evaluation on the two driving scene understanding datasets.

Cityscapes

Table 7.7: Evaluation results of FSSS models on Cityscapes and Mapillary datasets with various novel class setups. The average IoUs are reported based on 10 different runs utilizing shuffled support image sets.

$\mathcal{C}^n$	Optimizations Hier.	Optimizations Backbone	1-shot $\text{IoU}_{\text{novel}}$				5-shot $\text{IoU}_{\text{novel}}$			
Cityscapes										
			Train	Motorcycle	Truck	mIoU	Train	Motorcycle	Truck	mIoU
{Train}			$25.0_{\pm 3.1}$				$31.3_{\pm 1.8}$			
	✓		$33.4_{\pm 5.6}$				$39.6_{\pm 1.9}$			
		✓	$27.5_{\pm 3.6}$				$52.1_{\pm 2.3}$			
	✓	✓	$56.6_{\pm 0.3}$				$57.1_{\pm 0.1}$			
{Motorcycle}				$11.1_{\pm 1.2}$				$16.0_{\pm 1.6}$		
	✓			$12.6_{\pm 1.8}$				$15.0_{\pm 1.3}$		
		✓		$13.7_{\pm 2.2}$				$15.6_{\pm 0.9}$		
	✓	✓		$13.5_{\pm 2.4}$				$15.8_{\pm 0.4}$		
{Train, Truck, Motorcycle}			$24.9_{\pm 7.0}$	$11.5_{\pm 2.1}$	$7.0_{\pm 1.7}$	$14.5_{\pm 2.9}$	$36.5_{\pm 5.5}$	$13.1_{\pm 1.0}$	$8.7_{\pm 1.5}$	$19.5_{\pm 1.9}$
	✓		$38.7_{\pm 6.0}$	$12.4_{\pm 2.3}$	$15.4_{\pm 2.7}$	$22.2_{\pm 2.5}$	$47.5_{\pm 4.3}$	$20.1_{\pm 0.6}$	$16.8_{\pm 2.2}$	$28.1_{\pm 1.8}$
		✓	$18.1_{\pm 4.7}$	$16.5_{\pm 2.0}$	$20.7_{\pm 2.1}$	$18.4_{\pm 2.0}$	$25.3_{\pm 3.7}$	$21.0_{\pm 0.5}$	$23.2_{\pm 1.0}$	$23.2_{\pm 1.5}$
	✓	✓	$56.9_{\pm 4.5}$	$11.8_{\pm 1.5}$	$22.8_{\pm 2.8}$	$30.5_{\pm 1.9}$	$59.2_{\pm 5.2}$	$13.9_{\pm 0.5}$	$24.3_{\pm 1.7}$	$32.5_{\pm 1.9}$
Mapillary										
			Caravan	Trailer	On Rail	mIoU	Caravan	Trailer	On Rail	mIoU
{On Rail}					$26.7_{\pm 4.1}$				$36.0_{\pm 2.8}$	
	✓				$45.7_{\pm 5.4}$				$54.2_{\pm 2.9}$	
		✓			$23.2_{\pm 5.5}$				$37.1_{\pm 3.7}$	
	✓	✓			$57.4_{\pm 0.4}$				$57.8_{\pm 0.2}$	
{Caravan, Trailer, On Rail}			$20.9_{\pm 11.1}$	$17.2_{\pm 3.6}$	$30.8_{\pm 6.7}$	$23.0_{\pm 3.0}$	$24.3_{\pm 7.0}$	$18.9_{\pm 3.0}$	$47.7_{\pm 3.9}$	$30.3_{\pm 2.2}$
	✓		$51.1_{\pm 13.1}$	$27.4_{\pm 2.2}$	$46.5_{\pm 4.6}$	$41.7_{\pm 3.4}$	$68.4_{\pm 5.5}$	$31.4_{\pm 1.8}$	$53.4_{\pm 1.8}$	$51.1_{\pm 1.5}$
		✓	$34.8_{\pm 11.7}$	$9.4_{\pm 2.6}$	$16.0_{\pm 2.5}$	$20.0_{\pm 3.6}$	$52.1_{\pm 1.8}$	$11.4_{\pm 2.6}$	$26.3_{\pm 2.2}$	$29.9_{\pm 1.3}$
	✓	✓	$58.8_{\pm 12.6}$	$30.7_{\pm 2.1}$	$55.8_{\pm 1.1}$	$48.4_{\pm 4.2}$	$63.8_{\pm 7.0}$	$30.7_{\pm 0.7}$	$56.4_{\pm 0.5}$	$50.3_{\pm 2.2}$
{Caravan, Trailer, On Rail, Boat, Wheeled Slow}			$27.0_{\pm 10.6}$	$18.1_{\pm 3.3}$	$27.2_{\pm 5.7}$	$17.1_{\pm 2.6}$	$27.0_{\pm 5.8}$	$24.8_{\pm 1.1}$	$45.0_{\pm 1.4}$	$24.6_{\pm 1.6}$
	✓		$56.7_{\pm 15.2}$	$26.0_{\pm 3.0}$	$50.5_{\pm 2.0}$	$33.2_{\pm 3.3}$	$72.7_{\pm 2.9}$	$31.9_{\pm 0.7}$	$53.3_{\pm 0.7}$	$40.4_{\pm 0.8}$
		✓	$56.2_{\pm 10.3}$	$8.5_{\pm 3.9}$	$17.4_{\pm 6.5}$	$17.9_{\pm 2.7}$	$61.8_{\pm 2.8}$	$7.4_{\pm 2.5}$	$27.9_{\pm 4.2}$	$20.3_{\pm 1.2}$
	✓	✓	$50.3_{\pm 18.4}$	$28.3_{\pm 2.3}$	$55.0_{\pm 0.5}$	$32.8_{\pm 4.3}$	$70.4_{\pm 2.5}$	$29.1_{\pm 0.6}$	$55.1_{\pm 0.3}$	$38.7_{\pm 0.5}$

As discussed in Sec. 7.2.2, the evaluation commences with two commonly-used driving datasets. According to the previous experiments about support image selection, unless otherwise specified, the baseline model always refers to the model that employs the top 20% sorted support image set during inference. The base class setup is utilized according to Eqs. (7.3) to (7.7) for experiments involving knowledge integration of class hierarchy. For comparison, a predefined set of random seeds is used to run inference, thereby assessing their performance across different support image sets. The evaluation results from the Cityscapes and the Mapillary dataset are depicted in Tab. 7.7.

With the novel class *train* from Cityscapes, it is observed that using a more effective feature extractor or considering class hierarchy during the training

of meta learner alone results in only modest improvements in the segmentation performance. However, by combining those two methods or in a 5-shot setting, a significant performance enhancement is achieved, as generating effective and representative feature representations notably elevates the segmentation performance of the model from 25.0% to 57.1% IoU. Furthermore, the 5-shot variants exhibit lower standard deviations when the generated features are stabilized by multiple support images that contain the novel class instances during inference. Finally, it can be observed that the performance gap between 1-shot and 5-shot segmentation narrows after introducing the proposed optimization methods, indicating that the extracted feature representations are more robust and reliable despite the limited data.

However, the class *motorcycle* shows limited improvement in segmentation performance with the applied methodologies, as the IoU ranges from 11.1% to 13.7% in the 1-shot setting and remains around 15% in the 5-shot setting. This limited improvement is attributed to the higher aleatoric uncertainty of the *motorcycle* class compared to the class *train*, as the support images and query images for *motorcycle* may contain varying visual appearances, poses, and occlusions from the driving environment and the riders, thereby restricting the potential for improvement.

In the evaluation incorporating three novel classes, individual class performance improves similarly after applying the proposed methods compared to the setup where the meta learner is supposed only to segment the *train* and *motorcycle* classes. Compared with the base setup for one novel class, this setup, where the model aims to segment three novel classes, demonstrates that the proposed methods significantly boost the segmentation performance, doubling the mIoU in the 1-shot setting from 14.5% to 30.5% with lower standard deviation. Additionally, learning multiple novel classes does not adversely affect performance, as evidenced by comparing the *train* and *motorcycle* results across these two setups. Furthermore, the experiments reveal that the absence of certain classes during the meta-training phase does not impair the model's generalization capability. In some cases, utilizing fewer classes

for the meta-training improves the segmentation performance, which will be further evaluated in Sec. 7.2.5.

Mapillary

After the evaluation on the Cityscapes dataset, the proposed methods are evaluated on the Mapillary dataset, which is regarded as covering more complex, worldwide driving scenarios and exhibits a long-tail data distribution in the annotation policy. As evidenced by the inclusion of the *caravan* class from Tab. 7.7, the standard deviation for this class is significantly higher than for other classes, attributed to the presence of only ten images that contain this class in the whole validation set. Similar to the results of the class *train* from the Cityscapes dataset, the class *on rail* from Mapillary dataset significantly enhances the segmentation performance in both 1-shot and 5-shot settings. This improvement indicates that the utilization of class hierarchy during meta-training is more effective on Mapillary than on the Cityscapes dataset despite the more challenging scenarios. By combining the proposed optimization methods, the model attains promising results for this class, thereby demonstrating the efficacy of the proposed methodologies in more complex domains. Similarly to the Cityscapes dataset, the 1-shot model with the proposed methodologies reduces the performance disparity with the 5-shot model, indicating that the feature extraction of the novel classes is more effective.

The generalization ability of the meta learner is further evaluated in scenarios involving three and five novel classes as introduced in Sec. 7.2.3. Observed by the class *trailer*, introducing transformer-inspired feature extractor negatively influences the segmentation performance, whereas integrating a class hierarchy and combining the two methods almost double the segmentation performance measured by IoU in both 1-shot and 5-shot settings. Additionally, a comparative analysis of the *trailer* and *on rail* classes across different novel class settings suggests that the diversity of available base classes during meta learner training positively impacts model generalization, thereby enhancing overall performance. Together with the findings from Cityscapes, Section 7.2.5 will delve deeper into this correlation, examining the impact of

base class variety during meta learner training, the use of class hierarchy, and the generalization capacity of the model.

Generally, the combination of effective feature extraction and prior knowledge augmentation yields performance improvements, except for certain experiments where the network architectures with ResNet-50 backbone outperform the others. One explanation could be that the final feature maps generated by ConvNeXt-Tiny have a lower resolution, as discussed in Sec. 7.2.2, complicating the detection of object instances that necessitate fine-grained feature representations due to their size or texture. In addition, the observed decrease in mean IoU from the three-novel-classes setup to the five-novel-classes setup indicates the limited performance of those specific classes, such as *wheeled slow* and *boat*, with their IoUs lower than 10% IoU. This limitation is indicative of the inherent complexity of these novel classes. It suggests that one necessary prerequisite of the proposed hierarchical knowledge integration is that the novel classes share common visual properties with base classes.

7.2.5 Ablation Study

Extending from Sec. 7.2.4, the ablation study further explores the impact of deploying various base class setups during the meta-training phase on the performance of novel class discovery. Table 7.8 presents a comparison of the model's performance based on various base class configurations in a 5-shot setting, as this setup enriches the feature representations from the novel classes from more novel object instances. The experiments follow the setup that contains three novel classes from the Mapillary dataset. To ensure fairness and comparability, an iteration-based runner is applied to mitigate the potential impact of dataset size discrepancies, as the number of training images is related to the available base classes. As the baseline for comparison, all classes according to the labeling policy (v1.2) from the Mapillary dataset are utilized for training.

Consistent with observations in Sec. 7.2.4, incorporating class hierarchy during meta learner training enhances performance, observed by the mIoU improvements from utilizing all base classes (29.9%) to only leveraging the *vehicle* classes (50.3%). In contrast, when classes from another visually dissimilar category, e.g., *human*, are exclusively used during the training process, the meta learner fails to transfer knowledge to novel classes in the *vehicle* category, which indicates the importance of prior knowledge for the FSSS performance.

However, if the classes used for meta-training are mixed with other foreground classes in addition to the *vehicle* classes, including *other rider* or the full integration of the *human* category, an improvement in performance on novel classes is observed. This suggests that the meta learner can enhance its generalization ability using class instances from certain visual invariant object categories. Therefore, balancing category-specific features with generalization is crucial for optimal performance of FSSS models. In contrast, integrating all possible base classes during the meta-training phase negatively affects performance.

Table 7.8: Comparison of various configurations for the base class selection during meta learner training. Models utilize features extracted from the ConvNeXt backbone in 5-shot setting.

Meta Learner Base Class	Caravan	Trailer	On-Rail	$\text{mIoU}_{\text{novel}}$
All Base Classes	$52.1_{\pm 1.8}$	$11.4_{\pm 2.6}$	$26.3_{\pm 2.2}$	$29.9_{\pm 1.3}$
All Human Classes	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.3_{\pm 0.0}$	$0.1_{\pm 0.0}$
All Vehicle Classes	$63.8_{\pm 7.0}$	$30.7_{\pm 0.7}$	$56.4_{\pm 0.5}$	$50.3_{\pm 2.2}$
All Vehicle Classes with *other rider*	$69.4_{\pm 2.1}$	$33.0_{\pm 0.3}$	$53.5_{\pm 0.1}$	$52.0_{\pm 0.7}$
All Vehicle Classes with *human*	$68.5_{\pm 2.0}$	$33.1_{\pm 0.4}$	$56.0_{\pm 0.2}$	$52.5_{\pm 0.7}$

7.2.6 Realistic Evaluation

The current evaluation scheme for FSSS operates under the strong assumption that all evaluation images contain novel classes [Tia20, Iqb22, Min21], an assumption that is deemed unrealistic for real-world applications. One potential solution to address such requirements involves first determining the presence and type of novel classes in the query images prior to FSSS [Kan22].

However, such an approach is not considered in this section as it lies beyond the scope of this work. In order to evaluate the effectiveness of the proposed methodologies under more realistic conditions, the performance of the FSSS model, tasked with segmenting three novel classes from the Cityscapes dataset, is assessed across the standard validation set that contains 500 images and compared to results from the traditional evaluation scheme.

As illustrated in Tab. 7.9, significant performance degradation is observed when transitioning to the revised evaluation paradigm in which images do not necessarily contain novel classes. The decreases in IoU scores indicate that the meta learning model generates a substantial number of false positive predictions, resulting in a relative mean IoU decrease of approximately 75%. A similar trend is evident in the 5-shot setting, where the baseline model exhibits notably higher performance loss with the new evaluation scheme. The proposed methodology, however, can partially mitigate this performance loss in both 1-shot and 5-shot settings. Nevertheless, such an issue, associated with generalized few-shot semantic segmentation, needs to be further explored in future work, as false positive predictions are also prevalent with conventional FSSS datasets, such as Pascal VOC-5^i and COCO-20^i. Additionally, selecting the optimal weights during training can be particularly challenging, especially when access to the validation images is restricted. Ensuring that the validation set is representative of potential use cases covered during inference is essential, as any misalignment between the validation and test sets can significantly impact the model's generalization. Such misalignment is particularly problematic given that driving scenes are inherently more complex than conventional setups.

Table 7.9: Comparison of the IoU results on the novel image set and whole validation set of Cityscapes. Rows in grey are evaluated for the whole validation dataset.

	Train	Motorcycle	Truck	$\text{mIoU}_{\text{novel}}$
		1-shot		
Baseline	$24.9_{\pm 7.0}$	$11.5_{\pm 2.1}$	$7.0_{\pm 1.7}$	$14.5_{\pm 2.9}$
	$5.2_{\pm 1.0}$	$5.4_{\pm 0.7}$	$4.2_{\pm 1.5}$	$4.9_{\pm 0.5}$
Proposed	$56.9_{\pm 4.5}$	$11.8_{\pm 1.5}$	$22.8_{\pm 2.8}$	$30.5_{\pm 1.9}$
	$25.2_{\pm 4.8}$	$5.6_{\pm 0.7}$	$11.4_{\pm 1.6}$	$14.1_{\pm 1.5}$
		5-shot		
Baseline	$36.5_{\pm 5.5}$	$13.1_{\pm 1.0}$	$8.7_{\pm 1.5}$	$19.5_{\pm 1.9}$
	$7.5_{\pm 1.0}$	$6.4_{\pm 1.0}$	$5.3_{\pm 1.0}$	$6.4_{\pm 0.6}$
Proposed	$59.2_{\pm 5.2}$	$13.9_{\pm 0.5}$	$24.3_{\pm 1.7}$	$32.5_{\pm 1.9}$
	$26.8_{\pm 3.6}$	$7.3_{\pm 0.9}$	$12.4_{\pm 1.1}$	$15.5_{\pm 1.1}$

7.2.7 Conclusion

This section concentrates on the evaluation of utilizing current few-shot semantic segmentation (FSSS) methods on complex driving scenes with a priori knowledge. The unique challenges posed by such complex driving scenes were first analyzed by comparing traditional object-centric datasets with driving scene datasets, focusing on the difference between foreground and background class distribution, showing that the low presence of foreground dynamic objects has a negative impact on FSSS. In this section, three methods were proposed to mitigate the limitations of complex driving scenes, including integration of class taxonomies, effective feature extraction, and support image selection during inference. These methodologies were utilized to enhance the performance and robustness of discovering and segmenting novel objects using limited information from the support image set. The proposed methods were validated using two frequently used street-scene semantic segmentation datasets with various novel class configurations, revealing conditions and setups under which these methodologies outperform conventional training setups. Lastly, the study contemplates the practical application of FSSS for mitigating of corner cases in the context of assisted and automated driving functions.

8 Meta Information as Contextual Knowledge

Chapters 6 and 7 explored the use of class hierarchies as contextual knowledge for unknown-aware object detection, domain generalization evaluation for semantic segmentation models, as well as few-shot semantic segmentation. This chapter shifts the focus to other contextual knowledge that can be obtained from external data sources, for instance, publicly available information or existing vehicle fleets equipped with quality-assured sensor systems. The following research question is investigated:

Research Question 4: Beyond class taxonomies, how can other contextual knowledge be leveraged to mitigate corner cases?

- Hypothesis 4.1: Meta information from the target domain can provide efficient support during the domain adaptation process of a data-driven model.

With the help of such publicly available meta information, the functions can be developed and validated without utilizing cost-intensive manual annotations for the data. Thereby, this also addresses cases where a corner case occurs in the *development dataset* without annotations (*IP-INO*), as discussed in Ch. 5. With an automatic data pipeline, the unlabeled data can be enriched and further leveraged to mitigate corner cases.

Specifically, this chapter addresses lane detection leveraging prior knowledge containing road structure information, such as Number-of-Lanes (NoL), as an

illustrative example, showcasing how such external knowledge can be integrated into data-driven models. The results are based on the author's publication [Zho24b]. This work was recognized with the distinction of the Best Paper Award of the Safe Artificial Intelligence for All Domains Workshop (SAIAD) at the 2024 IEEE / CVF Computer Vision and Pattern Recognition Conference.

8.1 Preliminary Study

The preliminary study aims to evaluate and compare the frequently used lane detection models from Sec. 2.3 in order to assess their generalization ability as a baseline. This experiment utilizes two datasets: TuSimple, which represents highway driving scenarios in the USA characterized by relatively simple driving conditions, and CULane, which encompasses more complex urban driving scenarios featuring lane occlusions. During the experiment, the models are first trained on the source dataset $\mathcal{D}_S$, and their performance on the target dataset $\mathcal{D}_T$ without any model adaptation or configuration changes indicates their generalization ability. For comparison, the models are trained exclusively on the target dataset $\mathcal{D}_T$ with identical initialization to indicate the performance upper bound. As shown in Tab. 8.1, significant performance degradation is observed across all tasks and models compared with their upper bounds. Particularly severely affected by the domain shift are the models that are pretrained on TuSimple, where a disparity of about 50% F1-score is observed with the best-performing model and about 65% with the worst generalized model. Even though the task CULane $\rightarrow$ TuSimple is considered to be less complex than the reverse task as the simple lane structure on the highway facilitates the lane detection, the models struggle with the new task as the difference in F1-scores ranges between 20% and 40%.

Table 8.1: Domain generalization evaluation of five lane detection models on two datasets. Performance of the models trained directly on the target domain are shown in **(bold)**

Model	$\mathcal{D}_s$CULane → $\mathcal{D}_T$TuSimple				$\mathcal{D}_s$TuSimple → $\mathcal{D}_T$CULane		
	F1 ↑	FPR ↓	FNR ↓	Accuracy ↑	F1 ↑	Precision ↑	Recall ↑
CondLaneNet [Liu21a]	70.2 **(97.0)**	17.4 **(2.2)**	52.6 **(3.8)**	58.7 **(95.5)**	12.7 **(77.7)**	22.8 **(83.1)**	8.8 **(73.1)**
CLRNet [Zhe22]	70.3 **(95.3)**	32.6 **(5.5)**	26.3 **(3.8)**	85.9 **(95.1)**	23.0 **(78.8)**	50.9 **(84.5)**	14.8 **(73.8)**
LaneATT [Tab21]	54.9 **(95.1)**	49.6 **(5.9)**	39.4 **(3.7)**	81.3 **(94.9)**	23.4 **(76.0)**	67.3 **(82.7)**	14.1 **(70.4)**
GANet [Wan22b]	74.0 **(97.8)**	24.0 **(1.6)**	29.4 **(2.9)**	82.0 **(95.8)**	21.9 **(78.5)**	54.3 **(85.4)**	13.7 **(72.6)**
CLRerNet [Hon24]	69.9 **(97.6)**	24.7 **(1.7)**	40.2 **(3.1)**	73.2 **(96.5)**	31.1 **(81.1)**	38.0 **(88.1)**	26.4 **(75.2)**

It is shown that even the state-of-the-art lane detection models significantly suffer from visual appearance shifts of the lanes and from occlusions caused by other traffic participants. Furthermore, in practice, visual appearance shift does not encapsulate the entirety of domain shift, as there are other cases where lanes and road boundaries should only be detected in certain circumstances, e.g., dealing with hard shoulders on highways. This further motivates the integration of available meta information during training for the domain adaptation of lane detection models.

8.2 Methodology

This section introduces the methodology of leveraging meta information for domain adaptation. As discussed in Sec. 2.3, ϕ_θ^{S} is learned from the annotated source domain data $\mathcal{D}_{\mathrm{S}} = \{(x_s, y_s)\}$. The preliminary study suggests that the mapping function ϕ_θ that is only trained on the source domain is insufficient to deal with constantly changing driving situations. In the previous work, the unsupervised domain adaptation paradigm is investigated as the target dataset $\mathcal{D}_{\mathrm{T}}^{\mathrm{UDA}} = \{(x_t)\}$ is available without any annotation. For the proposed *Weakly Supervised Domain Adaptation framework for Lane Detection* (WSDAL), the number of lanes n_t in each target image is known during adaptation with $\mathcal{D}_{\mathrm{T}}^{\mathrm{WSDAL}} = \{(x_t, n_t)\}$.

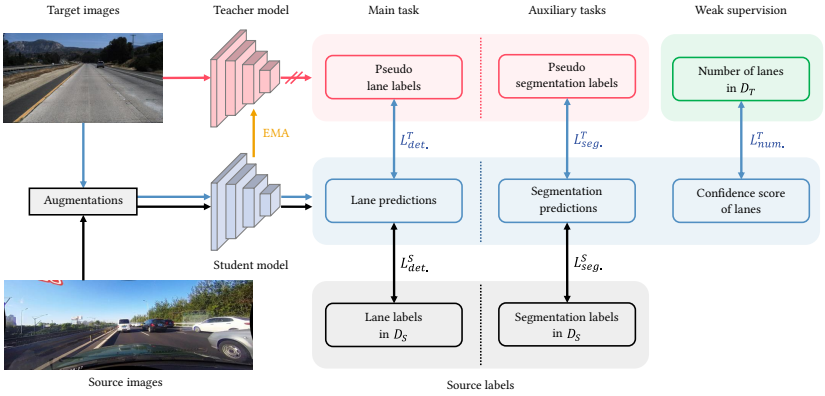


Figure 8.1: The proposed Weakly Supervised Domain Adaptation framework for Lane Detection (WSDAL), which consists of a teacher–student network, a segmentation head as an auxiliary task during training and weak supervision head utilizing Number-of-Lanes labels during domain adaptation.

Architecture Overview

Figure 8.1 depicts the proposed framework, which is heavily inspired by the current advances in the unsupervised domain adaptation research, where a self-distilling teacher–student training strategy is employed. The teacher model shares model architecture and initialization with the student model at the beginning of the adaptation. Similar to the unsupervised setup, the student model makes predictions on heavily augmented images from the target domain during training, while the teacher model infers directly on the raw images, creating pseudo lane labels from the target domain. As demonstrated by Tranheden et al. [Tra21], Zhang et al. [Zha21a], and Hoyer et al. [Hoy22], such a strategy is effective in transforming low-quality predictions into pseudo labels and thus enabling the supervision and adaptation of the student model. At the same time, the robustness of the student model is enhanced as it is fed with data from the target domain with strong data augmentation. In the implementation of this work, a confidence threshold is applied to select the pseudo lane labels that exceed a predefined confidence for the corresponding lane prediction following $l_n^{(L-1)} > \lambda_{\text{LD}}$. In

order to further exploit meta information from the target domain, only the lane predictions with the highest confidence scores will be kept when the selected pseudo lane labels exceed the maximum number of available lanes $N_{\mathcal{D}}$ from the target domain. Note that backpropagation of the teacher model is deactivated during the adaptation process, as the teacher model is merely updated by the student model from the target domain with Exponential Moving Average (EMA) strategy [Tar17]:

$$\theta'_t = \omega_{\text{EMA}}\theta'_{t-1} + (1 - \omega_{\text{EMA}})\theta_t \tag{8.1}$$

where θ'_t denotes the parameters of the teacher network at step t, θ_t represents the parameters of the student network at step t, and ω_{EMA} is a smoothing coefficient that controls the update rate. With EMA, the teacher network also implicitly receives information from the target domain as learned by the student network, whose parameters are directly influenced by the gradients that are backpropagated into the network, as the prediction from the teacher and student networks must exhibit consistency. The loss function for the lane detection task, weighted by $\omega_{\text{det.}}$, is calculated with

$$\mathcal{L}_{\text{det.}} = \omega_{\text{det.}}\mathcal{L}_D\left(\bar{y}_{\text{det.}}, \hat{y}_{\text{det.}}\right) \tag{8.2}$$

where $\mathcal{L}_D$ denotes the loss function from the anchor-based detector, $\bar{y}_{\text{det.}}$ is the pseudo lane label generated by teacher model and $\hat{y}_{\text{det.}}$ is the lane predicted by the student model.

Auxiliary Task

Instance segmentation is frequently used as an auxiliary task for lane detection models [Qin20, Zhe22], as the vanilla lane annotations, which are based on key points of the lanes, can be interpolated and transferred to pixel-level segmentation labels given a certain width of the lane markings. WSDAL also supports various segmentation tasks, including instance segmentation, as an auxiliary task during the training of lane detection models. In the ablation study, the impact of leveraging different segmentation tasks will be further

evaluated. The segmentation head shares feature representations that are extracted from the image encoder. For multi-scale feature maps, the features are downsampled with bilinear interpolation to be connected with the segmentation head.

Similar to the lane detection task on the target domain, the teacher model generates a segmentation map $\hat{y}_{\text{seg.}}$ of c channels according to the input image x_t without any data augmentation. The pseudo segmentation label is generated based on filtering the softmax probability score according to a predefined threshold λ_{LS}:

$$\bar{y}_{\text{seg.}} = \begin{cases} \text{argmax}_c\, \hat{y}_{\text{seg., c}}, & \max \hat{y}_{\text{seg., c}} > \lambda_{\text{LS}} \\ 0, & \text{else} \end{cases} \tag{8.3}$$

Therefore, the weighted loss functions for the segmentation task on the target and source domain are defined as follows:

$$\mathcal{L}^{T}_{\text{seg.}} = \omega^{T}_{\text{seg.}} \mathcal{L}_{\text{CE}} \left(\bar{y}_{\text{seg.}}, \hat{y}_{\text{seg.}} \right) \tag{8.4}$$

$$\mathcal{L}^{S}_{\text{seg.}} = \omega^{S}_{\text{seg.}} \mathcal{L}_{\text{CE}} \left(y_{\text{src.}}, \hat{y}_{\text{seg.}} \right) \tag{8.5}$$

Weak Supervision Task

As shown by Honda et al. [Hon24], the misalignment of the confidence scores and the lane IoU leads to difficulties in training and inferring on in-distribution and out-of-distribution data, as usually the correct lanes exist among the predictions from the lane anchors. Such a property also impedes the process of filtering out potential correct lane predictions for pseudo labels. In this chapter, an additional loss function is introduced to leverage the known Number-of-Lane (NoL) information that can be easily extracted from the metadata. This function assists the lane detector in adapting to the target domain efficiently. For each lane prediction l_n in y during domain adaptation, one major objective is that the predicted foreground score $l_n^{(L-1)}$ and the background score $l_n^{(L)}$ satisfy $\max(l_n^{(L-1)}, l_n^{(L)}) \approx 1$ after the softmax function.

During the calculation of the number of lane loss, the lane predictions are filtered with high confidence scores from the student model by $l_n^{(L-1)} > \lambda_{\text{LD}}$ as positive classifications. The sum of their confidence scores is taken as the prediction of the number of lanes currently achieved by the model. In the initial stage, this value is usually smaller than the number of lanes during domain adaptation. Therefore, the model under the supervision of weak labels increases the confidence score of lane predictions through the propagated gradients, especially the lane predictions whose confidence score is higher than λ_{LD} in the initial stage. This helps the lane predictions with confidence scores above the threshold to diverge from the other lane predictions. As the model is trained, the number of lane predictions that exceed the threshold λ_{LD} is anticipated to exceed the ground truth lanes. By this time, the confidence scores of both the positive predictions and a portion of the negative predictions have become high, which encourages the lane anchors that do not represent the labeling policy to increase their background scores. The weighted number of lanes loss for the weak supervision is calculated by

$$\mathcal{L}_{\text{num.}} = \omega_{\text{num.}} \mathcal{L}_{\text{smooth-L}_1} \left(n_t, \sum_{i \in \mathcal{A}_{\text{L}}} l_i^{(L-1)} \right) \tag{8.6}$$

where n_t is the number of lanes and lane anchor set $\mathcal{A}_{\text{L}}=\{i \mid l_i^{(L-1)} > \lambda_{\text{LD}}\}$. It is noteworthy that the model does **not** acquire any meta information from the environment during inference after the domain adaptation process.

Training Process

As shown in Fig. 8.1, three major paths are inferring the data from the source and target domain for the weakly supervised domain adaptation:

1 Target images are fed to the teacher model to generate pseudo lane labels for the lane detection task and the auxiliary lane segmentation task based on the predefined detection and segmentation thresholds λ_{LD} and λ_{LS}. The teacher model is not directly affected by calculating the gradients. Instead, it is updated by exponential moving average of the weights from the student model.

2 For the student model, the target domain images are heavily augmented. Backpropagation through the comparison of lane detection predictions and lane segmentation predictions generates gradients for optimizing the student model. Furthermore, the foreground scores of a set of selected predictions with confidence scores above λ_{LD} are used to derive the NoL loss $\mathcal{L}^{T}_{\text{num.}}$ for the weak supervision.

3 For the third path shown in black, the losses from the source domain, where the exhaustive lane annotations are available, are calculated based on detection key points ($\mathcal{L}^{S}_{\text{det.}}$) and the lane segmentation task ($\mathcal{L}^{S}_{\text{seg.}}$) as introduced in Sec. 2.3 to ensure stability during the adaptation process.

The overall loss function is stated as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L}^{T}_{\text{det.}} + \mathcal{L}^{T}_{\text{seg.}} + \mathcal{L}^{T}_{\text{num.}} + \mathcal{L}^{S}_{\text{det.}} + \mathcal{L}^{S}_{\text{seg.}} \tag{8.7}$$

8.3 Experiments

8.3.1 Experimental Setup

To simplify the experimental setup, this work concentrates on two major anchor-based lane detection architectures: LaneATT [Tab21] and CLRNet [Zhe22]. These two network architectures serve as baseline models for further comparison and optimization [Wan22b, Hon24]. The AdamW optimizer with the cosine annealing learning rate strategy is applied for training on the source domain. In addition, the default image resolution settings for each dataset are applied as prescribed by their official implementations. The image encoders are pretrained on ImageNet-1k.

During domain adaptation, the models are first trained on their corresponding source dataset and then share identical weights as a starting point. Unless specified differently, the experiments are conducted three times with a predefined list of seeds for a fair comparison. ResNet-18 is used as the primary image encoder for the experiments. For comparison, other mainstream backbones that are frequently used in lane detection research, e.g., ConvNeXt, DLA, and ERFnet, are also evaluated. The heavy data augmentation used for training the student network contains horizontal flip, channel shuffle, color jitter, motion blur, median blur, random rotation, and random scaling. If not mentioned otherwise, binary segmentation instead of instance segmentation for the lane markings is used as an auxiliary task for the lane detection task. The thresholds, including λ_{LD} and λ_{LS}, are determined empirically, aiming for the best performance on the source and target domains based on a step size of 0.1. To further enrich the comparison of the weakly supervised paradigm for the lane detection task, the experiments also contain two baseline variants: one that directly runs the evaluation without utilizing any adaptation methods, noted as the Domain Generalization evaluation scheme (DG), and the one that uses deactivated NoL loss to perform the domain adaptation in an unsupervised fashion, noted as Teacher–Student Unsupervised Domain Adaptation (TSUDA).

8.3.2 Target Dataset TuSimple

The experiments start with transferring to the TuSimple dataset, which represents homogeneous US highway driving scenarios and is considered the least challenging dataset in the whole evaluation. Tab. 8.2 depicts the comparison of the domain adaptation performance utilizing various domain adaptation methods. Two base architectures are evaluated, and it is observed that the LaneATT model shows poorer generalization ability in comparison with the CLRNet model, which is consistent with the findings in the preliminary study in Sec. 8.1. Generally speaking, the proposed weakly-supervised domain adaptation method significantly improves the performance of the model on the target domain, as shown by the higher F1-scores and accuracies as well as the lower FPRs and FNRs. For architecture like LaneATT, the model is able to close the gap to the other architecture in the weakly-supervised setup, despite its comparatively weak performance without any adaptation or with only unsupervised adaptation. Across all backbone setups, from the lightweight ERFnet encoder with 2.4M parameters to the more complex DLA that contains 15.8M parameters, the proposed WSDAL demonstrates its advantage by further closing the performance discrepancy to the upper bound.

Moreover, it is noticeable that pretraining on different datasets has a significant impact on the model's generalization performance. For instance, the models that are previously trained on CurveLanes, which inherently contains more challenging driving scenarios, show significant advantages. In certain backbone combinations, e.g., with ERFnet, the UDA model even outperforms the CULane-pretrained WSDAL model, which is an indication that the model is able to re-use the learned sophisticated feature representations. In addition to that, it is shown that, with appropriate pretraining, the proposed weakly-supervised domain adaptation still shows its performance advantage over its unsupervised counterpart. With ResNet-18 encoder, the model is able to increase its F1-score by 6.7% on the TuSimple dataset and shows only a 4% difference to the upper bound trained directly on the target dataset without any manual annotation. From the lower FPRs, it can be concluded that the model is able to comprehend the priority of the lane candidates for the detection that share generalized properties of lane markings. Therefore, WSDAL also helps

suppress visual properties that are frequently detected as lane markings but are not annotated as lanes, e.g., road boundaries, tire marks, and road surface shifts.

Last but not least, feature encoders with higher learning capacity demonstrate their advantage in re-using the feature representations learned from the source domain, as ResNet-18 (with 11.7M parameters) and DLA significantly narrow the performance disparity to their corresponding upper bounds compared with smaller encoders like ConvNeXt-v2-atto that only contains 3.9M parameters and ERFnet. One possible explanation is that the smaller encoders struggle to generate high-quality pseudo labels to guide the domain adaptation, while it is considerably easier for the complex backbones.

Table 8.2: Comparison of the LaneATT and CLRNet models with various image encoders which are pretrained on the CULane or CurveLanes datasets and adapted to the TuSimple dataset. The scores are shown in percent.

$\mathcal{D}_s$	Model	Backbone	Method	F1	FPR	FNR	Accuracy
CULane	LaneATT	ResNet-18	DG	54.9	49.6	39.4	81.29
			TSUDA	$69.9_{\pm 0.7}$	$32.6_{\pm 0.7}$	$25.4_{\pm 0.6}$	$84.7_{\pm 0.2}$
			WSDAL	$81.7_{\pm 0.3}$	$18.8_{\pm 1.0}$	$17.5_{\pm 0.7}$	$88.8_{\pm 0.4}$
	CLRNet		DG	70.3	32.6	26.35	85.89
			TSUDA	$84.8_{\pm 0.3}$	$12.6_{\pm 0.1}$	$18.9_{\pm 0.9}$	$85.5_{\pm 0.7}$
			WSDAL	$86.9_{\pm 0.3}$	$11.5_{\pm 0.2}$	$15.2_{\pm 0.5}$	$89.0_{\pm 0.4}$
		ERFNet	DG	67.7	14.2	67.9	41.8
			TSUDA	$79.7_{\pm 0.4}$	$22.8_{\pm 0.3}$	$16.5_{\pm 0.6}$	$89.2_{\pm 0.4}$
			WSDAL	$81.5_{\pm 0.6}$	$19.4_{\pm 0.7}$	$17.1_{\pm 0.4}$	$89.5_{\pm 0.2}$
		ConvNeXt	DG	69.1	22.8	46.0	66.7
			TSUDA	$77.7_{\pm 0.9}$	$26.0_{\pm 0.9}$	$16.5_{\pm 1.0}$	$88.9_{\pm 0.5}$
			WSDAL	$81.6_{\pm 1.0}$	$19.7_{\pm 1.5}$	$16.6_{\pm 0.9}$	$89.5_{\pm 0.5}$
		DLA34	DG	70.4	12.9	60.4	49.1
			TSUDA	$81.7_{\pm 0.3}$	$20.3_{\pm 0.4}$	$15.4_{\pm 0.4}$	$89.4_{\pm 0.2}$
			WSDAL	$86.3_{\pm 0.3}$	$13.4_{\pm 0.3}$	$14.0_{\pm 0.4}$	$90.8_{\pm 0.2}$
CurveLanes	CLRNet	ResNet-18	DG	80.7	8.1	35.8	72.9
			TSUDA	$84.5_{\pm 0.2}$	$15.7_{\pm 0.5}$	$15.3_{\pm 0.6}$	$86.5_{\pm 0.4}$
			WSDAL	$91.2_{\pm 0.5}$	$8.1_{\pm 0.5}$	$9.7_{\pm 0.5}$	$91.2_{\pm 0.3}$
		ERFNet	DG	81.8	17.1	19.7	86.4
			TSUDA	$82.2_{\pm 0.0}$	$18.0_{\pm 0.0}$	$17.5_{\pm 0.0}$	$86.0_{\pm 0.0}$
			WSDAL	$87.2_{\pm 0.9}$	$12.4_{\pm 1.0}$	$13.2_{\pm 0.9}$	$90.5_{\pm 0.2}$
		ConvNeXt	DG	82.4	10.7	27.1	80.0
			TSUDA	$78.0_{\pm 2.1}$	$23.1_{\pm 1.9}$	$15.5_{\pm 1.9}$	$84.2_{\pm 0.6}$
			WSDAL	$88.9_{\pm 0.7}$	$10.7_{\pm 0.8}$	$11.6_{\pm 0.7}$	$90.4_{\pm 0.4}$
		DLA34	DG	86.6	10.7	17.0	87.2
			TSUDA	$84.0_{\pm 0.0}$	$18.1_{\pm 0.0}$	$13.2_{\pm 0.0}$	$88.3_{\pm 0.0}$
			WSDAL	$91.0_{\pm 0.1}$	$8.4_{\pm 0.1}$	$9.7_{\pm 0.2}$	$90.4_{\pm 0.2}$
$\mathcal{D}_T$	CLRNet	ResNet-18		95.3	5.5	3.8	95.1
$\mathcal{D}_T$	CLRNet	ERFNet		95.2	6.0	3.4	95.5
$\mathcal{D}_T$	CLRNet	ConvNeXt		95.8	5.6	2.8	95.6
$\mathcal{D}_T$	CLRNet	DLA34		96.3	4.1	3.2	95.6
$\mathcal{D}_T$	LaneATT	ResNet-18		95.1	5.9	3.7	94.9

8.3.3 Target Dataset CULane

As introduced in Sec. 2.4, the CULane dataset focuses on the urban driving situation in China with more significant object occlusion, various lighting, and weather conditions. Table 8.3 demonstrates the adaptation performance of models that previously trained on TuSimple and CurveLanes. When pretrained on the TuSimple dataset, similar to the previous findings, the models still exhibit a large disparity compared with models trained directly on the target dataset, although they improve markedly over the baseline without any adaptation. This remaining gap is caused by the more challenging conditions in comparison to the TuSimple dataset. LaneATT reveals its weakness in this setup when adapting from TuSimple to CULane, as the models based on this architecture are not able to improve the models' performance as its counterpart does, where the CLRNet with weak supervision doubles the detection performance as measured by the total F1-score.

When the models are pretrained on source datasets like CurveLanes, it can be observed that the models are able to achieve higher F1-scores in the target domain compared to the baseline variant that does not undergo any domain adaptation. According to the evaluation scheme of CULane, consistent improvements can be observed in the evaluation splits that contain *Crowd*, *Dazzle*, and *Night*, further reflecting the effectiveness of the proposed weakly supervised domain adaptation method. Furthermore, a noticeable decrease of false positive predictions in the category *cross roads* can be observed as the labeling policy of the dataset specifies that no lanes are supposed to be detected at the intersections. The improvements in the subset *Night* indicate that the model begins to emphasize the most salient visual structures during the domain adaptation, as the pretrained dataset contains only clear weather conditions while the target domain requires handling more complex driving situations without manual annotations. In addition, it can be observed that the unsupervised variants have sometimes worse performance compared to the baselines. One reason for this is the challenge of choosing an appropriate threshold, namely λ_{LD}, to generate pseudo labels from the lane detections. As shown in Honda et al. [Hon24], striking a balance between the quality and quantity of these labels is challenging, as correct lane predictions are typically

hidden among many candidates, yet the confidence scores are uncalibrated and the true lane IoU can hardly be estimated. Relying solely on score-based filtering leads to choosing the incorrect lane candidates and visual features that result in pseudo labels that do not accurately reflect the labeling policy of the target dataset.

Table 8.3: Comparison of the LaneATT and CLRNet models with various image encoders which are pretrained on TuSimple or CurveLanes datasets and adapt to CULane dataset. The scores are shown in percent.

$\mathcal{D}_s$	Model	Backbone	Method	Normal	Crowd	Dazzle	Shadow	Noline	Arrow	Curve	Cross	Night	Total
TuSimple	CLRNet	ResNet-18	DG	42.0	20.4	11.6	5.6	8.2	28.1	22.3	1950	2.9	23.0
			TSUDA	$59.9_{\pm 0.7}$	$40.2_{\pm 0.5}$	$32.3_{\pm 1.4}$	$24.9_{\pm 0.8}$	$26.3_{\pm 0.8}$	$54.1_{\pm 0.5}$	$44.6_{\pm 0.7}$	$4510_{\pm 388}$	$30.2_{\pm 1.5}$	$43.0_{\pm 0.9}$
			WSDAL	$65.2_{\pm 0.4}$	$43.4_{\pm 0.7}$	$35.6_{\pm 1.4}$	$28.2_{\pm 1.5}$	$29.3_{\pm 0.5}$	$57.5_{\pm 0.7}$	$48.7_{\pm 0.2}$	$1975_{\pm 178}$	$33.4_{\pm 0.8}$	$47.2_{\pm 0.6}$
	LaneATT		DG	38.9	18.1	10.9	3.9	6.5	26.1	21.4	569	2.6	21.5
			TSUDA	$33.6_{\pm 0.8}$	$19.8_{\pm 0.4}$	$19.9_{\pm 0.7}$	$10.8_{\pm 0.2}$	$13.5_{\pm 0.3}$	$25.6_{\pm 0.3}$	$21.3_{\pm 0.8}$	$7378_{\pm 250}$	$9.0_{\pm 0.3}$	$21.4_{\pm 0.5}$
			WSDAL	$49.0_{\pm 0.9}$	$28.0_{\pm 0.7}$	$25.2_{\pm 1.4}$	$15.6_{\pm 0.9}$	$22.0_{\pm 0.2}$	$37.3_{\pm 0.8}$	$36.6_{\pm 0.3}$	$2560_{\pm 90}$	$21.6_{\pm 0.7}$	$32.8_{\pm 0.6}$
CurveLanes	CLRNet	ResNet-18	DG	82.2	66.0	60.2	70.1	48.0	76.8	72.0	6730	62.9	66.7
			TSUDA	$68.5_{\pm 0.5}$	$50.1_{\pm 0.6}$	$42.4_{\pm 0.4}$	$48.5_{\pm 0.7}$	$30.8_{\pm 0.4}$	$63.6_{\pm 0.7}$	$56.5_{\pm 0.3}$	$889_{\pm 21}$	$51.7_{\pm 0.5}$	$55.0_{\pm 0.5}$
			WSDAL	$86.2_{\pm 0.0}$	$68.5_{\pm 0.2}$	$60.1_{\pm 0.2}$	$72.7_{\pm 0.1}$	$47.3_{\pm 0.1}$	$80.5_{\pm 0.2}$	$73.0_{\pm 0.4}$	$2370_{\pm 15}$	$66.6_{\pm 0.1}$	$70.9_{\pm 0.1}$
		ERFNet	DG	80.1	65.2	56.6	63.6	44.5	74.0	68.0	4070	61.7	65.9
			TSUDA	$71.5_{\pm 0.2}$	$54.6_{\pm 0.2}$	$44.1_{\pm 0.3}$	$54.5_{\pm 0.2}$	$34.2_{\pm 0.2}$	$67.3_{\pm 0.0}$	$59.3_{\pm 0.3}$	$5190_{\pm 156}$	$53.6_{\pm 0.2}$	$56.4_{\pm 0.2}$
			WSDAL	$85.8_{\pm 0.1}$	$68.3_{\pm 0.1}$	$60.9_{\pm 0.3}$	$69.5_{\pm 0.0}$	$46.0_{\pm 0.2}$	$80.8_{\pm 0.2}$	$68.5_{\pm 0.5}$	$2642_{\pm 59}$	$66.6_{\pm 0.2}$	$70.4_{\pm 0.2}$
		ConvNeXt	DG	78.5	63.6	54.6	66.3	42.4	72.7	69.0	3208	59.3	64.7
			TSUDA	$49.9_{\pm 0.4}$	$27.5_{\pm 0.2}$	$26.1_{\pm 0.4}$	$40.4_{\pm 0.8}$	$20.8_{\pm 0.2}$	$39.6_{\pm 0.5}$	$47.4_{\pm 0.2}$	$5803_{\pm 144}$	$34.5_{\pm 0.2}$	$35.5_{\pm 0.2}$
			WSDAL	$86.7_{\pm 0.2}$	$68.2_{\pm 0.2}$	$62.8_{\pm 0.3}$	$71.1_{\pm 0.2}$	$46.7_{\pm 0.2}$	$81.5_{\pm 0.3}$	$70.8_{\pm 0.2}$	$2311_{\pm 12}$	$67.2_{\pm 0.3}$	$71.0_{\pm 0.2}$
		DLA34	DG	83.2	67.8	62.1	71.8	48.7	78.9	73.9	5973	64.8	68.4
			TSUDA	$67.1_{\pm 0.4}$	$51.8_{\pm 0.6}$	$41.5_{\pm 0.6}$	$58.0_{\pm 0.2}$	$34.0_{\pm 0.5}$	$63.9_{\pm 0.5}$	$58.0_{\pm 0.1}$	$5142_{\pm 41}$	$50.8_{\pm 0.5}$	$53.8_{\pm 0.4}$
			WSDAL	$87.9_{\pm 0.2}$	$70.9_{\pm 0.1}$	$64.9_{\pm 0.3}$	$74.2_{\pm 0.4}$	$48.6_{\pm 0.3}$	$84.0_{\pm 0.2}$	$72.2_{\pm 0.2}$	$2448_{\pm 33}$	$68.6_{\pm 0.3}$	$72.8_{\pm 0.1}$
$\mathcal{D}_T$	CLRNet	ResNet-18		93.0	77.5	70.1	77.4	52.7	89.3	68.2	1239	77.5	78.5
$\mathcal{D}_T$	CLRNet	ERFNet		92.0	75.6	69.2	77.4	48.3	88.1	63.4	1030	72.0	77.3
$\mathcal{D}_T$	CLRNet	ConvNeXt		91.1	73.6	67.9	72.5	48.3	85.5	66.1	1025	71.1	76.1
$\mathcal{D}_T$	CLRNet	DLA34		93.1	77.8	71.4	81.9	52.5	89.6	69.9	1127	74.0	79.1
$\mathcal{D}_T$	LaneATT	ResNet-18		91.0	73.0	65.6	76.7	48.8	86.7	65.1	1173	70.1	75.5

8.3.4 Target Dataset CurveLanes

As observed in the previous sections, the CurveLanes dataset shares some common properties with the CULane dataset, with significantly more annotated lanes (up to 14 lanes per image) and a smaller radius of curvature from the lane markings, bringing even more challenging driving scenes. For the evaluation on the CurveLanes dataset, the experiments are conducted solely based on the CLRNet architecture, as the LaneATT architecture does not exhibit promising generalization performance in the previous setups. Tab. 8.4 demonstrates the comparison of CLRNet architectures evaluated by three IoU thresholds. When evaluated with the standard IoU threshold of 0.5, it can be observed that the F1-score of the model increases by nearly 3% in comparison with the baseline model, mainly contributed by a significantly higher recall value that indicates that the models miss fewer available lanes in the image. The decreased precision can be explained by the increasing number of predictions that are inspired by the pseudo labels, which do not have a positional constraint for regression during domain adaptation as NoL is the only available information from the target domain. It is worth mentioning that leveraging weak labels does not provide any direct supervision regarding the positions of the lanes to the ego vehicle, which are highly variable depending on the annotator.

To examine the hypothesis that slight lane position drift causes false positive predictions, similar to Pan et al. [Pan18b], loose evaluation schemes are also utilized for the evaluation that tolerates higher positional deviations of the predictions with the IoU threshold of 0.2 and 0.3. It can be seen that the disparity between the baseline and weak-supervised models increases when the tolerance for the position error increases, indicating the effectiveness of the proposed method.

The overall results from the task CULane → CurveLanes also correspond to the observations from Sec. 8.3.3, where networks with smaller learning capacity struggle to generate high-quality pseudo labels due to the quality-quantity dilemma for unsupervised domain adaptation. With the NoL information

from the target domain, the models are able to close the gap to the corresponding oracle models that were previously trained on the target dataset. With higher positional tolerances, the models bring performance improvements to a large extent by reducing the difference to 8% for the DLA-encoded model without any lane annotations on the target domain.

8.3.5 Qualitative Results

Figure 8.2 presents qualitative results obtained by adapting the model to target datasets. For visualization, an error-tolerant IoU threshold of 0.3 is employed. The WSDAL framework demonstrates a clear advantage, as it substantially reduces false-positive detections that are typically caused by road boundaries, variations in road surface material, or lane-marking-like structures, such as tire marks, as shown in rows a) and b) when adapting from CurveLanes to TuSimple dataset. Moreover, the network learns to regulate its confidence when multiple lanes are detected simultaneously. It preferentially retains the detections that exhibit high confidence while suppressing spurious ones. Row c) depicts the situation, which constitutes a domain shift toward a more challenging driving condition relative to the source dataset (CULane → CurveLanes). After applying WSDAL, the adapted models produce a greater number of predictions with higher confidence scores than both the unsupervised baseline and the model without domain adaptation. Row d) further highlights the ability of WSDAL to eliminate false positives by exploiting only weak supervision signals.

Table 8.4: Comparison of task CUlane → CurveLanes of CLRNet architectures based on various IoU thresholds.

				$IoU_{0.2}$			$IoU_{0.3}$			$IoU_{0.5}$	
$\mathcal{D}_s$	Backbone	Method	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall
CULane	ResNet-18	DG	69.5	90.1	52.3	67.6	92.0	53.5	61.4	82.5	48.5
		TSUDA	$70.6_{\pm 0.1}$	$94.9_{\pm 0.1}$	$56.2_{\pm 0.1}$	$68.8_{\pm 0.1}$	$92.5_{\pm 0.1}$	$54.8_{\pm 0.1}$	$62.4_{\pm 0.0}$	$83.9_{\pm 0.1}$	$49.7_{\pm 0.1}$
		WSDAL	$77.4_{\pm 0.1}$	$86.8_{\pm 0.1}$	$69.8_{\pm 0.2}$	$74.2_{\pm 0.1}$	$83.2_{\pm 0.2}$	$66.9_{\pm 0.2}$	$64.1_{\pm 0.1}$	$71.9_{\pm 0.1}$	$57.8_{\pm 0.2}$
	ConvNeXt	DG	67.4	93.8	52.6	65.4	91.1	51.1	58.9	82.0	46.0
		TSUDA	$66.1_{\pm 0.2}$	$83.1_{\pm 1.0}$	$54.9_{\pm 0.7}$	$63.7_{\pm 0.1}$	$80.1_{\pm 1.1}$	$52.9_{\pm 0.5}$	$53.6_{\pm 0.4}$	$67.4_{\pm 1.3}$	$44.5_{\pm 0.3}$
		WSDAL	$74.5_{\pm 0.1}$	$86.3_{\pm 0.3}$	$65.5_{\pm 0.2}$	$70.9_{\pm 0.0}$	$82.2_{\pm 0.3}$	$62.4_{\pm 0.2}$	$59.6_{\pm 0.1}$	$69.1_{\pm 0.3}$	$52.5_{\pm 0.1}$
	ERFNet	DG	70.8	91.6	57.8	68.5	88.5	55.9	61.7	79.7	50.3
		TSUDA	$69.9_{\pm 0.0}$	$91.5_{\pm 0.2}$	$56.5_{\pm 0.1}$	$67.9_{\pm 0.0}$	$88.9_{\pm 0.2}$	$54.9_{\pm 0.1}$	$61.5_{\pm 0.0}$	$80.5_{\pm 0.2}$	$49.7_{\pm 0.1}$
		WSDAL	$76.6_{\pm 0.0}$	$85.8_{\pm 0.1}$	$69.1_{\pm 0.1}$	$73.4_{\pm 0.0}$	$82.3_{\pm 0.2}$	$66.3_{\pm 0.1}$	$63.8_{\pm 0.1}$	$71.5_{\pm 0.2}$	$57.6_{\pm 0.0}$
	DLA34	DG	67.5	95.3	52.3	65.9	93.0	51.1	60.4	85.2	46.8
		TSUDA	$70.9_{\pm 0.2}$	$91.7_{\pm 0.3}$	$57.8_{\pm 0.4}$	$68.8_{\pm 0.1}$	$89.0_{\pm 0.4}$	$56.1_{\pm 0.3}$	$61.8_{\pm 0.1}$	$80.0_{\pm 0.5}$	$50.4_{\pm 0.2}$
		WSDAL	$78.0_{\pm 0.1}$	$88.7_{\pm 0.2}$	$69.6_{\pm 0.0}$	$74.9_{\pm 0.1}$	$85.2_{\pm 0.2}$	$66.8_{\pm 0.0}$	$65.2_{\pm 0.3}$	$74.1_{\pm 0.4}$	$58.2_{\pm 0.2}$
$\mathcal{D}_T$	ResNet-18		86.0	92.8	80.2	83.3	89.8	77.7	75.2	81.1	70.1
$\mathcal{D}_T$	ConvNeXt		85.5	93.0	79.1	83.0	90.3	76.8	75.4	82.0	69.7
$\mathcal{D}_T$	ERFNet		85.4	91.6	80.0	82.8	88.7	77.5	74.9	80.3	70.1
$\mathcal{D}_T$	DLA34		86.0	93.6	79.6	83.6	90.9	77.4	76.1	82.8	70.5

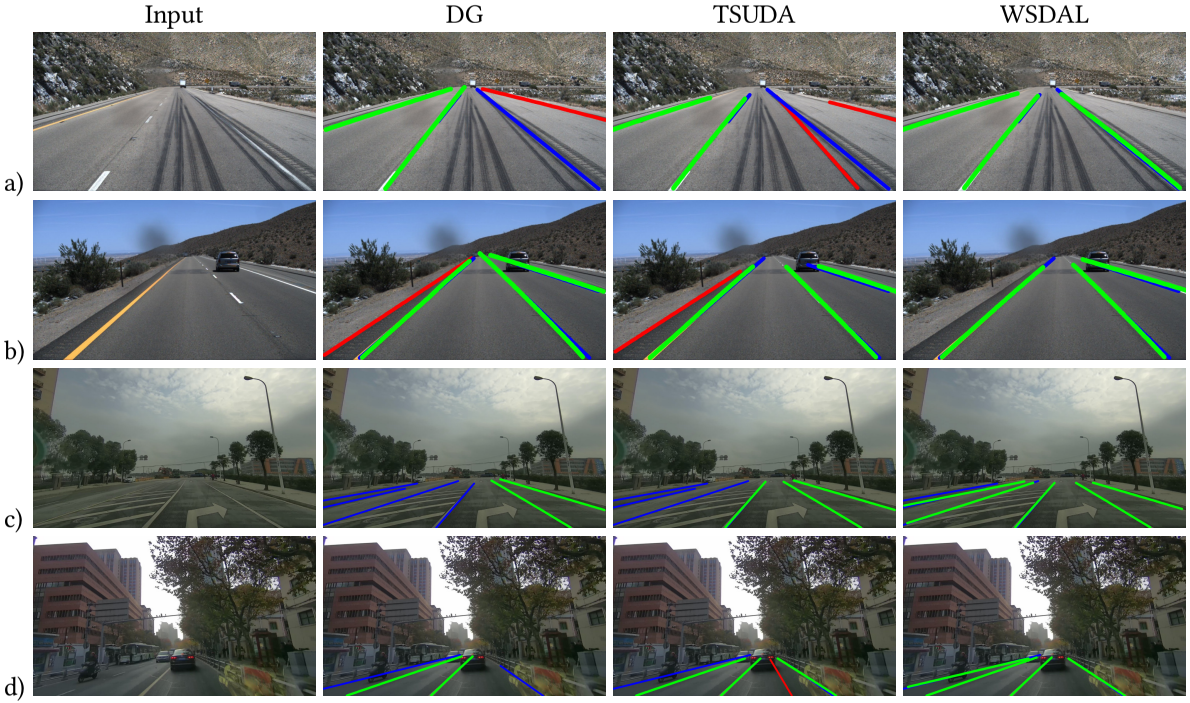


Figure 8.2: Qualitative results of various domain adaptation methods utilizing CLRNet with ResNet-18 backbone. TP predictions are shown in green, FP predictions in red, and ground truth in blue.

8.4 Ablation Study

Auxiliary Task

Although utilizing the segmentation task as an auxiliary task is widely adopted in lane detection networks [Qin20, Zhe22], the ablation study starts with the comparison of the performance on source domain data. As shown in Tab. 8.5, leveraging segmentation as an auxiliary task only provides limited improvement on the source domain. Besides that, the two segmentation tasks that are compared based on source domain show no notable distinction in their scores. However, Tab. 8.6 depicts the vanilla domain adaptation performance, where the models are directly inferred on the target domain after training on the source domain. The enhancement in the generalization ability with the auxiliary task is significant as the networks with binary segmentation auxiliary task constantly outperform the baseline and instance segmentation setup. As the auxiliary segmentation head shares the image embeddings, it forces the image encoder to concentrate on more generic textures and structures, which is beneficial for the lane detection task.

Table 8.5: The LaneATT and CLRNet models with segmentation tasks in the source domain CULane. All values in %.

Model	Segmentation	F1	Precision	Recall
CLRNet	—	77.8	84.6	72.0
CLRNet	Instance	78.5	85.9	72.3
CLRNet	Binary	78.8	84.5	73.8
LaneATT	—	75.5	82.2	69.8
LaneATT	Instance	76.3	82.8	70.7
LaneATT	Binary	76.0	82.7	70.4

Table 8.6: The CLRNet and LaneATT models with segmentation tasks are trained on CULane and evaluated on the target domain TuSimple. All values in %.

Model	Segmentation	F1	FPR	FNR	Accuracy
CLRNet	—	70.4	32.6	26.3	85.9
CLRNet	Instance	71.0	33.1	24.3	87.6
CLRNet	Binary	73.4	28.3	24.7	86.9
LaneATT	—	55.0	49.7	39.4	81.3
LaneATT	Instance	58.9	42.8	39.2	80.5
LaneATT	Binary	60.0	43.0	36.7	81.2

As it is evident that both auxiliary segmentation tasks bring better generalization ability, further experiments with weakly supervised domain adaptation setup are conducted with the results in Tab. 8.7. Similar to the results from the domain generalization part, the binary segmentation auxiliary task consistently outperforms the baseline and the frequently adopted instance segmentation scheme. One possible explanation is that the instance segmentation provides redundant information regarding the position of the lanes, as the lanes on the left side of the image are always assigned a smaller ID number than the lanes on the right side of the image. In addition, the lane detection task with the lane anchors only classifies whether the lane candidate belongs to the foreground (lane marking). Such characteristics render the binary segmentation task more effective during optimization.

Table 8.7: Ablation study for auxiliary segmentation task CULane → TuSimple. All values in %.

Model	Segmentation	F1	FPR	FNR	Accuracy
CLRNet	—	80.2	16.5	22.8	85.1
CLRNet	Instance	83.2	14.5	19.0	86.5
CLRNet	Binary	86.9	11.5	15.2	89.0
LaneATT	—	76.0	25.5	22.5	87.3
LaneATT	Instance	79.8	20.4	19.9	87.7
LaneATT	Binary	81.7	18.8	17.5	88.8

In addition, the upcoming subsection outlines the necessity of utilizing the teacher–student network in conjunction with the auxiliary task.

Ablation Teacher–Student Network

The WSDAL adapts the teacher–student network and EMA weight updating strategy as proposed and proven in [Soh20, Hoy22, Zha21a, Tra21]. The effects of leveraging the NoL loss, auxiliary segmentation task, and teacher–student network and their combinations are shown in Tab. 8.8. It can be observed that utilizing the NoL loss alone without any auxiliary task does not improve the detection performance, whereas integrating a segmentation task as an auxiliary task brings benefits. With the teacher–student network, the model is able to extract beneficial hints from the more accurate pseudo labels generated by the teacher network and thus achieve better detection performance on the target domain with WSDAL. Note that integrating the teacher–student network does not affect the model's inference time for deployment, as the steps are only taken during the weakly supervised domain adaptation process.

Table 8.8: Effects of adding a teacher–student network (TS) in the task CULane → TuSimple when CLRNet has different segmentation tasks. All values in %.

Method	Segmentation	NoL	TS	F1	FPR	FNR	Accuracy
DG	–			70.4	32.6	26.4	85.9
WSDAL	–	✓		69.1	29.4	32.3	82.4
WSDAL	–	✓	✓	80.2	16.5	22.8	85.1
DG	Instance			71.0	33.1	24.3	87.6
WSDAL	Instance	✓		77.1	20.6	25.1	84.5
WSDAL	Instance	✓	✓	83.2	14.5	19.0	86.5
DG	Binary			73.4	28.3	24.7	86.9
WSDAL	Binary	✓		82.7	15.1	19.4	90.2
WSDAL	Binary	✓	✓	86.9	11.5	15.2	89.0

Unreliable Labels

As there are constant changes in the driving environment so that the map data need to be updated, and potential inaccuracies in inference from the production vehicles, it is essential to discuss the potential impact of unreliable NoL

labels on weakly supervised domain adaptation. The main objective is to compare the minimum required label quality that assures that the models with weak supervision outperform their baseline models. CLRNet is selected as the base architecture for this evaluation, as it shows promising performance on the target domain. Similar to the experiments from Sec. 8.3, multiple backbones are evaluated in this setup to compare different image encoders. During domain adaptation, the training sessions share an identical setup of hyperparameters, with only the NoL labels from the target dataset $\mathcal{D}_T$ replaced with a random number between zero and the maximum number of lanes that is available from the dataset. The experiments are conducted with a step size of 5% up to 100% error rate. The experiments are repeated three times for each error rate using different seeds.

Figure 8.3 depicts the progression of the model's performance based on the proportion of the incorrect NoL labels from three frequently used backbones on the task of CurveLanes→TuSimple. Their corresponding baseline performance from unsupervised domain adaptation is marked by the dashed lines. The transition points are located at around 30% error rate, indicating that the proposed WSDAL model outperforms the conventional unsupervised domain adaptation methods when the label error rate is below this threshold.

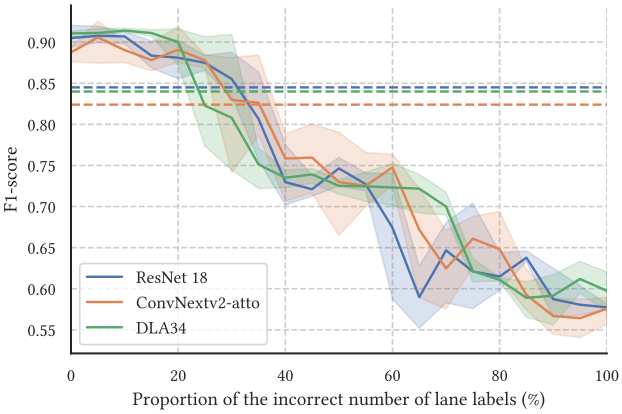


Figure 8.3: Results on CurveLanes → TuSimple with incorrect NoL labels on various image encoders with CLRNet. The dashed lines show the results from TSUDA, demonstrating the robustness to label inaccuracies.

In order to investigate the impact of the source dataset and model architecture, the characteristics of the model trained by the source domain data and the architectural design of the model are ablated. First, the model with the same initialization is utilized to train on two source datasets: the CULane and the CurveLanes dataset, with the remaining parameters and hyperparameters unchanged and repeat the experiments to understand the impact of the source dataset on the robustness of dealing with incorrect meta information. The result is shown in Fig. 8.4 as it can be observed that the network that previously trained on CurveLanes performs better on the adaptation task and can tolerate higher label errors when the label error is kept in a considerable amount (under 40%). With the increase in the label error rate, the model shows instability and is significantly outperformed by its unsupervised counterpart. In contrast, the curve for CULane remains relatively flat.

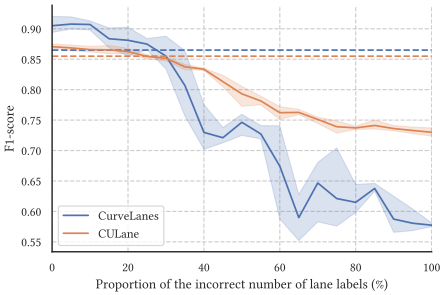


Figure 8.4: Comparison of using different datasets CurveLanes and CULane for the adaptation to TuSimple with incorrect NoL labels based on CLRNet with ResNet-18 backbone. The dashed lines show the results from TSUDA.

In addition, this work also compares the tolerance of incorrect meta labels across different network architectures. The evaluation is based on the task of transferring from TuSimple to CULane, which is considered more complex for adapting the models from the highway dataset to the urban driving dataset. As shown in Fig. 8.5, models become more resistant to label disturbances as the transition point significantly shifts to the right side, tolerating unreliable NoL labels. However, the disparity among baseline models in their unsupervised domain adaptation setup reveals differences in the generalization ability of their lane detection heads.

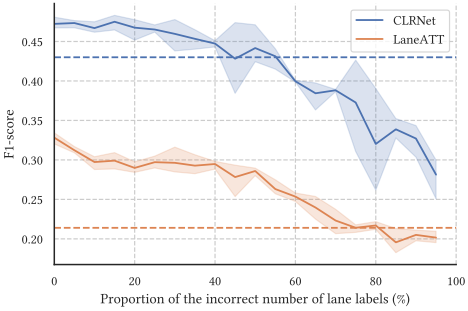


Figure 8.5: Results of two detector architectures adapting from TuSimple to CULane with incorrect NoL labels. The dashed lines show the results from TSUDA.

8.5 Conclusion

This chapter introduces a novel Weakly Supervised Domain Adaptation framework for Lane Detection (WSDAL) that leverages solely meta information from previously unseen driving environments to mitigate corner cases for the lane detection models. The weakly supervised domain adaptation consists of three major parts: An anchor-based lane detection model, an auxiliary segmentation task that shares the image encoder to improve the generalization ability, and the novel Number-of-Lanes prediction based on the confidence scores from the lane anchors. With the help of the teacher–student network from the framework, the teacher network is able to generate accurate lane detection and segmentation predictions, which are selected and converted to pseudo labels for the supervision of the student network. The effectiveness of the proposed framework is demonstrated by an extensive range of experiments containing three major lane detection datasets covering highway driving scenarios from TuSimple to complex urban scenarios from CurveLanes, showing that the lane labels enhance the confidence of lane predictions across the frequently used image encoders and thus further encourage the detector to create lane predictions with high confidence scores in the target domain. Consequently, the network is able to prioritize the lane detections with significant visual properties and textures when the annotation policy shifts between the datasets, solving the ambiguities of dealing with road boundaries and lane-like structures on the road. Furthermore, the proposed method is robust to disturbances from the NoL labels, as label quality cannot be guaranteed in real applications.

The proposed framework demonstrates how publicly available meta information can be leveraged to mitigate interpretation problems in the neural networks without extensive manual data annotation. However, as it is shown in Sec. 8.4, segmentation maps also contribute to the generalization ability of the network. Future work may concentrate on how those weak labels can also be utilized for lane segmentation models and other types of auxiliary segmentation tasks, such as deriving driveable areas.

9 Conclusion and Outlook

This thesis aims, on the one hand, to provide categorizations and metrics for assessing corner cases, which are frequently regarded as safety-critical events for assisted and automated driving functions, from a data perspective, to enhance the understanding of their occurrence. On the other hand, it aims to ensure that such corner cases can be mitigated efficiently in advance by leveraging contextual knowledge from various sources.

Although widely discussed, the definition of corner cases remains highly ambiguous, hindering systematical understanding and subsequent mitigation of these safety-relevant cases. Starting from a preliminary study from Ch. 4, the impacts of data anonymization on semantic segmentation are evaluated qualitatively and quantitatively. It is observed that the absence of an essential part of the dynamic objects, in this case, the human faces, leads to statistically significant impairment of segmentation performance in data-driven models, especially for the missegmented occluded pedestrians, which are often regarded as corner cases. However, if such properties are preserved during postprocessing, e.g., via generative models that reconstruct the original data distribution, model performance is not affected by the required anonymization step. By connecting the network errors and the availability of the data that are utilized for training, the first research question is raised:

Research Question 1. *What are the core properties of corner cases in the context of real-world ADAS and HAD applications?*

In Ch. 5, it is shown that the corner cases can also be classified according to the availability of the data points during development and validation, in case the corner cases are caused by interpretation problems of the data-driven neural networks, as other factors caused by hardware limitations and risks

of certain driving behavior are also present in classical, handcrafted functions. For discussion, the assumption is taken that the *development dataset* is considered a joint effort of all the parties involved, ranging from OEMs to certification bodies. Data appearing outside the *development dataset* are assumed to be encountered by customers. Consequently, this section provides a comprehensive examination of potential technical solutions aimed at advancing the capabilities of data-driven driving functions to deal with such out-of-distribution cases.

Given the significant cost and labor intensity associated with large-scale data annotation for mitigating such corner cases of scene understanding tasks, the subsequent sections of this thesis focus on leveraging existing contextual knowledge rather than relying solely on manual annotation of the raw data. This approach has the advantage that potential corner cases can be mitigated without necessitating additional effort for data collection or annotation. One major aspect of contextual knowledge is the hierarchical taxonomy of object classes, which leads to the second research question:

Research Question 2. *How can hierarchical taxonomies be leveraged in downstream computer vision tasks to evaluate and improve the generalization ability of the trained neural networks?*

Beginning with the object detection model, it is shown in Ch. 6 that utilizing the class hierarchy is advantageous when dealing with unknown object classes, while the known classes share the same object category with them during training. The experiments examine various training strategies that incorporate class hierarchies to evaluate their effectiveness in detecting unknown objects while preserving the ability to classify learned known classes.

Subsequently, Sec. 7.1 evaluates how a use-case-relevant class taxonomy can facilitate the evaluation of domain generalization of semantic segmentation models when encountering novel classes. This section introduces a novel evaluation metric, Critical Error Rate (CER), for semantic segmentation that explicitly addresses the differentiation between intra- and inter-category errors. Experiments on a series of scene understanding datasets reveal that the optimization process for underrepresented classes is heavily influenced

by appearance-similar classes and reflects the importance of data diversity, which has a more significant impact on neural network performance than the learning capacity of the models.

Chapter 6 and Sec. 7.1 demonstrate not only that the class hierarchy is task-agnostic and applicable to various downstream tasks for data-driven scene understanding but also the necessity of defining class taxonomies that are specifically tailored to the use-case and visual similarity of objects. The ablation studies investigate the effect of using class taxonomies that are appearance-independent for category-related semantic segmentation and unknown-aware object detection evaluation.

It is shown that the class hierarchy can be utilized to evaluate domain generalization performance and detect previously unknown classes. For error mitigation, it is essential to learn the underrepresented novel classes by leveraging the limited amount of collected training data. Therefore, the third research question is raised as

Research Question 3. *Does the class hierarchy reduce the required effort for training in computer vision tasks since the training process becomes more effective and the amount of data needed for training decreases?*

Section 7.2 focuses on the evaluation of few-shot semantic segmentation that is applied to complex driving environments. This investigation highlights discrepancies in the distribution of foreground and background classes by identifying the distinct challenges inherent in these environments, compared with traditional object-centric datasets. It is evident that the sparse occurrence of foreground dynamic objects adversely affects the efficacy of FSSS. This study proposes novel strategies for leveraging prior knowledge, e.g., incorporating the class hierarchy, implementing effective feature extraction methods, and selecting support images during the inference phase. These approaches improve the performance and resilience in identifying and segmenting new objects with limited information from the support image set.

Last but not least, this thesis also takes other easy-to-obtain prior knowledge in addition to the class hierarchy that can be utilized to solve the interpretation problems in data-driven solutions:

Research Question 4. *Beyond class taxonomies, how can other contextual knowledge be leveraged to mitigate corner cases?*

Chapter 8 addresses the mitigation of corner cases in lane detection models by utilizing map information from the unseen driving scenes during training, e.g., Number-of-Lanes (NoL) labels. With the proposed Weakly Supervised Domain Adaptation framework for Lane Detection (WSDAL), the findings indicate that the integration of lane labels improves the performance in lane predictions across commonly used image encoders, thereby enhancing the detector's capability to produce high-confidence lane predictions in the target domain. It can be observed that the adapted data-driven neural networks effectively prioritize lane detections with distinct visual properties and textures amid variations in annotation policies between datasets, thereby resolving ambiguities associated with road boundaries and lane-like structures. Furthermore, the proposed method exhibits robustness to disturbances from weak labels, even when label quality cannot be guaranteed in practical applications. This exemplifies how publicly available meta information can be utilized to alleviate interpretation problems in neural networks without the need for extensive manual data annotation.

In summary, the findings of this thesis demonstrate how corner cases can be defined and classified for data-driven applications according to availability of data during development. Subsequently, this perspective enables systematic mitigation of corner cases grounded in their foundational root causes. In addition to the conventional methods of exhaustive data annotation, this work focuses on leveraging approaches that utilize easy-to-obtain prior knowledge, including class hierarchy and meta information. By employing the proposed methodologies, the robustness and learning efficiency of the deep learning-based solutions are demonstrably increased when confronted with unforeseen scenarios in practical applications.

Outlook

Leveraging Foundation Models

The recent advent of large-scale pretrained models, commonly referred to as Foundation Models [Rad21, Xia24, Kir23, Rav24, Car26], marks a paradigm shift in computer vision. One significant contributing factor is the evolution of Large Language Models (LLMs) [Ach23, Tea23, Guo25], which are trained on web-scale datasets. By connecting text with visual concepts using large-scale datasets, the models demonstrate advanced generalization capabilities in dealing with open world data across many downstream computer vision tasks. As shown by Miller et al. [Mil21], the out-of-distribution performance strongly correlates with the models' in-distribution ability. At the same time, text-image pairs are easier to obtain than manual annotations. In addition to that, there are text-conditioned generative models such as Stable Diffusion [Rom22] and DALL-E [Ram21], for generating realistic images given text input. Specifically, in the domain of assisted and automated driving functions, such models are used for high-fidelity synthetic data generation, scene reconstruction, scenario variation and provide explanations for the decision-making to enhance the explainability of such models [Mar24, Che24, Hu23a, Yan23b]. One prime example of the foundation model is SAM [Kir23, Rav24], a segmentation model that outputs class-agnostic masks based on image and geometric prompts. Pretrained on billion-scale datasets, SAM generalizes to tasks including medical imaging, image inpainting, 3D object detection, and human-robot interaction applications [Ma24, Car26]. However, studies show that the model underperforms on more challenging vision tasks involving camouflaged and transparent objects [Ji23].

Dataset Engineering

In the classical machine learning paradigm, the benchmark methodology follows the principle of a fixed dataset and varying network architectures. However, as also observed in Sec. 7.1, the quality of the data plays an essential role in the generalization ability of the network in addition to the quantity

of the data used for training. DataComp [Gad24] proposed a novel evaluation scheme where the training budget and the model are fixed. The main objective is to filter the optimal subset to achieve the best performance, as large-scale multimodal datasets pose new challenges in terms of their size and annotation schemes.

It is also shown that the training distribution is the main reason for the robustness of the CLIP-based models, rather than language supervision or contrastive learning paradigm [Fan22a]. By comparing the similarity of the caption and the masked image concentrating on visual properties, Maini et al. [Mai24] showed that their proposed method further improved the model's performance on datasets both with and without distribution shift. Leveraging large-scale pretrained models for active learning also facilitates sample selection for visual question answering and image classification tasks [Ban24].

Trustworthy AI Methods and Technical Conformity

The inherent complexity of tasks, environments, and systems poses significant challenges in providing safety assurance arguments for AI/ML-based automated driving systems. Existing standards, such as ISO 26262 for functional safety [ISO18] and ISO 21448 for the safety of intended functionality [ISO22], emphasize the need to mitigate risks associated with hardware and software errors, as well as functional insufficiencies and foreseeable misuse. There are also other concepts, e.g., ISO PAS 8800 [ISO24], which introduce principles vital for the safety of data-driven AI systems, including fault models, iterative safety lifecycles, development measures, and rigorous data lifecycle management. Addressing uncertainties in understanding environments, data accuracy, and system decision-making processes is critical. The aim is to provide vital guidelines, substantial task- and application-specific interpretation and adaptation, which are still needed to manage the complexity and ensure an acceptable level of residual risk for data-driven assisted and automated driving functions.

Bibliography

[Ach20] Acharya, Manoj; Hayes, Tyler L and Kanan, Christopher: "Rodeo: Replay for online object detection". In: *British Machine Vision Conference*. 2020 (cit. on p. 67).

[Ach23] Achiam, Josh; Adler, Steven; Agarwal, Sandhini; Ahmad, Lama; Akkaya, Ilge; Aleman, Florencia Leoni; Almeida, Diogo; Altenschmidt, Janko; Altman, Sam; Anadkat, Shyamal et al.: "Gpt-4 technical report". In: *arXiv preprint arXiv:2303.08774* (2023) (cit. on pp. 5, 159).

[Ahm16] Ahmed, Karim; Baig, Mohammad Haris and Torresani, Lorenzo: "Network of experts for large-scale image categorization". In: *European Conference on Computer Vision*. Springer. 2016, pp. 516–532 (cit. on p. 39).

[Aka15] Akata, Zeynep; Reed, Scott; Walter, Daniel; Lee, Honglak and Schiele, Bernt: "Evaluation of output embeddings for fine-grained image classification". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2015, pp. 2927–2936 (cit. on p. 39).

[Ba16] Ba, Jimmy Lei; Kiros, Jamie Ryan and Hinton, Geoffrey E: "Layer normalization". In: (2016) (cit. on p. 14).

[Bai21] Bai, Yutong; Mei, Jieru; Yuille, Alan L and Xie, Cihang: "Are transformers more robust than cnns?" In: *NeurIPS* 34 (2021), pp. 26831–26843 (cit. on pp. 18, 32, 80, 102).

[Ban24] Bang, Jihwan; Ahn, Sumyeong and Lee, Jae-Gil: "Active Prompt Learning in Vision Language Models". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*. IEEE, 2024,

pp. 26994–27004. DOI: 10.1109/CVPR52733.2024.02550. URL: https://doi.org/10.1109/CVPR52733.2024.02550 (cit. on p. 160).

[Bea16] Bearman, Amy; Russakovsky, Olga; Ferrari, Vittorio and Fei-Fei, Li: "What's the point: Semantic segmentation with point supervision". In: *European Conference on Computer Vision*. Springer. 2016, pp. 549–565 (cit. on p. 38).

[Ben10] Bengio, Samy; Weston, Jason and Grangier, David: "Label Embedding Trees for Large Multi-Class Tasks". In: *Proceedings of NeuRIPS*. Vol. 23. 2010 (cit. on p. 39).

[Ber20] Bertinetto, Luca; Mueller, Romain; Tertikas, Konstantinos; Samangooei, Sina and Lord, Nicholas A: "Making better mistakes: Leveraging class hierarchies with deep networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 12506–12515 (cit. on p. 39).

[Bho21] Bhojanapalli, Srinadh; Chakrabarti, Ayan; Glasner, Daniel; Li, Daliang; Unterthiner, Thomas and Veit, Andreas: "Understanding robustness of transformers for image classification". In: *CVPR*. 2021 (cit. on p. 32).

[Bi11a] Bi, Wei and Kwok, James T: "Multi-label classification on tree- and dag-structured hierarchies". In: *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*. 2011, pp. 17–24 (cit. on p. 76).

[Bi11b] Bi, Wei and Kwok, James T: "Multilabel classification on tree- and DAG-structured hierarchies". In: *ICML*. 2011 (cit. on p. 39).

[Bil17] Bilal, Alsallakh; Jourabloo, Amin; Ye, Mao; Liu, Xiaoming and Ren, Liu: "Do convolutional neural networks learn class hierarchy?" In: *IEEE transactions on visualization and computer graphics* 24.1 (2017), pp. 152–162 (cit. on p. 39).

[Boc20] Bochkovskiy, Alexey; Wang, Chien-Yao and Liao, Hong-Yuan Mark: "Yolov4: Optimal speed and accuracy of object detection". In: *arXiv preprint arXiv:2004.10934* (2020) (cit. on p. 20).

[Bog21] Bogdoll, Daniel; Breitenstein, Jasmin; Heidecker, Florian; Bieshaar, Maarten; Sick, Bernhard; Fingscheidt, Tim and Zöllner, Marius: "Description of Corner Cases in Automated Driving: Goals and Challenges". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021 (cit. on pp. 4, 30, 58, 59).

[Boj16] Bojarski, Mariusz; Del Testa, Davide; Dworakowski, Daniel; Firner, Bernhard; Flepp, Beat; Goyal, Prasoon; Jackel, Lawrence D; Monfort, Mathew; Muller, Urs; Zhang, Jiakai et al.: "End to end learning for self-driving cars". In: *arXiv preprint arXiv:1604.07316* (2016) (cit. on p. 3).

[Bol19a] Bolte, Jan-Aike; Bar, Andreas; Lipinski, Daniel and Fingscheidt, Tim: "Towards corner case detection for autonomous driving". In: *2019 IEEE Intelligent vehicles symposium (IV)*. IEEE. 2019, pp. 438–445 (cit. on pp. 4, 30, 58, 65).

[Bol19b] Bolya, Daniel; Zhou, Chong; Xiao, Fanyi and Lee, Yong Jae: "Yolact: Real-time instance segmentation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 9157–9166 (cit. on p. 3).

[Bre20] Breitenstein, Jasmin; Termöhlen, Jan-Aike; Lipinski, Daniel and Fingscheidt, Tim: "Systematization of Corner Cases for Visual Perception in Automated Driving". In: *IV*. IEEE. 2020 (cit. on pp. 4, 30, 58, 59).

[Bre21] Breitenstein, Jasmin; Termöhlen, Jan-Aike; Lipinski, Daniel and Fingscheidt, Tim: "Corner Cases for Visual Perception in Automated Driving: Some Guidance on Detection Approaches". In: *arXiv preprint arXiv:2102.05897* (2021) (cit. on pp. 58, 65).

[Bro20] Brown, Tom B. et al.: "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by Larochelle, Hugo; Ranzato, Marc'Aurelio; Hadsell, Raia; Balcan, Maria-Florina and Lin, Hsuan-Tien. 2020.

URL: https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html (cit. on p. 17).

[But17] Butepage, Judith; Black, Michael J; Kragic, Danica and Kjellstrom, Hedvig: "Deep representation learning for human motion prediction and classification". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2017, pp. 6158–6166 (cit. on p. 3).

[Cae18] Caesar, Holger; Uijlings, Jasper and Ferrari, Vittorio: "COCO-Stuff: Thing and stuff classes in context". In: *Proceedings of the CVPR*. IEEE. 2018 (cit. on p. 41).

[Cae20] Caesar, Holger; Bankiti, Varun; Lang, Alex H; Vora, Sourabh; Liong, Venice Erin; Xu, Qiang; Krishnan, Anush; Pan, Yu; Baldan, Giancarlo and Beijbom, Oscar: "nuScenes: A multimodal dataset for autonomous driving". In: *Proceedings of the CVPR*. 2020, pp. 11621–11631 (cit. on pp. 24, 60, 64, 73, 224).

[Can86] Canny, John: "A Computational Approach to Edge Detection". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8.6 (1986), pp. 679–698. DOI: 10.1109/TPAMI.1986.4767851 (cit. on pp. 3, 22).

[Car20] Carion, Nicolas; Massa, Francisco; Synnaeve, Gabriel; Usunier, Nicolas; Kirillov, Alexander and Zagoruyko, Sergey: "End-to-end object detection with transformers". In: *Proceedings of ECCV*. Springer. 2020, pp. 213–229 (cit. on p. 20).

[Car21] Caron, Mathilde; Touvron, Hugo; Misra, Ishan; Jégou, Hervé; Mairal, Julien; Bojanowski, Piotr and Joulin, Armand: "Emerging Properties in Self-Supervised Vision Transformers". In: *Proceedings of the ICCV*. 2021 (cit. on p. 79).

[Car26] Carion, Nicolas; Gustafson, Laura; Hu, Yuan-Ting; Debnath, Shoubhik; Hu, Ronghang; Suris Coll-Vinent, Didac; Ryali, Chaitanya; Alwala, Kalyan Vasudev; Khedr, Haitham; Huang, Andrew et al.: "Sam 3: Segment anything with concepts". In: *International Conference on Learning Representations*. Vol. 2026. 2026, pp. 138846–138923 (cit. on p. 159).

[Cen21] Cen, Jun; Yun, Peng; Cai, Junhao; Wang, Michael Yu and Liu, Ming: "Open-set 3D Object Detection". In: *2021 International Conference on 3D Vision (3DV)*. IEEE. 2021, pp. 869–878 (cit. on pp. 58, 63).

[Cha14] Chai, Yu; Wei, Su Jing and Li, Xin Chun: "The multi-scale Hough transform lane detection method based on the algorithm of Otsu and Canny". In: *Advanced Materials Research* 1042 (2014), pp. 126–130 (cit. on p. 22).

[Cha19] Chalapathy, Raghavendra; Toth, Edward and Chawla, Sanjay: "Group anomaly detection using deep generative models". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2019 (cit. on pp. 58, 65).

[Cha20] Chattopadhyay, Prithvijit; Balaji, Yogesh and Hoffman, Judy: "Learning to balance specificity and invariance for in and out of domain generalization". In: *ECCV*. Springer. 2020, pp. 301–318 (cit. on pp. 58, 67).

[Che04] Chen, Yangchi; Crawford, M.M. and Ghosh, J.: "Integrating support vector machines in a hierarchical output space decomposition framework". In: *IGARSS 2004. 2004 IEEE International Geoscience and Remote Sensing Symposium*. Vol. 2. 2004, 949–952 vol.2. doi: 10.1109/IGARSS.2004.1368565 (cit. on p. 39).

[Che17] Chen, Shanzhi; Hu, Jinling; Shi, Yan; Peng, Ying; Fang, Jiayi; Zhao, Rui and Zhao, Li: "Vehicle-to-Everything (v2x) Services Supported by LTE-Based Systems and 5G". In: *IEEE Communications Standards Magazine* (2017). doi: 10.1109/MCOMSTD.2017.1700015 (cit. on p. 60).

[Che18] Chen, Liang-Chieh; Zhu, Yukun; Papandreou, George; Schroff, Florian and Adam, Hartwig: "Encoder-decoder with atrous separable convolution for semantic image segmentation". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 801–818 (cit. on pp. 21, 95).

[Che19] Chen, Minghao; Xue, Hongyang and Cai, Deng: "Domain Adaptation for Semantic Segmentation With Maximum Squares Loss". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 2090–2099. DOI: 10.1109/ICCV.2019.00218 (cit. on p. 38).

[Che20a] Chen, Wuyang; Yu, Zhiding; Wang, Zhangyang and Anandkumar, Animashree: "Automated synthetic-to-real generalization". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 1746–1756 (cit. on pp. 32, 33).

[Che20b] Chen, Xinlei; Fan, Haoqi; Girshick, Ross and He, Kaiming: "Improved Baselines with Momentum Contrastive Learning". In: *arXiv preprint arXiv:2003.04297* (2020) (cit. on pp. 32, 58, 62).

[Che20c] Cheng, Shuyang; Leng, Zhaoqi; Cubuk, Ekin Dogus; Zoph, Barret; Bai, Chunyan; Ngiam, Jiquan; Song, Yang; Caine, Benjamin; Vasudevan, Vijay; Li, Congcong et al.: "Improving 3d object detection through progressive population based augmentation". In: *European Conference on Computer Vision (ECCV)*. Springer. 2020, pp. 279–294 (cit. on p. 68).

[Che21] Chen, Xiaokang; Yuan, Yuhui; Zeng, Gang and Wang, Jingdong: "Semi-supervised semantic segmentation with cross pseudo supervision". In: *CVPR*. 2021 (cit. on pp. 58, 62).

[Che22] Cheng, Bowen; Misra, Ishan; Schwing, Alexander G; Kirillov, Alexander and Girdhar, Rohit: "Masked-attention mask transformer for universal image segmentation". In: *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*. 2022, pp. 1290–1299 (cit. on p. 3).

[Che24] Chen, Long; Sinavski, Oleg; Hünermann, Jan; Karnsund, Alice; Willmott, Andrew James; Birch, Danny; Maund, Daniel and Shotton, Jamie: "Driving with LLMs: Fusing object-level vector modality for explainable autonomous driving". In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2024, pp. 14093–14100 (cit. on p. 159).

[Cho18] Chou, Glen; Sahin, Yunus Emre; Yang, Liren; Rutledge, Kwesi J; Nilsson, Petter and Ozay, Necmiye: "Using control synthesis to generate corner cases: A case study on autonomous driving". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 37.11 (2018), pp. 2906–2917 (cit. on p. 30).

[Cho21] Choi, Sungha; Jung, Sanghun; Yun, Huiwon; Kim, Joanne T; Kim, Seungryong and Choo, Jaegul: "Robustnet: Improving domain generalization in urban-scene segmentation via instance selective whitening". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 11580–11590 (cit. on pp. 32, 33, 101).

[Cor16] Cordts, Marius; Omran, Mohamed; Ramos, Sebastian; Rehfeld, Timo; Enzweiler, Markus; Benenson, Rodrigo; Franke, Uwe; Roth, Stefan and Schiele, Bernt: "The cityscapes dataset for semantic urban scene understanding". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2016, pp. 3213–3223 (cit. on pp. 5, 23, 44, 48, 228).

[Cos22] Cossu, Andrea; Graffieti, Gabriele; Pellegrini, Lorenzo; Maltoni, Davide; Bacciu, Davide; Carta, Antonio and Lomonaco, Vincenzo: "Is Class-Incremental Enough for Continual Learning?" In: *Frontiers in Artificial Intelligence* 5 (2022) (cit. on p. 67).

[Cub18] Cubuk, Ekin D; Zoph, Barret; Mane, Dandelion; Vasudevan, Vijay and Le, Quoc V: "Autoaugment: Learning augmentation policies from data". In: *arXiv preprint arXiv:1805.09501* (2018) (cit. on p. 68).

[Cui22] Cui, Yutao; Jiang, Cheng; Wang, Limin and Wu, Gangshan: "MixFormer: End-to-End Tracking with Iterative Mixed Attention". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 13608–13618 (cit. on p. 3).

[Dai15] Dai, Jifeng; He, Kaiming and Sun, Jian: "Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 1635–1643 (cit. on pp. 37, 38).

[dAs21] d'Ascoli, Stéphane; Touvron, Hugo; Leavitt, Matthew L; Morcos, Ari S; Biroli, Giulio and Sagun, Levent: "Convit: Improving vision transformers with soft convolutional inductive biases". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 2286–2296 (cit. on p. 18).

[Das23] Das, Anurag; Xian, Yongqin; Dai, Dengxin and Schiele, Bernt: "Weakly-Supervised Domain Adaptive Semantic Segmentation With Prototypical Contrastive Learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2023, pp. 15434–15443 (cit. on p. 38).

[Den09] Deng, Jia; Dong, Wei; Socher, Richard; Li, Li-Jia; Li, Kai and Fei-Fei, Li: "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on Computer Vision and Pattern Recognition*. Ieee. 2009, pp. 248–255. doi: 10.1109/CVPR.2009.5206848 (cit. on p. 62).

[Den10] Deng, Jia; Berg, Alexander C; Li, Kai and Fei-Fei, Li: "What does classifying more than 10,000 image categories tell us?" In: *European Conference on Computer Vision*. Springer. 2010, pp. 71–84 (cit. on p. 39).

[Dev19] Devlin, Jacob; Chang, Ming-Wei; Lee, Kenton and Toutanova, Kristina: "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. Ed. by Burstein, Jill; Doran, Christy

and Solorio, Thamar. Association for Computational Linguistics, 2019, pp. 4171–4186. DOI: 10.18653/V1/N19-1423. URL: https://doi.org/10.18653/v1/n19-1423 (cit. on p. 17).

[Dha20] Dhamija, Akshay; Gunther, Manuel; Ventura, Jonathan and Boult, Terrance: "The overlooked elephant of object detection: Open set". In: *Proceedings of the WACV*. 2020, pp. 1021–1030 (cit. on pp. 34, 35, 75, 77, 83).

[Din19] Ding, Gavin Weiguang; Lui, Kry Yik Chau; Jin, Xiaomeng; Wang, Luyu and Huang, Ruitong: "On the Sensitivity of Adversarial Robustness to Input Data Distributions." In: *ICLR (poster)* 4 (2019) (cit. on p. 33).

[Dos17] Dosovitskiy, Alexey; Ros, German; Codevilla, Felipe; Lopez, Antonio and Koltun, Vladlen: "CARLA: An Open Urban Driving Simulator". In: *Proceedings of the 1st Annual Conference on Robot Learning*. PMLR. 2017, pp. 1–16 (cit. on p. 72).

[Dos21] Dosovitskiy, Alexey et al.: "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=YicbFdNTTy (cit. on p. 17).

[Du22a] Du, Xuefeng; Gozum, Gabriel; Ming, Yifei and Li, Yixuan: "SIREN: Shaping Representations for Detecting Out-of-Distribution Objects". In: *Proceedings of NeuRIPS*. 2022, pp. 20434–20449 (cit. on p. 34).

[Du22b] Du, Xuefeng; Wang, Zhaoning; Cai, Mu and Li, Yixuan: "Vos: Learning what you don't know by virtual outlier synthesis". In: *Proceedings of International Conference on Learning Representations* (2022) (cit. on p. 34).

[Duc11] Duchi, John; Hazan, Elad and Singer, Yoram: "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization". In: *Journal of Machine Learning Research* 12.61 (2011), pp. 2121–2159. URL: http://jmlr.org/papers/v12/duchi11a.html (cit. on p. 13).

[Dud72] Duda, Richard O. and Hart, Peter E.: "Use of the Hough transformation to detect lines and curves in pictures". In: *Commun. ACM* 15.1 (Jan. 1972), pp. 11–15. doi: 10.1145/361237.361242. url: https://doi.org/10.1145/361237.361242 (cit. on p. 3).

[Ebe24] Eberhart, Bernd: Weitblick. Aug. 2024. url: https://www.porsche.com/germany/aboutporsche/christophorusmagazine/archive/380/articleoverview/article07/ (visited on 08/22/2024) (cit. on p. 1).

[Eur21] European Commission: Regulation of the European Parliament and of the Council Laying Down Harmonised Rules on Artificial Intelligence (Artificial Intelligence Act) and Amending Certain Union Legislative Acts. Apr. 2021 (cit. on pp. 4, 59).

[Eve12] Everingham, M.; Van Gool, L.; Williams, C. K. I.; Winn, J. and Zisserman, A.: The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. 2012 (cit. on p. 41).

[Fan22a] Fang, Alex; Ilharco, Gabriel; Wortsman, Mitchell; Wan, Yuhao; Shankar, Vaishaal; Dave, Achal and Schmidt, Ludwig: "Data determines distributional robustness in contrastive language image pre-training (clip)". In: *International Conference on Machine Learning*. PMLR. 2022, pp. 6216–6234 (cit. on p. 160).

[Fan22b] Fang, Zhen; Li, Yixuan; Lu, Jie; Dong, Jiahua; Han, Bo and Liu, Feng: "Is Out-of-Distribution Detection Learnable?" In: *ECCV* (2022) (cit. on pp. 58, 66).

[Fin17] Finn, Chelsea; Abbeel, Pieter and Levine, Sergey: "Model-agnostic meta-learning for fast adaptation of deep networks". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 1126–1135 (cit. on p. 40).

[Fis22] Fisch, Adam; Jaakkola, Tommi and Barzilay, Regina: "Calibrated selective classification". In: *Transactions on Machine Learning Research* (2022) (cit. on p. 65).

[Fra13] Franke, Uwe; Pfeiffer, David; Rabe, Clemens; Knoeppel, Carsten; Enzweiler, Markus; Stein, Fridtjof and Herrtwich, Ralf G.: "Making Bertha See". In: *2013 IEEE International Conference on Computer Vision Workshops*. 2013, pp. 214–221. doi: 10.1109/ICCVW.2013.36 (cit. on p. 3).

[Gad24] Gadre, Samir Yitzhak; Ilharco, Gabriel; Fang, Alex; Hayase, Jonathan; Smyrnis, Georgios; Nguyen, Thao; Marten, Ryan; Wortsman, Mitchell; Ghosh, Dhruba; Zhang, Jieyu et al.: "Datacomp: In search of the next generation of multimodal datasets". In: *Advances in Neural Information Processing Systems* 36 (2024) (cit. on pp. 4, 160).

[Gao18] Gao, Mingfei; Li, Ang; Yu, Ruichi; Morariu, Vlad I and Davis, Larry S: "C-wsl: Count-guided weakly supervised localization". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 152–168 (cit. on pp. 37, 38).

[Gar20] Garnett, Noa; Uziel, Roy; Efrat, Netalee and Levi, Dan: "Synthetic-to-real domain adaptation for lane detection". In: *Proceedings of the Asian Conference on Computer Vision*. 2020 (cit. on p. 39).

[Gei18] Geirhos, Robert; Temme, Carlos RM; Rauber, Jonas; Schütt, Heiko H; Bethge, Matthias and Wichmann, Felix A: "Generalisation in humans and deep neural networks". In: *NeurIPS* 31 (2018) (cit. on p. 31).

[Gei19] Geirhos, Robert; Rubisch, Patricia; Michaelis, Claudio; Bethge, Matthias; Wichmann, Felix A. and Brendel, Wieland: "ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness". In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. url: https://openreview.net/forum?id=Bygh9j09KX (cit. on p. 17).

[Gey20] Geyer, Jakob; Kassahun, Yohannes; Mahmudi, Mentar; Ricou, Xavier; Durgesh, Rupesh; Chung, Andrew S; Hauswald,

Lorenz; Pham, Viet Hoang; Mühlegg, Maximilian; Dorn, Sebastian et al.: "A2d2: Audi autonomous driving dataset". In: *arXiv preprint arXiv:2004.06320* (2020) (cit. on pp. 24, 33, 51, 60).

[Ghi15] Ghifary, Muhammad; Kleijn, W Bastiaan; Zhang, Mengjie and Balduzzi, David: "Domain generalization for object recognition with multi-task autoencoders". In: *ICCV*. 2015, pp. 2551–2559 (cit. on pp. 32, 58, 67).

[Giu20] Giunchiglia, Eleonora and Lukasiewicz, Thomas: "Coherent hierarchical multi-label classification networks". In: *NeurIPS* (2020) (cit. on p. 39).

[Goo16] Goodfellow, Ian J.; Bengio, Yoshua and Courville, Aaron: Deep Learning. http://www.deeplearningbook.org. Cambridge, MA, USA: MIT Press, 2016 (cit. on p. 16).

[Gra22] Graaff, Thies de and Ribeiro de Menezes, Arthur: "Capsule Networks for Hierarchical Novelty Detection in Object Classification". In: *2022 IEEE Intelligent Vehicles Symposium (IV)*. 2022, pp. 1795–1800. doi: 10.1109/IV51971.2022.9827249 (cit. on p. 40).

[Gud20] Gudovskiy, Denis; Hodgkinson, Alec; Yamaguchi, Takuya and Tsukizawa, Sotaro: "Deep active learning for biased datasets via fisher kernel self-supervision". In: *CVPR*. 2020 (cit. on p. 67).

[Guo22] Guo, Meng-Hao; Lu, Cheng-Ze; Hou, Qibin; Liu, Zhengning; Cheng, Ming-Ming and Hu, Shi-Min: "Segnext: Rethinking convolutional attention design for semantic segmentation". In: *arXiv preprint arXiv:2209.08575* (2022) (cit. on pp. 3, 18, 95).

[Guo25] Guo, Daya; Yang, Dejian; Zhang, Haowei; Song, Junxiao; Zhang, Ruoyu; Xu, Runxin; Zhu, Qihao; Ma, Shirong; Wang, Peiyi; Bi, Xiao et al.: "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning". In: *arXiv preprint arXiv:2501.12948* (2025) (cit. on p. 159).

[Gup22] Gupta, Akshita; Narayan, Sanath; Joseph, KJ; Khan, Salman; Khan, Fahad Shahbaz and Shah, Mubarak: "Ow-detr: Open-world detection transformer". In: *Proceedings of the CVPR*. 2022, pp. 9235–9244 (cit. on pp. 34, 35, 58, 63, 79).

[Hah22] Hahner, Martin; Sakaridis, Christos; Bijelic, Mario; Heide, Felix; Yu, Fisher; Dai, Dengxin and Van Gool, Luc: "Lidar snowfall simulation for robust 3d object detection". In: *CVPR*. 2022 (cit. on pp. 32, 60, 64).

[Haj23] Hajimiri, Sina; Boudiaf, Malik; Ben Ayed, Ismail and Dolz, Jose: "A Strong Baseline for Generalized Few-Shot Semantic Segmentation". In: *Proceedings of the CVPR*. 2023, pp. 11269–11278 (cit. on p. 41).

[Has16] Hasan, Mahmudul; Choi, Jonghyun; Neumann, Jan; Roy-Chowdhury, Amit K and Davis, Larry S: "Learning temporal regularity in video sequences". In: *CVPR*. 2016 (cit. on pp. 58, 65).

[Has21] Hasterok, Constanze; Stompe, Janina; Pfrommer, Julius; Usländer, Thomas; Ziehn, Jens; Reiter, Sebastian; Weber, Michael and Riedel, Till: PAISE – Process Model for AI Systems Engineering. 2021. doi: 10.1515/auto-2022-0020 (cit. on p. 4).

[Has23] Hassani, Ali; Walton, Steven; Li, Jiachen; Li, Shen and Shi, Humphrey: "Neighborhood attention transformer". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 6185–6194 (cit. on p. 18).

[He15] He, Kaiming; Zhang, Xiangyu; Ren, Shaoqing and Sun, Jian: "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 1026–1034 (cit. on p. 3).

[He16] He, Kaiming; Zhang, Xiangyu; Ren, Shaoqing and Sun, Jian: "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2016, pp. 770–778. arXiv: 1512.03385 [cs.CV] (cit. on p. 17).

[He21] He, Xin; Zhao, Kaiyong and Chu, Xiaowen: "AutoML: A survey of the state-of-the-art". In: *Knowledge-Based Systems* 212 (2021), p. 106622 (cit. on pp. 58, 62).

[Hei21] Heidecker, Florian; Breitenstein, Jasmin; Rösch, Kevin; Löhdefink, Jonas; Bieshaar, Maarten; Stiller, Christoph; Fingscheidt, Tim and Sick, Bernhard: "An Application-Driven Conceptualization of Corner Cases for Perception in Highly Automated Driving". In: *IV* (2021) (cit. on pp. 4, 30, 58, 59).

[Hen17] Hendrycks, Dan and Gimpel, Kevin: "A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks". In: *International Conference on Learning Representations*. 2017 (cit. on pp. 58, 65).

[Hen19] Hendrycks, Dan and Dietterich, Thomas: "Benchmarking Neural Network Robustness to Common Corruptions and Perturbations". In: *Proceedings of the International Conference on Learning Representations* (2019) (cit. on p. 32).

[Hen21] Hendrycks, Dan; Zhao, Kevin; Basart, Steven; Steinhardt, Jacob and Song, Dawn: "Natural adversarial examples". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 15262–15271 (cit. on p. 32).

[Hon24] Honda, Hiroto and Uchida, Yusuke: "CLRerNet: improving confidence of lane detection with LaneIoU". In: *Proc. of the WACV*. 2024 (cit. on pp. 3, 23, 131, 134, 137, 141).

[Hos22] Hosoya, Yusuke; Suganuma, Masanori and Okatani, Takayuki: Rectifying Open-set Object Detection: A Taxonomy, Practical Applications, and Proper Evaluation. 2022. doi: 10.48550/ARXIV.2207.09775. url: https://arxiv.org/abs/2207.09775 (cit. on pp. 34, 35, 75, 79).

[How17] Howard, Andrew G; Zhu, Menglong; Chen, Bo; Kalenichenko, Dmitry; Wang, Weijun; Weyand, Tobias; Andreetto, Marco and Adam, Hartwig: "Mobilenets: Efficient convolutional neural networks for mobile vision applications". In: *arXiv preprint arXiv:1704.04861* (2017) (cit. on pp. 3, 20).

[How19] Howard, Andrew; Sandler, Mark; Chu, Grace; Chen, Liang-Chieh; Chen, Bo; Tan, Mingxing; Wang, Weijun; Zhu, Yukun; Pang, Ruoming; Vasudevan, Vijay et al.: "Searching for mobilenetv3". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 1314–1324 (cit. on p. 20).

[Hoy22] Hoyer, Lukas; Dai, Dengxin and Van Gool, Luc: "DAFormer: Improving Network Architectures and Training Strategies for Domain-Adaptive Semantic Segmentation". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 9914–9925. DOI: 10.1109/CVPR52688.2022.00969 (cit. on pp. 132, 150).

[Hoy23] Hoyer, Lukas; Dai, Dengxin; Wang, Haoran and Van Gool, Luc: "MIC: Masked image consistency for context-enhanced domain adaptation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 11721–11732 (cit. on p. 38).

[Hu22] Hu, Chuqing; Hudson, Sinclair; Ethier, Martin; Al-Sharman, Mohammad; Rayside, Derek and Melek, William: "Sim-to-Real Domain Adaptation for Lane Detection and Classification in Autonomous Driving". In: *2022 IEEE Intelligent Vehicles Symposium (IV)*. 2022, pp. 457–463. DOI: 10.1109/IV51971.2022.9827450 (cit. on p. 39).

[Hu23a] Hu, Anthony; Russell, Lloyd; Yeo, Hudson; Murez, Zak; Fedoseev, George; Kendall, Alex; Shotton, Jamie and Corrado, Gianluca: "Gaia-1: A generative world model for autonomous driving". In: *arXiv preprint arXiv:2309.17080* (2023) (cit. on p. 159).

[Hu23b] Hu, Yihan; Yang, Jiazhi; Chen, Li; Li, Keyu; Sima, Chonghao; Zhu, Xizhou; Chai, Siqi; Du, Senyao; Lin, Tianwei; Wang, Wenhai et al.: "Planning-oriented autonomous driving". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 17853–17862 (cit. on p. 3).

[Hu23c] Hu, Zhengdong; Sun, Yifan and Yang, Yi: "Suppressing the Heterogeneity: A Strong Feature Extractor for Few-shot Segmentation". In: *Proceedings of the ICLR*. 2023 (cit. on pp. 41, 115).

[Hua17] Huang, Gao; Liu, Zhuang; Van Der Maaten, Laurens and Weinberger, Kilian Q: "Densely connected convolutional networks". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2017, pp. 4700–4708 (cit. on p. 21).

[Hua21a] Huang, Jiaxing; Guan, Dayan; Xiao, Aoran and Lu, Shijian: "Fsdr: Frequency space domain randomization for domain generalization". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 6891–6902 (cit. on pp. 32, 33).

[Hua21b] Huang, Rui; Geng, Andrew and Li, Yixuan: "On the importance of gradients for detecting distributional shifts in the wild". In: *NeurIPS* (2021) (cit. on pp. 58, 65).

[Huk19] Hukkelås, Håkon; Mester, Rudolf and Lindseth, Frank: "Deepprivacy: A generative adversarial network for face anonymization". In: *International Symposium on Visual Computing*. Springer. 2019, pp. 565–578 (cit. on pp. 45, 47).

[Ino18] Inoue, Naoto; Furuta, Ryosuke; Yamasaki, Toshihiko and Aizawa, Kiyoharu: "Cross-Domain Weakly-Supervised Object Detection Through Progressive Domain Adaptation". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 5001–5009. DOI: 10.1109/CVPR.2018.00525 (cit. on p. 38).

[Iqb22] Iqbal, Ehtesham; Safarov, Sirojbek and Bang, Seongdeok: MSANet: Multi-Similarity and Attention Guidance for Boosting Few-Shot Segmentation. 2022. arXiv: 2206.09667 [cs.CV] (cit. on pp. 41, 125).

[ISO18] ISO: ISO 26262: Road vehicles—Functional Safety. Norm. 2018 (cit. on pp. 4, 160).

[ISO22] ISO: ISO 21448: R.V.—Safety of the Intended Functionality. Norm. 2022 (cit. on pp. 4, 160).

[ISO24] ISO: ISO 8800: Road vehicles — Safety and artificial intelligence. Norm. 2024 (cit. on pp. 3, 160).

[Jae23] Jaeger, Bernhard; Chitta, Kashyap and Geiger, Andreas: "Hidden biases of end-to-end driving models". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 8240–8249 (cit. on p. 3).

[Jai23] Jain, Jitesh; Li, Jiachen; Chiu, Mang Tik; Hassani, Ali; Orlov, Nikita and Shi, Humphrey: "Oneformer: One transformer to rule universal image segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 2989–2998 (cit. on p. 3).

[Ji23] Ji, Ge-Peng; Fan, Deng-Ping; Xu, Peng; Cheng, Ming-Ming; Zhou, Bowen and Van Gool, Luc: "SAM Struggles in Concealed Scenes–Empirical Study on Segment Anything". In: *arXiv preprint arXiv:2304.06022* (2023) (cit. on p. 159).

[Jia22] Jiang, Chiyu Max; Najibi, Mahyar; Qi, Charles R; Zhou, Yin and Anguelov, Dragomir: "Improving the Intra-class Long-Tail in 3D Detection via Rare Example Mining". In: *ECCV*. Springer. 2022 (cit. on pp. 65, 67).

[Joo17] Joon Oh, Seong; Benenson, Rodrigo; Khoreva, Anna; Akata, Zeynep; Fritz, Mario and Schiele, Bernt: "Exploiting saliency for object segmentation from image level labels". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2017, pp. 4410–4419 (cit. on pp. 37, 38).

[Jos21] Joseph, KJ; Khan, Salman; Khan, Fahad Shahbaz and Balasubramanian, Vineeth N: "Towards open world object detection". In: *Proceedings of the CVPR*. 2021, pp. 5830–5840 (cit. on pp. 34, 35).

[Kam20] Kamann, Christoph and Rother, Carsten: "Benchmarking the robustness of semantic segmentation models". In: *CVPR*. 2020 (cit. on p. 32).

[Kan19a] Kang, Bingyi; Liu, Zhuang; Wang, Xin; Yu, Fisher; Feng, Jiashi and Darrell, Trevor: "Few-shot object detection via feature reweighting". In: *Proceedings of the ICCV*. 2019, pp. 8420–8429 (cit. on p. 40).

[Kan19b] Kang, Guoliang; Jiang, Lu; Yang, Yi and Hauptmann, Alexander G: "Contrastive adaptation network for unsupervised domain adaptation". In: *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*. 2019, pp. 4893–4902 (cit. on p. 38).

[Kan22] Kang, Dahyun and Cho, Minsu: "Integrative few-shot learning for classification and segmentation". In: *Proceedings of the CVPR*. 2022, pp. 9979–9990 (cit. on p. 125).

[Kar19] Karras, Tero; Laine, Samuli and Aila, Timo: "A style-based generator architecture for generative adversarial networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 4401–4410 (cit. on p. 45).

[Kar24] Karnchanachari, Napat; Geromichalos, Dimitris; Tan, Kok Seang; Li, Nanxiang; Eriksen, Christopher; Yaghoubi, Shakiba; Mehdipour, Noushin; Bernasconi, Gianmarco; Fong, Whye Kit; Guo, Yiluan et al.: "Towards learning-based planning: The nuPlan benchmark for real-world autonomous driving". In: *arXiv preprint arXiv:2403.04133* (2024) (cit. on p. 3).

[Ken17] Kendall, Alex and Gal, Yarin: "What uncertainties do we need in bayesian deep learning for computer vision?" In: *NeurIPS* (2017) (cit. on pp. 58, 65).

[Kho17] Khoreva, Anna; Benenson, Rodrigo; Hosang, Jan; Hein, Matthias and Schiele, Bernt: "Simple does it: Weakly supervised instance and semantic segmentation". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2017, pp. 876–885 (cit. on p. 38).

[Kim22] Kim, Dahun; Lin, Tsung-Yi; Angelova, Anelia; Kweon, In So and Kuo, Weicheng: "Learning Open-World Object Proposals without Learning to Classify". In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 5453–5460 (cit. on p. 33).

[Kin15] Kingma, Diederik P. and Ba, Jimmy: "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Bengio, Yoshua and LeCun, Yann. 2015. URL: http://arxiv.org/abs/1412.6980 (cit. on p. 13).

[Kir17] Kirkpatrick, James; Pascanu, Razvan; Rabinowitz, Neil; Veness, Joel; Desjardins, Guillaume; Rusu, Andrei A; Milan, Kieran; Quan, John; Ramalho, Tiago; Grabska-Barwinska, Agnieszka et al.: "Overcoming catastrophic forgetting in neural networks". In: *Proceedings of the national academy of sciences* (2017) (cit. on p. 67).

[Kir23] Kirillov, Alexander et al.: "Segment Anything". In: *arXiv:2304.02643* (2023) (cit. on pp. 33, 159).

[Kis21] Kishore, Aman; Choe, Tae Eun; Kwon, Junghyun; Park, Minwoo; Hao, Pengfei and Mittel, Akshita: "Synthetic data generation using imitation training". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 3078–3086 (cit. on p. 30).

[Koe23] Koenig, Christophe: Level 3 highly automated driving available in the new BMW 7 Series from next spring. Nov. 2023. URL: https://www.press.bmwgroup.com/global/article/detail/T0438214EN/level-3-highly-automated-driving-available-in-the-new-bmw-7-series-from-next-spring (visited on 08/22/2024) (cit. on p. 1).

[Kok20] Kokhlikyan, Narine; Miglani, Vivek; Martin, Miguel; Wang, Edward; Alsallakh, Bilal; Reynolds, Jonathan; Melnikov, Alexander; Kliushkina, Natalia; Araya, Carlos; Yan, Siqi et al.: "Captum: A unified and generic model interpretability library

for pytorch". In: *arXiv preprint arXiv:2009.07896* (2020) (cit. on p. 53).

[Kot22] Kothawade, Suraj; Roy, Donna; Fenzi, Michele; Haussmann, Elmar; Alvarez, Jose M. and Angerer, Christoph: "Object-Level Targeted Selection via Deep Template Matching". In: *IV*. 2022 (cit. on pp. 58, 63).

[Kri12] Krizhevsky, Alex; Sutskever, Ilya and Hinton, Geoffrey E.: "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*. Ed. by Bartlett, Peter L.; Pereira, Fernando C. N.; Burges, Christopher J. C.; Bottou, Léon and Weinberger, Kilian Q. 2012, pp. 1106–1114. url: https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html (cit. on p. 16).

[Lam20] Lambert, John; Liu, Zhuang; Sener, Ozan; Hays, James and Koltun, Vladlen: "MSeg: A composite dataset for multi-domain semantic segmentation". In: *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*. 2020, pp. 2879–2888 (cit. on p. 33).

[Lan22] Lang, Chunbo; Cheng, Gong; Tu, Binfei and Han, Junwei: "Learning what not to segment: A new perspective on few-shot segmentation". In: *Proceedings of the CVPR*. 2022, pp. 8057–8067 (cit. on pp. 41, 114, 117).

[LeC98] LeCun, Yann; Bottou, Léon; Bengio, Yoshua and Haffner, Patrick: "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324 (cit. on p. 14).

[Lee18a] Lee, Kibok; Lee, Kimin; Min, Kyle; Zhang, Yuting; Shin, Jinwoo and Lee, Honglak: "Hierarchical novelty detection for

visual object recognition". In: *Proceedings of the CVPR*. 2018, pp. 1034–1042 (cit. on pp. 40, 75, 77, 78).

[Lee18b] Lee, Kimin; Lee, Kibok; Lee, Honglak and Shin, Jinwoo: "A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks". In: *NeurIPS*. 2018 (cit. on pp. 58, 65).

[Li17] Li, Zhizhong and Hoiem, Derek: "Learning without forgetting". In: *IEEE transactions on pattern analysis and machine intelligence* (2017) (cit. on p. 67).

[Li18] Li, Yanghao; Wang, Naiyan; Shi, Jianping; Hou, Xiaodi and Liu, Jiaying: "Adaptive batch normalization for practical domain adaptation". In: *Pattern Recognition* 80 (2018), pp. 109–117 (cit. on p. 38).

[Li20a] Li, Xiang; Li, Jun; Hu, Xiaolin and Yang, Jian: "Line-CNN: End-to-End Traffic Line Detection With Line Proposal Unit". In: *IEEE Transactions on Intelligent Transportation Systems* 21 (2020), pp. 248–258. DOI: 10.1109/TITS.2019.2890870 (cit. on p. 23).

[Li20b] Li, Xiang; Wei, Tianhan; Chen, Yau Pun; Tai, Yu-Wing and Tang, Chi-Keung: "Fss-1000: A 1000-class dataset for few-shot segmentation". In: *Proceedings of the CVPR*. 2020, pp. 2869–2878 (cit. on p. 41).

[Li21] Li, Gen; Jampani, Varun; Sevilla-Lara, Laura; Sun, Deqing; Kim, Jonghyun and Kim, Joongkyu: "Adaptive prototype learning and allocation for few-shot segmentation". In: *Proceedings of the CVPR*. 2021, pp. 8334–8343 (cit. on p. 41).

[Li22a] Li, Aodi; Zhuang, Liansheng; Fan, Shuo and Wang, Shafei: "Learning Common and Specific Visual Prompts for Domain Generalization". In: *Proceedings of the Asian Conference on Computer Vision (ACCV)*. Dec. 2022, pp. 4260–4275 (cit. on p. 18).

[Li22b] Li, Chenguang; Zhang, Boheng; Shi, Jia and Cheng, Guangliang: "Multi-Level Domain Adaptation for Lane Detection". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2022, pp. 4379–4388. DOI: 10.1109/CVPRW56347.2022.00484 (cit. on p. 39).

[Li22c] Li, Liulei; Zhou, Tianfei; Wang, Wenguan; Li, Jianwu and Yang, Yi: "Deep hierarchical semantic segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 1246–1257 (cit. on pp. 39, 40, 75, 76).

[Li23] Li, Yifan; Du, Yifan; Zhou, Kun; Wang, Jinpeng; Zhao, Wayne Xin and Wen, Ji-Rong: "Evaluating object hallucination in large vision-language models". In: *arXiv preprint arXiv:2305.10355* (2023) (cit. on p. 5).

[Lia19] Lian, Qing; Duan, Lixin; Lv, Fengmao and Gong, Boqing: "Constructing Self-Motivated Pyramid Curriculums for Cross-Domain Semantic Segmentation: A Non-Adversarial Approach". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 6757–6766. DOI: 10.1109/ICCV.2019.00686 (cit. on p. 38).

[Lia20] Liang, Junwei; Jiang, Lu and Hauptmann, Alexander: "Simaug: Learning robust representations from simulation for trajectory prediction". In: *ECCV*. Springer. 2020 (cit. on p. 68).

[Lin16] Lin, Di; Dai, Jifeng; Jia, Jiaya; He, Kaiming and Sun, Jian: "Scribblesup: Scribble-supervised convolutional networks for semantic segmentation". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2016, pp. 3159–3167 (cit. on pp. 37, 38).

[Lis19] Lis, Krzysztof; Nakka, Krishna; Fua, Pascal and Salzmann, Mathieu: "Detecting the unexpected via image resynthesis". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 2152–2161 (cit. on pp. 58, 65).

[Liu16] Liu, Wei; Anguelov, Dragomir; Erhan, Dumitru; Szegedy, Christian; Reed, Scott; Fu, Cheng-Yang and Berg, Alexander C: "Ssd: Single shot multibox detector". In: *European Conference on Computer Vision*. Springer. 2016, pp. 21–37 (cit. on pp. 3, 20).

[Liu20a] Liu, Tong; Chen, Zhaowei; Yang, Yi; Wu, Zehao and Li, Haowei: "Lane detection in low-light conditions using an efficient data enhancement: Light conditions style transfer". In: *2020 IEEE intelligent vehicles symposium (IV)*. IEEE. 2020, pp. 1394–1399 (cit. on p. 22).

[Liu20b] Liu, Weitang; Wang, Xiaoyun; Owens, John and Li, Yixuan: "Energy-based out-of-distribution detection". In: *NeurIPS* (2020) (cit. on pp. 58, 65).

[Liu21a] Liu, Lizhe; Chen, Xiaohao; Zhu, Siyu and Tan, Ping: "CondLaneNet: A Top-to-down Lane Detection Framework Based on Conditional Convolution". In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 3753–3762. doi: 10.1109/ICCV48922.2021.00375 (cit. on pp. 3, 23, 131).

[Liu21b] Liu, Ze; Lin, Yutong; Cao, Yue; Hu, Han; Wei, Yixuan; Zhang, Zheng; Lin, Stephen and Guo, Baining: "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 10012–10022 (cit. on pp. 17, 18, 22, 95).

[Liu21c] Liu, Zhijian; Amini, Alexander; Zhu, Sibo; Karaman, Sertac; Han, Song and Rus, Daniela L: "Efficient and robust lidar-based end-to-end navigation". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 13247–13254 (cit. on p. 32).

[Liu22a] Liu, Xiaofeng; Yoo, Chaehwa; Xing, Fangxu; Oh, Hyejin; El Fakhri, Georges; Kang, Je-Won; Woo, Jonghye et al.: "Deep unsupervised domain adaptation: A review of recent advances and perspectives". In: *APSIPA Transactions on Signal and Information Processing* 11.1 (2022) (cit. on p. 38).

[Liu22b] Liu, Zhuang; Mao, Hanzi; Wu, Chao-Yuan; Feichtenhofer, Christoph; Darrell, Trevor and Xie, Saining: "A ConvNet for the 2020s". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 11966–11976. DOI: 10.1109/CVPR52688.2022.01167 (cit. on pp. 18, 95).

[Liu23] Liu, Sun-Ao; Zhang, Yiheng; Qiu, Zhaofan; Xie, Hongtao; Zhang, Yongdong and Yao, Ting: "Learning Orthogonal Prototypes for Generalized Few-Shot Semantic Segmentation". In: *Proceedings of the CVPR*. 2023, pp. 11319–11328. DOI: 10.1109/CVPR52729.2023.01089 (cit. on p. 41).

[Liu24] Liu, Hanchao; Xue, Wenyuan; Chen, Yifei; Chen, Dapeng; Zhao, Xiutian; Wang, Ke; Hou, Liping; Li, Rongjun and Peng, Wei: "A survey on hallucination in large vision-language models". In: *arXiv preprint arXiv:2402.00253* (2024) (cit. on p. 5).

[Lon15] Long, Jonathan; Shelhamer, Evan and Darrell, Trevor: "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2015, pp. 3431–3440. arXiv: 1411.4038 [cs.CV] (cit. on pp. 21, 46, 95).

[Los19] Loshchilov, Ilya and Hutter, Frank: Decoupled Weight Decay Regularization. 2019. arXiv: 1711.05101 [cs.LG] (cit. on p. 13).

[Low04] Lowe, David G: "Distinctive image features from scale-invariant keypoints". In: *International journal of computer vision* 60 (2004), pp. 91–110 (cit. on p. 22).

[Lv21] Lv, Fengmao; Lin, Guosheng; Liu, Peng; Yang, Guowu; Pan, Sinno Jialin and Duan, Lixin: "Weakly-Supervised Cross-Domain Road Scene Segmentation via Multi-Level Curriculum Adaptation". In: *IEEE Transactions on Circuits and Systems for Video Technology* 31 (2021), pp. 3493–3503. DOI: 10.1109/TCSVT.2020.3040343 (cit. on p. 38).

[Ma24] Ma, Jun; He, Yuting; Li, Feifei; Han, Lin; You, Chenyu and Wang, Bo: "Segment anything in medical images". In: *Nature communications* 15.1 (2024), p. 654 (cit. on p. 159).

[Mai24] Maini, Pratyush; Goyal, Sachin; Lipton, Zachary Chase; Kolter, J. Zico and Raghunathan, Aditi: "T-MARS: Improving Visual Representations by Circumventing Text Feature Learning". In: *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL: https://openreview.net/forum?id=ViPtjIVzUw (cit. on p. 160).

[Man47] Mann, Henry B and Whitney, Donald R: "On a test of whether one of two random variables is stochastically larger than the other". In: *The annals of mathematical statistics* (1947), pp. 50–60 (cit. on p. 48).

[Mar17] Maria Carlucci, Fabio; Porzi, Lorenzo; Caputo, Barbara; Ricci, Elisa and Rota Bulo, Samuel: "Autodial: Automatic domain alignment layers". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 5067–5075 (cit. on p. 38).

[Mar24] Marcu, Ana-Maria; Chen, Long; Hünermann, Jan; Karnsund, Alice; Hanotte, Benoit; Chidananda, Prajwal; Nair, Saurabh; Badrinarayanan, Vijay; Kendall, Alex; Shotton, Jamie et al.: "LingoQA: Visual question answering for autonomous driving". In: *European Conference on Computer Vision*. Springer. 2024, pp. 252–269 (cit. on p. 159).

[McC43] McCulloch, Warren S and Pitts, Walter: "A logical calculus of the ideas immanent in nervous activity". In: *The bulletin of mathematical biophysics* 5 (1943), pp. 115–133 (cit. on p. 11).

[Mer21] Mercedes-Benz AG: First internationally valid system approval for L3 system. Dec. 2021. URL: https://group.mercedes-benz.com/innovation/product-innovation/autonomous-driving/system-approval-for-conditionally-automated-driving.html (visited on 08/22/2024) (cit. on p. 1).

[Mil18] Miller, Dimity; Nicholson, Lachlan; Dayoub, Feras and Sünderhauf, Niko: "Dropout sampling for robust object detection in open-set conditions". In: *2018 IEEE International Conference*

on Robotics and Automation (ICRA). 2018, pp. 3243–3249 (cit. on p. 35).

[Mil21] Miller, John P; Taori, Rohan; Raghunathan, Aditi; Sagawa, Shiori; Koh, Pang Wei; Shankar, Vaishaal; Liang, Percy; Carmon, Yair and Schmidt, Ludwig: "Accuracy on the line: on the strong correlation between out-of-distribution and in-distribution generalization". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 7721–7735 (cit. on p. 159).

[Min21] Min, Juhong; Kang, Dahyun and Cho, Minsu: "Hypercorrelation squeeze for few-shot segmentation". In: *Proceedings of the ICCV*. 2021, pp. 6941–6952 (cit. on pp. 41, 125).

[MMS20] MMSegmentation Contributors: MMSegmentation: OpenMMLab Semantic Segmentation Toolbox and Benchmark. https://github.com/open-mmlab/mmsegmentation. 2020 (cit. on p. 95).

[Mob24] Mobileye: Mobileye Chauffeur - Bringing an eyes-off driving platform to consumer vehicles. Aug. 2024. url: https://www.mobileye.com/solutions/chauffeur/ (visited on 08/22/2024) (cit. on p. 2).

[Möl21] Möller, Felix; Botache, Diego; Huseljic, Denis; Heidecker, Florian; Bieshaar, Maarten and Sick, Bernhard: "Out-of-distribution Detection and Generation using Soft Brownian Offset Sampling and Autoencoders". In: *CVPR Workshops* (2021) (cit. on p. 30).

[Mu14] Mu, Chunyang and Ma, Xing: "Lane detection based on object segmentation and piecewise fitting". In: *TELKOMNIKA Indonesian Journal of Electrical Engineering* 12.5 (2014), pp. 3491–3500 (cit. on p. 22).

[Mye22] Myers-Dean, Josh; Zhao, Yinan; Price, Brian; Cohen, Scott and Gurari, Danna: Generalized Few-Shot Semantic Segmentation: All You Need is Fine-Tuning. 2022. arXiv: 2112.10982 [cs.CV] (cit. on p. 41).

[Naj19] Najibi, Mahyar; Ji, Jingwei; Zhou, Yin; Qi, Charles R; Yan, Xinchen; Ettinger, Scott and Anguelov, Dragomir: "Motion Inspired Unsupervised Perception and Prediction in Autonomous Driving". In: *ECCV*. Springer. 2019 (cit. on pp. 58, 63, 65).

[Nal19] Nalisnick, Eric; Matsukawa, Akihiro; Teh, Yee Whye; Gorur, Dilan and Lakshminarayanan, Balaji: "Do deep generative models know what they don't know?" In: *ICLR*. 2019 (cit. on p. 65).

[Nas21] Naseer, Muhammad Muzammal; Ranasinghe, Kanchana; Khan, Salman H; Hayat, Munawar; Shahbaz Khan, Fahad and Yang, Ming-Hsuan: "Intriguing properties of vision transformers". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 23296–23308 (cit. on p. 18).

[Neu17] Neuhold, Gerhard; Ollmann, Tobias; Rota Bulo, Samuel and Kontschieder, Peter: "The mapillary vistas dataset for semantic understanding of street scenes". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 4990–4999 (cit. on pp. 24, 228).

[Nev18] Neven, Davy; De Brabandere, Bert; Georgoulis, Stamatios; Proesmans, Marc and Van Gool, Luc: Towards End-to-End Lane Detection: An Instance Segmentation Approach. 2018 (cit. on p. 22).

[Ngu15] Nguyen, Anh; Yosinski, Jason and Clune, Jeff: "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images". In: *CVPR*. 2015 (cit. on p. 4).

[Niu16] Niu, Jianwei; Lu, Jie; Xu, Mingliang; Lv, Pei and Zhao, Xiaoke: "Robust lane detection using two-stage feature extraction with curve fitting". In: *Pattern Recognition* 59 (2016), pp. 225–233 (cit. on p. 22).

[Nor23] Nordhoff, Sina; Lee, John D; Calvert, Simeon C; Berge, Siri; Hagenzieker, Marjan and Happee, Riender: "(Mis-) use of standard Autopilot and Full Self-Driving (FSD) Beta: Results from interviews with users of Tesla's FSD Beta". In: *Frontiers in psychology* 14 (2023), p. 1101520 (cit. on p. 1).

[Ouy21] Ouyang, Tinghui; Marco, Vicent Sanz; Isobe, Yoshinao; Asoh, Hideki; Oiwa, Yutaka and Seo, Yoshiki: "Corner case data description and detection". In: *2021 IEEE/ACM 1st Workshop on AI Engineering-Software Engineering for AI (WAIN)*. IEEE. 2021, pp. 19–26 (cit. on p. 31).

[Oza19] Oza, Poojan and Patel, Vishal M: "C2ae: Class conditioned auto-encoder for open-set recognition". In: *CVPR*. 2019 (cit. on pp. 58, 65).

[Pan18a] Pan, Xingang; Luo, Ping; Shi, Jianping and Tang, Xiaoou: "Two at once: Enhancing learning and generalization capacities via ibn-net". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 464–479 (cit. on p. 32).

[Pan18b] Pan, Xingang; Shi, Jianping; Luo, Ping; Wang, Xiaogang and Tang, Xiaoou: "Spatial as Deep: Spatial CNN for Traffic Scene Understanding". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32 (2018). doi: 10.1609/aaai.v32i1.12301 (cit. on pp. 22, 24, 144).

[Pan19] Pan, Xingang; Zhan, Xiaohang; Shi, Jianping; Tang, Xiaoou and Luo, Ping: "Switchable whitening for deep representation learning". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 1863–1871 (cit. on p. 32).

[Pau22] Paul, Sayak and Chen, Pin-Yu: "Vision transformers are robust learners". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 2022, pp. 2071–2081 (cit. on pp. 18, 32).

[Pei17] Pei, Kexin; Cao, Yinzhi; Yang, Junfeng and Jana, Suman: "Deepxplore: Automated whitebox testing of deep learning systems". In: *proceedings of the 26th Symposium on Operating Systems Principles*. 2017 (cit. on p. 30).

[Pen21] Peng, Duo; Lei, Yinjie; Liu, Lingqiao; Zhang, Pingping and Liu, Jun: "Global and local texture randomization for synthetic-to-real semantic segmentation". In: *IEEE Transactions on Image Processing* 30 (2021), pp. 6594–6608 (cit. on pp. 32, 33).

[Pfe10] Pfeiffer, David and Franke, Uwe: "Efficient representation of traffic scenes by means of dynamic stixels". In: *2010 IEEE Intelligent Vehicles Symposium*. 2010, pp. 217–224. doi: 10.1109/IVS.2010.5548114 (cit. on p. 3).

[Pfe22] Pfeil, Jerg; Wieland, Jochen; Michalke, Thomas and Theissler, Andreas: "On Why the System Makes the Corner Case: AI-based Holistic Anomaly Detection for Autonomous Driving". In: *IV*. IEEE. 2022 (cit. on pp. 30, 58).

[Pir20] Piratla, Vihari; Netrapalli, Praneeth and Sarawagi, Sunita: "Efficient domain generalization via common-specific low-rank decomposition". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 7728–7738 (cit. on pp. 58, 67).

[Piv23] Piva, Fabrizio J.; Geus, Daan de and Dubbelman, Gijs: "Empirical Generalization Study: Unsupervised Domain Adaptation vs. Domain Generalization Methods for Semantic Segmentation in the Wild". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Jan. 2023, pp. 499–508 (cit. on p. 102).

[Pom96] Pomerleau, D. and Jochem, T.: "Rapidly adapting machine vision for automated vehicle steering". In: *IEEE Expert* 11.2 (1996), pp. 19–27. doi: 10.1109/64.491277 (cit. on p. 1).

[Qin20] Qin, Zequn; Wang, Huanyu and Li, Xi: "Ultra fast structure-aware deep lane detection". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*. Springer. 2020, pp. 276–291 (cit. on pp. 23, 133, 148).

[Rad21] Radford, Alec; Kim, Jong Wook; Hallacy, Chris; Ramesh, Aditya; Goh, Gabriel; Agarwal, Sandhini; Sastry, Girish; Askell, Amanda; Mishkin, Pamela; Clark, Jack et al.: "Learning transferable visual models from natural language supervision". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 8748–8763 (cit. on pp. 5, 32, 58, 62, 159).

[Ram21] Ramesh, Aditya; Pavlov, Mikhail; Goh, Gabriel; Gray, Scott; Voss, Chelsea; Radford, Alec; Chen, Mark and Sutskever, Ilya: "Zero-shot text-to-image generation". In: *International Conference on Machine Learning*. Pmlr. 2021, pp. 8821–8831 (cit. on pp. 5, 159).

[Rav24] Ravi, Nikhila; Gabeur, Valentin; Hu, Yuan-Ting; Hu, Ronghang; Ryali, Chaitanya; Ma, Tengyu; Khedr, Haitham; Rädle, Roman; Rolland, Chloe; Gustafson, Laura et al.: "Sam 2: Segment anything in images and videos". In: *arXiv preprint arXiv:2408.00714* (2024) (cit. on p. 159).

[Red16] Redmon, Joseph; Divvala, Santosh; Girshick, Ross and Farhadi, Ali: "You only look once: Unified, real-time object detection". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2016, pp. 779–788 (cit. on pp. 3, 20).

[Red17] Redmon, Joseph and Farhadi, Ali: "YOLO9000: better, faster, stronger". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2017, pp. 7263–7271 (cit. on pp. 3, 20).

[Red18] Redmon, Joseph and Farhadi, Ali: "Yolov3: An incremental improvement". In: *arXiv preprint arXiv:1804.02767* (2018) (cit. on p. 20).

[Ren15] Ren, Shaoqing; He, Kaiming; Girshick, Ross and Sun, Jian: "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems* 28 (2015), pp. 91–99 (cit. on pp. 3, 20, 78).

[Ren19] Ren, Jie; Liu, Peter J; Fertig, Emily; Snoek, Jasper; Poplin, Ryan; Depristo, Mark; Dillon, Joshua and Lakshminarayanan, Balaji: "Likelihood ratios for out-of-distribution detection". In: *NeurIPS* (2019) (cit. on pp. 58, 65).

[Ren21] Ren, Jie; Fort, Stanislav; Liu, Jeremiah; Roy, Abhijit Guha; Padhy, Shreyas and Lakshminarayanan, Balaji: "A simple fix to mahalanobis distance for improving near-ood detection". In: *arXiv preprint arXiv:2106.09022* (2021) (cit. on pp. 58, 65).

[Ric16] Richter, Stephan R; Vineet, Vibhav; Roth, Stefan and Koltun, Vladlen: "Playing for data: Ground truth from computer games". In: *European Conference on Computer Vision*. Springer. 2016, pp. 102–118 (cit. on p. 33).

[Rie19] Ries, Lennart; Langner, Jacob; Otten, Stefan; Bach, Johannes and Sax, Eric: "A driving scenario representation for scalable real-data analytics with neural networks". In: *IV*. IEEE. 2019, pp. 2215–2222 (cit. on p. 30).

[Rob24a] Robert Bosch GmbH: Automated Valet Parking. Aug. 2024. url: https://www.bosch-mobility.com/en/solutions/parking/automated-valet-parking/ (visited on 08/22/2024) (cit. on p. 2).

[Rob24b] Robobus, Apollo: Apollo Robobus - A L4 autonomous driving bus dedicated to a new experience of green and low-carbon public mobility. Aug. 2024. url: https://en.apollo.auto/robobus (visited on 08/22/2024) (cit. on p. 1).

[Rom22] Rombach, Robin; Blattmann, Andreas; Lorenz, Dominik; Esser, Patrick and Ommer, Björn: "High-resolution image synthesis with latent diffusion models". In: *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*. 2022, pp. 10684–10695 (cit. on pp. 5, 159).

[Ros16] Ros, German; Sellart, Laura; Materzynska, Joanna; Vazquez, David and Lopez, Antonio M: "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2016, pp. 3234–3243 (cit. on p. 33).

[Rus15a] Russakovsky, Olga; Deng, Jia; Su, Hao; Krause, Jonathan; Satheesh, Sanjeev; Ma, Sean; Huang, Zhiheng; Karpathy, Andrej; Khosla, Aditya; Bernstein, Michael et al.: "Imagenet large scale visual recognition challenge". In: *International journal of computer vision* 115 (2015), pp. 211–252 (cit. on pp. 14, 32, 95).

[Rus15b] Russakovsky, Olga et al.: "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision (IJCV)* 115.3 (2015), pp. 211–252. DOI: 10.1007/s11263-015-0816-y (cit. on p. 34).

[Sak21] Sakaridis, Christos; Dai, Dengxin and Van Gool, Luc: "ACDC: The Adverse Conditions Dataset with Correspondences for Semantic Driving Scene Understanding". In: *arXiv preprint arXiv:2104.13395* (2021), pp. 10765–10775 (cit. on p. 23).

[San18] Sandler, Mark; Howard, Andrew; Zhu, Menglong; Zhmoginov, Andrey and Chen, Liang-Chieh: "Mobilenetv2: Inverted residuals and linear bottlenecks". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2018, pp. 4510–4520 (cit. on p. 20).

[Sel17] Selvaraju, Ramprasaath R; Cogswell, Michael; Das, Abhishek; Vedantam, Ramakrishna; Parikh, Devi and Batra, Dhruv: "Grad-cam: Visual explanations from deep networks via gradient-based localization". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 618–626 (cit. on pp. 36, 37).

[Sha17] Shaban, Amirreza; Bansal, Shray; Liu, Zhen; Essa, Irfan and Boots, Byron: "One-Shot Learning for Semantic Segmentation". In: *Proceedings of the BMVC*. 2017 (cit. on p. 40).

[Sha23] Shao, Hao; Wang, Letian; Chen, Ruobing; Waslander, Steven L; Li, Hongsheng and Liu, Yu: "Reasonnet: End-to-end driving with temporal and global reasoning". In: *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*. 2023, pp. 13723–13733 (cit. on p. 3).

[Shi20] Shin, Inkyu; Woo, Sanghyun; Pan, Fei and Kweon, In So: "Two-phase pseudo label densification for self-training based domain adaptation". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*. Springer. 2020, pp. 532–548 (cit. on p. 38).

[Shi22] Shi, Xinyu; Wei, Dong; Zhang, Yu; Lu, Donghuan; Ning, Munan; Chen, Jiashun; Ma, Kai and Zheng, Yefeng: "Dense Cross-Query-and-Support Attention Weighted Mask Aggregation for Few-Shot Segmentation". In: *Proceedings of the ECCV*. Springer. 2022, pp. 151–168 (cit. on p. 41).

[Shi24] Shi, Shaoshuai; Jiang, Li; Dai, Dengxin and Schiele, Bernt: "Mtr++: Multi-agent motion prediction with symmetric scene modeling and guided intention querying". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024) (cit. on p. 3).

[Shr16] Shrivastava, Abhinav; Gupta, Abhinav and Girshick, Ross: "Training region-based object detectors with online hard example mining". In: *CVPR*. 2016 (cit. on pp. 58, 62).

[Shu21] Shu, Yang; Cao, Zhangjie; Wang, Chenyu; Wang, Jianmin and Long, Mingsheng: "Open domain generalization with domain-augmented meta-learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 9624–9633 (cit. on p. 32).

[Sim15] Simonyan, Karen and Zisserman, Andrew: "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Bengio, Yoshua and LeCun, Yann. 2015. URL: http://arxiv.org/abs/1409.1556 (cit. on p. 17).

[Sin18] Singh, Krishna Kumar; Yu, Hao; Sarmasi, Aron; Pradeep, Gautam and Lee, Yong Jae: Hide-and-Seek: A Data Augmentation Technique for Weakly-Supervised Localization and Beyond. 2018 (cit. on p. 37).

[Sit18] Sitawarin, Chawin; Bhagoji, Arjun Nitin; Mosenia, Arsalan; Chiang, Mung and Mittal, Prateek: "Darts: Deceiving autonomous cars with toxic signs". In: *arXiv preprint arXiv:1802.06430* (2018) (cit. on pp. 58, 66).

[Soh20] Sohn, Kihyuk; Berthelot, David; Carlini, Nicholas; Zhang, Zizhao; Zhang, Han; Raffel, Colin A; Cubuk, Ekin Dogus; Kurakin, Alexey and Li, Chun-Liang: "Fixmatch: Simplifying semi-supervised learning with consistency and confidence". In: *Advances in neural information processing systems* 33 (2020), pp. 596–608 (cit. on p. 150).

[Son19] Song, Chunfeng; Huang, Yan; Ouyang, Wanli and Wang, Liang: "Box-Driven Class-Wise Region Masking and Filling Rate Guided Loss for Weakly Supervised Semantic Segmentation". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 3131–3140. doi: 10.1109/CVPR.2019.00325 (cit. on p. 38).

[Su21] Su, Jinming; Chen, Chao; Zhang, Ke; Luo, Junfeng; Wei, Xiaoming and Wei, Xiaolin: Structure Guided Lane Detection. 2021 (cit. on p. 23).

[Sun16] Sun, Baochen; Feng, Jiashi and Saenko, Kate: "Return of frustratingly easy domain adaptation". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 30. 1. 2016 (cit. on p. 38).

[Sun22] Sun, Yanpeng; Chen, Qiang; He, Xiangyu; Wang, Jian; Feng, Haocheng; Han, Junyu; Ding, Errui; Cheng, Jian; Li, Zechao and Wang, Jingdong: "Singular Value Fine-tuning: Few-shot Segmentation requires Few-parameters Fine-tuning". In: *Advances in Neural Information Processing Systems*. 2022 (cit. on p. 41).

[Sze14] Szegedy, Christian; Zaremba, Wojciech; Sutskever, Ilya; Bruna, Joan; Erhan, Dumitru; Goodfellow, Ian and Fergus, Rob: "Intriguing properties of neural networks". In: *Processings of International Conference on Learning Representations (ICLR)*. 2014 (cit. on p. 31).

[Sze16] Szegedy, Christian; Vanhoucke, Vincent; Ioffe, Sergey; Shlens, Jon and Wojna, Zbigniew: "Rethinking the inception architecture for computer vision". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2016, pp. 2818–2826 (cit. on p. 17).

[Tab21] Tabelini, Lucas; Berriel, Rodrigo; Paixao, Thiago M.; Badue, Claudine; De Souza, Alberto F. and Oliveira-Santos, Thiago: "Keep Your Eyes on the Lane: Real-Time Attention-Guided Lane Detection". In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 294–302. doi: 10.1109/CVPR46437.2021.00036 (cit. on pp. 3, 23, 131, 137).

[Tan21] Tang, Jigang; Li, Songbin and Liu, Peng: "A Review of Lane Detection Methods Based on Deep Learning". In: *Pattern Recognition* 111 (2021), p. 107623. doi: 10.1016/j.patcog.2020.107623 (cit. on p. 22).

[Tao20] Taori, Rohan; Dave, Achal; Shankar, Vaishaal; Carlini, Nicholas; Recht, Benjamin and Schmidt, Ludwig: "Measuring robustness to natural distribution shifts in image classification". In: *Advances in Neural Information Processing Systems (NeurIPS)* 33 (2020), pp. 18583–18599 (cit. on p. 31).

[Tar17] Tarvainen, Antti and Valpola, Harri: "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results". In: *Advances in neural information processing systems* 30 (2017) (cit. on p. 133).

[Tea23] Team, Gemini; Anil, Rohan; Borgeaud, Sebastian; Alayrac, Jean-Baptiste; Yu, Jiahui; Soricut, Radu; Schalkwyk, Johan; Dai, Andrew M; Hauth, Anja; Millican, Katie et al.: "Gemini: a family of highly capable multimodal models". In: *arXiv preprint arXiv:2312.11805* (2023) (cit. on p. 159).

[Tia20] Tian, Zhuotao; Zhao, Hengshuang; Shu, Michelle; Yang, Zhicheng; Li, Ruiyu and Jia, Jiaya: "Prior guided feature enrichment network for few-shot segmentation". In: *IEEE*

transactions on pattern analysis and machine intelligence 44.2 (2020), pp. 1050–1065 (cit. on pp. 41, 118, 125).

[Tia22] Tian, Zhuotao; Lai, Xin; Jiang, Li; Liu, Shu; Shu, Michelle; Zhao, Hengshuang and Jia, Jiaya: "Generalized few-shot semantic segmentation". In: *Proceedings of the CVPR*. 2022, pp. 11563–11572 (cit. on p. 41).

[Tou21] Touvron, Hugo; Cord, Matthieu; Douze, Matthijs; Massa, Francisco; Sablayrolles, Alexandre and Jégou, Hervé: "Training data-efficient image transformers & distillation through attention". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 10347–10357 (cit. on p. 18).

[Tra21] Tranheden, Wilhelm; Olsson, Viktor; Pinto, Juliano and Svensson, Lennart: "DACS: Domain Adaptation via Cross-Domain Mixed Sampling". In: *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2021, pp. 1378–1388. doi: 10.1109/WACV48630.2021.00142 (cit. on pp. 132, 150).

[TuS17] TuSimple: TuSimple Dataset Benchmark. https://github.com/TuSimple/tusimple-benchmark. 2017 (cit. on p. 24).

[Tze14] Tzeng, Eric; Hoffman, Judy; Zhang, Ning; Saenko, Kate and Darrell, Trevor: "Deep domain confusion: Maximizing for domain invariance". In: *arXiv preprint arXiv:1412.3474* (2014) (cit. on p. 38).

[Tze17] Tzeng, Eric; Hoffman, Judy; Saenko, Kate and Darrell, Trevor: "Adversarial discriminative domain adaptation". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2017, pp. 7167–7176 (cit. on p. 38).

[Uly16] Ulyanov, Dmitry; Vedaldi, Andrea and Lempitsky, Victor S.: "Instance Normalization: The Missing Ingredient for Fast Stylization". In: *CoRR* abs/1607.08022 (2016). arXiv: 1607.08022. url: http://arxiv.org/abs/1607.08022 (cit. on p. 14).

[Uly17] Ulyanov, Dmitry; Vedaldi, Andrea and Lempitsky, Victor: "Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2017, pp. 6924–6932 (cit. on p. 33).

[Van20] Van Amersfoort, Joost; Smith, Lewis; Teh, Yee Whye and Gal, Yarin: "Uncertainty estimation using a single deep deterministic neural network". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 9690–9700 (cit. on pp. 58, 65).

[Vas17] Vaswani, Ashish; Shazeer, Noam; Parmar, Niki; Uszkoreit, Jakob; Jones, Llion; Gomez, Aidan N.; Kaiser, Lukasz and Polosukhin, Illia: Attention Is All You Need. June 2017. arXiv: 1706.03762 [cs.CL] (cit. on p. 17).

[Ver12] Verma, Nakul; Mahajan, Dhruv; Sellamanickam, Sundararajan and Nair, Vinod: "Learning hierarchical similarity metrics". In: *2012 IEEE conference on Computer Vision and Pattern Recognition*. IEEE. 2012, pp. 2280–2287 (cit. on p. 39).

[Vin16] Vinyals, Oriol; Blundell, Charles; Lillicrap, Timothy; Wierstra, Daan et al.: "Matching networks for one shot learning". In: *Advances in neural information processing systems* 29 (2016) (cit. on p. 40).

[Vit22] Vitelli, Matt; Chang, Yan; Ye, Yawei; Ferreira, Ana; Wołczyk, Maciej; Osiński, Błażej; Niendorf, Moritz; Grimmett, Hugo; Huang, Qiangui; Jain, Ashesh et al.: "Safetynet: Safe planning for real-world self-driving vehicles using machine-learned policies". In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE. 2022, pp. 897–904 (cit. on p. 3).

[Vo21] Vo, Van Huy; Sizikova, Elena; Schmid, Cordelia; Pérez, Patrick and Ponce, Jean: "Large-scale unsupervised object discovery". In: *NeurIPS* 34 (2021), pp. 16764–16778 (cit. on p. 63).

[Vu19] Vu, Tuan-Hung; Jain, Himalaya; Bucher, Maxime; Cord, Matthieu and Pérez, Patrick: "Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation". In: *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*. 2019, pp. 2517–2526 (cit. on p. 38).

[Wan04] Wang, Zhou; Bovik, A.C.; Sheikh, H.R. and Simoncelli, E.P.: "Image quality assessment: from error visibility to structural similarity". In: *IEEE Transactions on Image Processing* 13.4 (2004), pp. 600–612. doi: 10.1109/TIP.2003.819861 (cit. on p. 46).

[Wan18] Wan, Fang; Wei, Pengxu; Jiao, Jianbin; Han, Zhenjun and Ye, Qixiang: "Min-entropy latent model for weakly supervised object detection". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2018, pp. 1297–1306 (cit. on p. 37).

[Wan19a] Wang, Kaixin; Liew, Jun Hao; Zou, Yingtian; Zhou, Daquan and Feng, Jiashi: "Panet: Few-shot image semantic segmentation with prototype alignment". In: *Proceedings of the CVPR*. 2019, pp. 9197–9206 (cit. on p. 41).

[Wan19b] Wang, Qi; Gao, Junyu and Li, Xuelong: "Weakly Supervised Adversarial Domain Adaptation for Semantic Segmentation in Urban Scenes". In: *IEEE Transactions on Image Processing* 28 (2019), pp. 4376–4386. doi: 10.1109/TIP.2019.2910667 (cit. on p. 38).

[Wan20a] Wang, Haoran; Shen, Tong; Zhang, Wei; Duan, Ling-Yu and Mei, Tao: "Classes matter: A fine-grained adversarial approach to cross-domain semantic segmentation". In: *ECCV*. Springer. 2020 (cit. on pp. 58, 67).

[Wan20b] Wang, Jingdong; Sun, Ke; Cheng, Tianheng; Jiang, Borui; Deng, Chaorui; Zhao, Yang; Liu, Dong; Mu, Yadong; Tan, Mingkui; Wang, Xinggang et al.: "Deep high-resolution representation learning for visual recognition". In: *IEEE transactions on pattern analysis and machine intelligence* 43.10 (2020), pp. 3349–3364 (cit. on p. 21).

[Wan21] Wang, Haoran; Liu, Weitang; Bocchieri, Alex and Li, Yixuan: "Can multi-label classification networks know what they don't know?" In: *NeurIPS* (2021) (cit. on pp. 58, 65).

[Wan22a] Wang, Jindong; Lan, Cuiling; Liu, Chang; Ouyang, Yidong; Qin, Tao; Lu, Wang; Chen, Yiqiang; Zeng, Wenjun and Yu, Philip: "Generalizing to unseen domains: A survey on domain generalization". In: *IEEE Transactions on Knowledge and Data Engineering* (2022) (cit. on p. 32).

[Wan22b] Wang, Jinsheng; Ma, Yinchao; Huang, Shaofei; Hui, Tianrui; Wang, Fei; Qian, Chen and Zhang, Tianzhu: "A Keypoint-Based Global Association Network for Lane Detection". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 1382–1391. doi: 10.1109/CVPR52688.2022.00145 (cit. on pp. 3, 131, 137).

[Wan23a] Wang, Runze; Jiang, Haoyu and Li, Yufei: "UPerNet with ConvNeXt for Semantic Segmentation". In: *2023 IEEE 3rd International Conference on Electronic Technology, Communication and Information (ICETCI)*. 2023, pp. 764–769. doi: 10.1109/ICETCI57876.2023.10177013 (cit. on p. 3).

[Wan23b] Wang, Yuting; Guerrero, Ricardo and Pavlovic, Vladimir: "D2F2WOD: Learning Object Proposals for Weakly-Supervised Object Detection via Progressive Domain Adaptation". In: *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2023, pp. 22–31. doi: 10.1109/WACV56688.2023.00011 (cit. on p. 38).

[Wan23c] Wang, Zeyu; Bai, Yutong; Zhou, Yuyin and Xie, Cihang: "Can CNNs Be More Robust Than Transformers?" In: *International Conference on Learning Representations*. 2023 (cit. on pp. 18, 102).

[Wan25] Wang, Yan; Luo, Wenjie; Bai, Junjie; Cao, Yulong; Che, Tong; Chen, Ke; Chen, Yuxiao; Diamond, Jenna; Ding, Yifan; Ding, Wenhao et al.: "Alpamayo-r1: Bridging reasoning and action prediction for generalizable autonomous driving in the long tail". In: *arXiv preprint arXiv:2511.00088* (2025) (cit. on p. 3).

[Way24] Waymo: Waymo One operates 24/7 across San Francisco and Daly City. Aug. 2024. URL: https://waymo.com/waymo-one-san-francisco/ (visited on 08/22/2024) (cit. on p. 1).

[Weh18] Wehrmann, Jonatas; Cerri, Ricardo and Barros, Rodrigo: "Hierarchical multi-label classification networks". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 5075–5084 (cit. on p. 39).

[Wei17] Wei, Yunchao; Feng, Jiashi; Liang, Xiaodan; Cheng, Ming-Ming; Zhao, Yao and Yan, Shuicheng: "Object Region Mining with Adversarial Erasing: A Simple Classification to Semantic Segmentation Approach". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 6488–6496. DOI: 10.1109/CVPR.2017.687 (cit. on p. 37).

[Wil45] Wilcoxon, Frank: "Individual comparisons by ranking methods". In: *Biometrics bulletin* 1.6 (1945), pp. 80–83 (cit. on p. 48).

[Woj17] Wojke, Nicolai; Bewley, Alex and Paulus, Dietrich: "Simple online and realtime tracking with a deep association metric". In: *2017 IEEE international conference on image processing (ICIP)*. IEEE. 2017, pp. 3645–3649 (cit. on p. 3).

[Won20] Wong, Kelvin; Wang, Shenlong; Ren, Mengye; Liang, Ming and Urtasun, Raquel: "Identifying unknown instances for autonomous driving". In: *Conference on Robot Learning*. PMLR. 2020, pp. 384–393 (cit. on pp. 58, 63).

[Wu14] Wu, Pei-Chen; Chang, Chin-Yu and Lin, Chang Hong: "Lane-mark extraction for automobiles under complex conditions". In: *Pattern Recognition* 47.8 (2014), pp. 2756–2767 (cit. on p. 22).

[Wu18] Wu, Yuxin and He, Kaiming: "Group normalization". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 3–19 (cit. on p. 14).

[Xia16] Xian, Yongqin; Akata, Zeynep; Sharma, Gaurav; Nguyen, Quynh; Hein, Matthias and Schiele, Bernt: "Latent embeddings for zero-shot classification". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2016, pp. 69–77 (cit. on p. 39).

[Xia18] Xiao, Tete; Liu, Yingcheng; Zhou, Bolei; Jiang, Yuning and Sun, Jian: Unified Perceptual Parsing for Scene Understanding. 2018. arXiv: 1807.10221 [cs.CV] (cit. on pp. 3, 21, 46, 95).

[Xia22] Xiao, Junfei; Xu, Zhichao; Lan, Shiyi; Yu, Zhiding; Yuille, Alan and Anandkumar, Anima: "1st Place Solution of The Robust Vision Challenge (RVC) 2022 Semantic Segmentation Track". In: *arXiv preprint arXiv:2210.12852* (2022) (cit. on p. 33).

[Xia24] Xiao, Bin; Wu, Haiping; Xu, Weijian; Dai, Xiyang; Hu, Houdong; Lu, Yumao; Zeng, Michael; Liu, Ce and Yuan, Lu: "Florence-2: Advancing a unified representation for a variety of vision tasks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 4818–4829 (cit. on pp. 5, 159).

[Xie17] Xie, Saining; Girshick, Ross; Dollár, Piotr; Tu, Zhuowen and He, Kaiming: "Aggregated residual transformations for deep neural networks". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2017, pp. 1492–1500 (cit. on p. 21).

[Xie20] Xie, Qizhe; Dai, Zihang; Hovy, Eduard; Luong, Thang and Le, Quoc: "Unsupervised data augmentation for consistency training". In: *Advances in neural information processing systems* 33 (2020), pp. 6256–6268 (cit. on p. 32).

[Xie21] Xie, Enze; Wang, Wenhai; Yu, Zhiding; Anandkumar, Anima; Alvarez, Jose M and Luo, Ping: "SegFormer: Simple and efficient design for semantic segmentation with transformers". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 12077–12090 (cit. on pp. 17, 21, 22).

[Xu20] Xu, Hang; Wang, Shaoju; Cai, Xinyue; Zhang, Wei; Liang, Xiaodan and Li, Zhenguo: CurveLane-NAS: Unifying Lane-Sensitive Architecture Search and Adaptive Point Blending. 2020 (cit. on pp. 22, 25).

[Yan15] Yan, Zhicheng; Zhang, Hao; Piramuthu, Robinson; Jagadeesh, Vignesh; DeCoste, Dennis; Di, Wei and Yu, Yizhou: "HD-CNN: hierarchical deep convolutional neural networks for large scale visual recognition". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 2740–2748 (cit. on p. 39).

[Yan18] Yang, Weijiao; Yang, Xianhai; Wei, Yudong and Cheng, Xiang: "Study of the recognition and tracking methods for lane lines based on image edge detections". In: *2018 International Symposium on Communication Engineering & Computer Science (CECS 2018)*. Atlantis Press. 2018, pp. 179–185 (cit. on p. 22).

[Yan19] Yang, Shun; Wu, Jian; Shan, Yanhu; Yu, Yinan and Zhang, Sumin: A novel vision-based framework for real-time lane detection and tracking. Tech. rep. SAE Technical Paper, 2019 (cit. on p. 22).

[Yan21a] Yang, Jingkang; Zhou, Kaiyang; Li, Yixuan and Liu, Ziwei: "Generalized Out-of-Distribution Detection: A Survey". In: *arXiv preprint arXiv:2110.11334* (2021) (cit. on p. 65).

[Yan21b] Yang, Kaiyu; Yau, Jacqueline; Fei-Fei, Li; Deng, Jia and Russakovsky, Olga: "A study of face obfuscation in imagenet". In: *arXiv preprint arXiv:2103.06191* (2021) (cit. on p. 33).

[Yan22] Yang, Suorong; Xiao, Weikang; Zhang, Mengcheng; Guo, Suhan; Zhao, Jian and Shen, Furao: "Image data augmentation for deep learning: A survey". In: *arXiv preprint arXiv:2204.08610* (2022) (cit. on p. 32).

[Yan23a] Yan, Xu; Zheng, Chaoda; Li, Zhen; Cui, Shuguang and Dai, Dengxin: "Benchmarking the Robustness of LiDAR Semantic Segmentation Models". In: *arXiv preprint arXiv:2301.00970* (2023) (cit. on p. 32).

[Yan23b] Yang, Ze; Chen, Yun; Wang, Jingkang; Manivasagam, Sivabalan; Ma, Wei-Chiu; Yang, Anqi Joyce and Urtasun, Raquel: "Unisim: A neural closed-loop sensor simulator". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 1389–1399 (cit. on p. 159).

[Yu19] Yu, Chunlei; Wen, Tuopu; Chen, Long and Jiang, Kun: "Common Bird-View Transformation for Robust Lane Detection". In: *2019 IEEE 9th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*. 2019, pp. 665–669. DOI: 10.1109/CYBER46603.2019.9066500 (cit. on p. 38).

[Yu20] Yu, Fisher; Chen, Haofeng; Wang, Xin; Xian, Wenqi; Chen, Yingying; Liu, Fangchen; Madhavan, Vashisht and Darrell, Trevor: "Bdd100k: A diverse driving dataset for heterogeneous multitask learning". In: *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*. 2020, pp. 2636–2645 (cit. on p. 24).

[Yua21] Yuan, Yuhui; Fu, Rao; Huang, Lang; Lin, Weihong; Zhang, Chao; Chen, Xilin and Wang, Jingdong: "HRFormer: High-Resolution Vision Transformer for Dense Predict". In: *Advances in Neural Information Processing Systems*. Ed. by Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.S. and Vaughan, J. Wortman. Vol. 34. Curran Associates, Inc., 2021, pp. 7281–7293 (cit. on pp. 21, 22).

[Yun19] Yun, Sangdoo; Han, Dongyoon; Oh, Seong Joon; Chun, Sanghyuk; Choe, Junsuk and Yoo, Youngjoon: "Cutmix: Regularization strategy to train strong classifiers with localizable features". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 6023–6032 (cit. on p. 32).

[Zae21] Zaeemzadeh, Alireza; Bisagno, Niccolo; Sambugaro, Zeno; Conci, Nicola; Rahnavard, Nazanin and Shah, Mubarak: "Out-of-distribution detection using union of 1-dimensional subspaces". In: *CVPR*. 2021 (cit. on pp. 58, 65).

[Zei14] Zeiler, Matthew D and Fergus, Rob: "Visualizing and understanding convolutional networks". In: *European Conference on Computer Vision*. Springer. 2014, pp. 818–833 (cit. on p. 17).

[Zha11] Zhao, Bin; Li, Fei and Xing, Eric: "Large-scale category structure aware image categorization". In: *NeurIPS* (2011) (cit. on p. 39).

[Zha17] Zhao, Hengshuang; Shi, Jianping; Qi, Xiaojuan; Wang, Xiaogang and Jia, Jiaya: "Pyramid scene parsing network". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2017, pp. 2881–2890. arXiv: 1612.01105 [cs.CV] (cit. on pp. 46, 95).

[Zha19] Zhang, Chi; Lin, Guosheng; Liu, Fayao; Yao, Rui and Shen, Chunhua: "Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning". In: *Proceedings of the CVPR*. 2019, pp. 5217–5226 (cit. on p. 41).

[Zha20] Zhang, Xiaolin; Wei, Yunchao; Yang, Yi and Huang, Thomas S: "Sg-one: Similarity guidance network for one-shot semantic segmentation". In: *IEEE transactions on cybernetics* 50.9 (2020), pp. 3855–3865 (cit. on p. 41).

[Zha21a] Zhang, Pan; Zhang, Bo; Zhang, Ting; Chen, Dong; Wang, Yong and Wen, Fang: "Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation". In: *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*. 2021, pp. 12414–12424 (cit. on pp. 132, 150).

[Zha21b] Zhao, Xiangyun; Vemulapalli, Raviteja; Mansfield, Philip Andrew; Gong, Boqing; Green, Bradley; Shapira, Lior and Wu, Ying: "Contrastive learning for label efficient semantic segmentation". In: *CVPR*. 2021 (cit. on pp. 58, 62).

[Zha22a] Zhang, Chongzhi; Zhang, Mingyuan; Zhang, Shanghang; Jin, Daisheng; Zhou, Qiang; Cai, Zhongang; Zhao, Haiyu; Liu, Xianglong and Liu, Ziwei: "Delving deep into the generalization of vision transformers under distribution shifts". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 7277–7286 (cit. on p. 18).

[Zha22b] Zhang, Dingwen; Zeng, Wenyuan; Yao, Jieru and Han, Junwei: "Weakly Supervised Object Detection Using Proposal- and Semantic-Level Relationships". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.6 (2022), pp. 3349–3363. DOI: 10.1109/TPAMI.2020.3046647 (cit. on p. 37).

[Zha22c] Zhang, Jian-Wei; Sun, Yifan; Yang, Yi and Chen, Wei: "Feature-Proxy Transformer for Few-Shot Segmentation". In: *Advances in Neural Information Processing Systems*. Ed. by Oh, Alice H.; Agarwal, Alekh; Belgrave, Danielle and Cho, Kyunghyun. 2022 (cit. on pp. 41, 115).

[Zha22d] Zhang, Yifu; Sun, Peize; Jiang, Yi; Yu, Dongdong; Weng, Fucheng; Yuan, Zehuan; Luo, Ping; Liu, Wenyu and Wang, Xinggang: "Bytetrack: Multi-object tracking by associating every detection box". In: *European Conference on Computer Vision*. Springer. 2022, pp. 1–21 (cit. on p. 3).

[Zha22e] Zhang, Youcheng; Lu, Zongqing; Zhang, Xuechen; Xue, Jing-Hao and Liao, Qingmin: "Deep Learning in Lane Marking Detection: A Survey". In: *IEEE Transactions on Intelligent Transportation Systems* 23 (2022), pp. 5976–5992. DOI: 10.1109/TITS.2021.3070111 (cit. on p. 22).

[Zhe21] Zheng, Tu; Fang, Hao; Zhang, Yi; Tang, Wenjian; Yang, Zheng; Liu, Haifeng and Cai, Deng: "RESA: Recurrent Feature-Shift Aggregator for Lane Detection". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35 (2021), pp. 3547–3554. DOI: 10.1609/aaai.v35i4.16469 (cit. on p. 22).

[Zhe22] Zheng, Tu; Huang, Yifei; Liu, Yang; Tang, Wenjian; Yang, Zheng; Cai, Deng and He, Xiaofei: "CLRNet: Cross Layer Refinement Network for Lane Detection". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 888–897. DOI: 10.1109/CVPR52688.2022.00097 (cit. on pp. 3, 23, 131, 133, 137, 148).

[Zho16] Zhou, Bolei; Khosla, Aditya; Lapedriza, Agata; Oliva, Aude and Torralba, Antonio: "Learning deep features for discriminative localization". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2016, pp. 2921–2929 (cit. on pp. 36, 37).

[Zho22a] Zhou, Jinghao; Wei, Chen; Wang, Huiyu; Shen, Wei; Xie, Cihang; Yuille, Alan and Kong, Tao: "ibot: Image bert pre-training with online tokenizer". In: *International Conference on Learning Representations*. 2022 (cit. on p. 32).

[Zho22b] Zhou, Jingxing and Beyerer, Jürgen: "Impacts of Data Anonymization on Semantic Segmentation". In: *2022 IEEE Intelligent Vehicles Symposium, IV 2022, Aachen, Germany, June 4-9, 2022*. IEEE, 2022, pp. 997–1004. DOI: 10.1109/IV51971.2022.9827262. URL: https://doi.org/10.1109/IV51971.2022.9827262 (cit. on pp. 6, 43).

[Zho22c] Zhou, Zikang; Ye, Luyao; Wang, Jianping; Wu, Kui and Lu, Kejie: "Hivt: Hierarchical vector transformer for multi-agent motion prediction". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 8823–8833 (cit. on p. 3).

[Zho23a] Zhou, Hongjie: "Few-Shot Semantic Segmentation Based on Dual-Branch Feature Extraction". In: *2023 International Conference on Pattern Recognition, Machine Vision and Intelligent Algorithms (PRMVIA)*. 2023, pp. 287–291. DOI: 10.1109/PRMVIA58252.2023.00053 (cit. on p. 115).

[Zho23b] Zhou, Jingxing and Beyerer, Jürgen: "Category Differences Matter: A Broad Analysis of Inter-Category Error in Semantic Segmentation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023 - Workshops, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, 2023, pp. 3870–3880. DOI: 10.1109/CVPRW59228.2023.00401. URL: https://doi.org/10.1109/CVPRW59228.2023.00401 (cit. on pp. 4, 7, 92).

[Zho23c] Zhou, Jingxing and Beyerer, Jürgen: "Corner Cases in Data-Driven Automated Driving: Definitions, Properties and Solutions". In: *IEEE Intelligent Vehicles Symposium, IV 2023, Anchorage, AK, USA, June 4-7, 2023*. IEEE, 2023. DOI: 10.1109/IV55152.2023.10186558. URL: https://doi.org/10.1109/IV55152.2023.10186558 (cit. on pp. 6, 57).

[Zho23d] Zhou, Jingxing; Wandelburg, Nick and Beyerer, Jürgen: "Unknown-Aware Hierarchical Object Detection in the Context of Automated Driving". In: *2023 IEEE 26th International Conference on Intelligent Transportation Systems, ITSC 2023, Bilbao, Spain, September 24-28, 2023*. IEEE, 2023, pp. 2501–2508. DOI: 10.1109/ITSC57777.2023.10422562. URL: https://doi.org/10.1109/ITSC57777.2023.10422562 (cit. on pp. 7, 71).

[Zho24a] Zhou, Jingxing; Chen, Ruei-Bo and Beyerer, Jürgen: "Few-Shot Semantic Segmentation for Complex Driving Scenes". In: *IEEE Intelligent Vehicles Symposium, IV 2024, Jeju Island, Republic of Korea, June 2-5, 2024*. IEEE, 2024, pp. 695–702. DOI: 10.1109/IV55156.2024.10588850. URL: https://doi.org/10.1109/IV55156.2024.10588850 (cit. on pp. 7, 92).

[Zho24b] Zhou, Jingxing; Zhang, Chongzhe and Beyerer, Jürgen: "Towards Weakly-Supervised Domain Adaptation for Lane Detection". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024 - Workshops, Seattle, WA, USA, June 17-18, 2024*. IEEE, 2024, pp. 3553–3563. DOI: 10.1109/CVPRW63382.2024.00359. URL: https://doi.org/10.1109/CVPRW63382.2024.00359 (cit. on pp. 8, 130).

[Zhu17] Zhu, Xinqi and Bain, Michael: "B-CNN: branch convolutional neural network for hierarchical classification". In: *arXiv preprint arXiv:1709.09890* (2017) (cit. on p. 39).

[Zhu21] Zhu, Xizhou; Su, Weijie; Lu, Lewei; Li, Bin; Wang, Xiaogang and Dai, Jifeng: "Deformable DETR: Deformable Transformers for End-to-End Object Detection". In: *Proceedings of ICLR*. 2021 (cit. on pp. 3, 20, 78).

[Zwe07] Zweig, Alon and Weinshall, Daphna: "Exploiting Object Hierarchy: Combining Models from Different Category Levels". In: *2007 IEEE 11th International Conference on Computer Vision*. 2007, pp. 1–8. doi: 10.1109/ICCV.2007.4409064 (cit. on p. 39).

[Zwe20] Zwemer, M. H.; Wijnhoven, R. G. J. and de With, P. H. N.: "SSD-ML: Hierarchical Object Classification for Traffic Surveillance". In: *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP)*. SciTePress, 2020, pp. 250–259. doi: 10.5220/0008902402500259 (cit. on p. 40).

[Zwe22] Zwemer, Matthijs H.; Wijnhoven, Rob G. J. and With, Peter H. N. de: "Hierarchical Object Detection and Classification Using SSD Multi-Loss". In: *Proceedings of VISIGRAPP*. Springer International Publishing, 2022, pp. 268–296 (cit. on p. 40).

Own publications

[1] Zhou, Jingxing and Beyerer, Jürgen: "Impacts of Data Anonymization on Semantic Segmentation". In: *2022 IEEE Intelligent Vehicles Symposium, IV 2022, Aachen, Germany, June 4-9, 2022*. IEEE, 2022, pp. 997–1004. DOI: 10.1109/IV51971.2022.9827262. URL: https://doi.org/10.1109/IV51971.2022.9827262.

[2] Eisemann, Leon et al.: "An Approach to Systematic Data Acquisition and Data-Driven Simulation for the Safety Testing of Automated Driving Functions". In: *2023 IEEE 26th International Conference on Intelligent Transportation Systems, ITSC 2023, Bilbao, Spain, September 24-28, 2023*. IEEE, 2023, pp. 2440–2447. DOI: 10.1109/ITSC57777.2023.10422676. URL: https://doi.org/10.1109/ITSC57777.2023.10422676.

[3] Kalb, Tobias; Ahuja, Niket; Zhou, Jingxing and Beyerer, Jürgen: "Effects of Architectures on Continual Semantic Segmentation". In: *IEEE Intelligent Vehicles Symposium, IV 2023, Anchorage, AK, USA, June 4-7, 2023*. IEEE, 2023. DOI: 10.1109/IV55152.2023.10186597. URL: https://doi.org/10.1109/IV55152.2023.10186597.

[4] Zhou, Jingxing and Beyerer, Jürgen: "Category Differences Matter: A Broad Analysis of Inter-Category Error in Semantic Segmentation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023 - Workshops, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, 2023, pp. 3870–3880. DOI: 10.1109/CVPRW59228.2023.00401. URL: https://doi.org/10.1109/CVPRW59228.2023.00401.

[5] Zhou, Jingxing and Beyerer, Jürgen: "Corner Cases in Data-Driven Automated Driving: Definitions, Properties and Solutions". In: *IEEE Intelligent Vehicles Symposium, IV 2023, Anchorage, AK, USA, June*

4-7, 2023. IEEE, 2023. DOI: 10.1109/IV55152.2023.10186558. URL: https://doi.org/10.1109/IV55152.2023.10186558.

[6] ZHOU, Jingxing; WANDELBURG, Nick and BEYERER, Jürgen: "Unknown-Aware Hierarchical Object Detection in the Context of Automated Driving". In: *2023 IEEE 26th International Conference on Intelligent Transportation Systems, ITSC 2023, Bilbao, Spain, September 24-28, 2023*. IEEE, 2023, pp. 2501–2508. DOI: 10.1109/ITSC57777.2023.10422562. URL: https://doi.org/10.1109/ITSC57777.2023.10422562.

[7] ZHOU, Jingxing; CHEN, Ruei-Bo and BEYERER, Jürgen: "Few-Shot Semantic Segmentation for Complex Driving Scenes". In: *IEEE Intelligent Vehicles Symposium, IV 2024, Jeju Island, Republic of Korea, June 2-5, 2024*. IEEE, 2024, pp. 695–702. DOI: 10.1109/IV55156.2024.10588850. URL: https://doi.org/10.1109/IV55156.2024.10588850.

[8] ZHOU, Jingxing; ZHANG, Chongzhe and BEYERER, Jürgen: "Towards Weakly-Supervised Domain Adaptation for Lane Detection". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024 - Workshops, Seattle, WA, USA, June 17-18, 2024*. IEEE, 2024, pp. 3553–3563. DOI: 10.1109/CVPRW63382.2024.00359. URL: https://doi.org/10.1109/CVPRW63382.2024.00359.

List of Figures

List of Tables

Acronyms

AD	Automated Driving
ADAS	Advanced Driver Assistance System
A-OSE	Absolute Open Set Error
AP	Average Precision
AR	Average Recall
ASPP	Atrous Spatial Pyramid Pooling
BAM	Base and the Meta learners
CAM	Class Activation Map
CCPA	California Consumer Privacy Act
CE	Cross-Entropy
CER	Critical Error Rate
CNN	Convolutional Neural Network
DETR	Detection Transformer
DG	Domain Generalization

DLv3+	DeepLabv3Plus
EMA	Exponential Moving Average
FCN	Fully Convolutional Network
FN	False Negative
FNR	False Negative Rate
FP	False Positive
FPR	False Positive Rate
FSSS	Few-Shot Semantic Segmentation
GAN	Generative Adversarial Network
GDPR	General Data Protection Regulation
GFS-Seg	Generalized Few-Shot Semantic Segmentation
HAD	Highly Automated Driving
HC	Hierarchical Constraints
i.i.d.	independent and identically distributed
IoU	Intersection over Union
LiDAR	Light Detection and Ranging
LLM	Large Language Model
LOO	Leave-One-Out

MAE	Mean Absolute Error
MAP	Masked Average Pooling
mAP	mean Average Precision
mIoU	mean Intersection-over-Union
MSE	Mean Squared Error
NLP	Natural Language Processing
NoL	Number-of-Lanes
OOD	out-of-distribution
ORE	Open World Object Detector
OW-DETR	Open World Detection Transformer
PII	Personally Identifiable Information
PIPL	Personal Information Protection Law
PSPNet	Pyramid Scene Parsing Network
RaDAR	Radio Detection and Ranging
ReLU	Rectified Linear Unit
RL	Relabeling
RoI	Region of Interest
RPN	Region Proposal Network

SGD	Stochastic Gradient Descent
SSD	Single Shot Detector
SSIM	Structural Similarity Index
SVF	Singular Value Fine-tuning
TP	True Positive
TSUDA	Teacher–Student Unsupervised Domain Adaptation
UDA	Unsupervised Domain Adaptation
UPerNet	Unified Perceptual Parsing Network
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-Everything
ViT	Vision Transformer
VRU	Vulnerable Road User
WI	Wilderness Impact
WSDAL	Weakly Supervised Domain Adaptation framework for Lane Detection

A Datasets for Unknown-aware Object Detection

The label distribution for the synthetic dataset that has been used in Ch. 6 is detailed in Tab. A.1. During training, images containing *truck* and *van* are excluded. For the evaluation of unknown-aware object detection models, those two classes are included in the evaluation as *unknown unknown* objects.

Table A.1: Training and validation set label distributions of the synthetic dataset.

Category	Annotations	Ratio
Training		
car	5,605	36.17%
motorcycle	1,907	12.31%
bicycle	1,910	12.33%
person	6,071	39.19%
truck	0	0.00%
van	0	0.00%
(total)	15,493	100.00%
Validation		
car	2,647	26.76%
motorcycle	897	9.07%
bicycle	945	9.56%
person	2,786	28.17%
truck	672	6.80%
van	1,942	19.64%
(total)	9,889	100.00%

In addition to the synthetic dataset that is generated using the simulation engine CARLA, a real-world dataset nuImages is utilized for evaluation. Table A.3 depicts the label distribution of the objects in the training and validation set. It is important to note that the number of images and annotations differs from the original dataset, which is shown in Tab. A.2, due to the exclusion of annotations for environmental object classes, including *barrier*, *debris*, *pushable*, *pullable*, *traffic cone*, and *bicycle rack*. Consequently, images lacking any annotations have been removed from the dataset. Additionally, images containing objects from the classes *ambulance* and *police* have been discarded due to their significant underrepresentation in the dataset.

Table A.2: Original nuImages label distributions according to [Cae20].

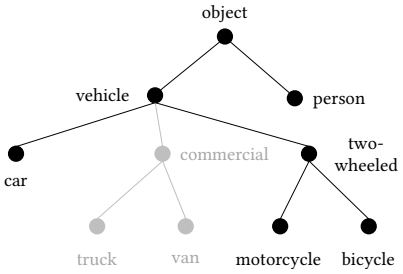
Category	Annotations	Ratio
animal	255	0.04%
human.pedestrian.adult	149,921	21.61%
human.pedestrian.child	1,934	0.28%
human.pedestrian.construction_worker	13,582	1.96%
human.pedestrian.personal_mobility	2,281	0.33%
human.pedestrian.police_officer	464	0.07%
human.pedestrian.stroller	363	0.05%
human.pedestrian.wheelchair	35	0.01%
vehicle.bicycle	17,060	2.46%
vehicle.bus.bendy	265	0.04%
vehicle.bus.rigid	8,361	1.21%
vehicle.car	250,088	36.05%
vehicle.construction	6,071	0.88%
vehicle.emergency.ambulance	42	0.01%
vehicle.emergency.police	139	0.02%
vehicle.motorcycle	16,779	2.42%
vehicle.trailer	3,771	0.54%
vehicle.truck	36,314	5.23%
movable_object.barrier	88,545	12.76%
movable_object.debris	3,171	0.46%
movable_object.pushable_pullable	3,675	0.53%
movable_object.trafficcone	87,603	12.63%
static_object.bicycle_rack	3,064	0.44%
(total)	693,783	100.00%

Table A.3: nuImages training and validation set label distributions.

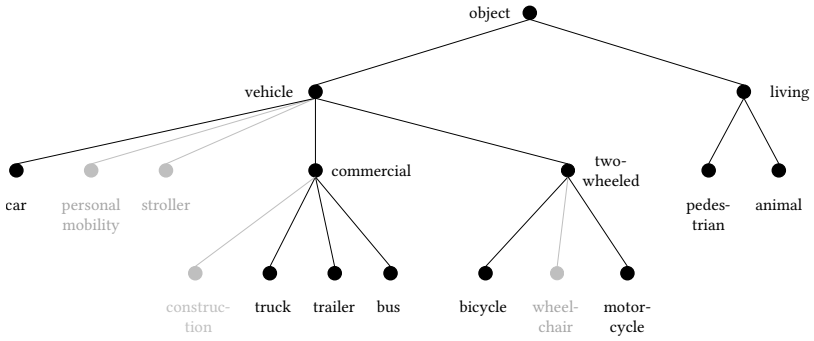
Category	Annotations	Ratio
Training		
car	179,997	51.28%
truck	25,059	7.14%
trailer	2,732	0.78%
bus	5,732	1.63%
bicycle	12,034	3.43%
motorcycle	12,245	3.49%
pedestrian	113,069	32.21%
animal	155	0.04%
personal mobility	0	0.00%
stroller	0	0.00%
construction	0	0.00%
wheelchair	0	0.00%
(total)	351,023	100.00%
Validation		
car	45,708	45.46%
truck	6,580	6.54%
trailer	464	0.46%
bus	1,764	1.75%
bicycle	3,225	3.21%
motorcycle	3,014	3.00%
pedestrian	30,970	30.80%
animal	81	0.08%
personal mobility	2,281	2.27%
stroller	363	0.36%
construction	6,071	6.04%
wheelchair	35	0.03%
(total)	100,556	100.00%

B Class Hierarchies

Unknown-aware Object Detection

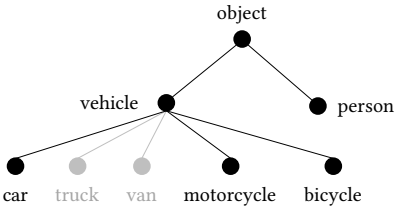


(a) Synthetic Dataset

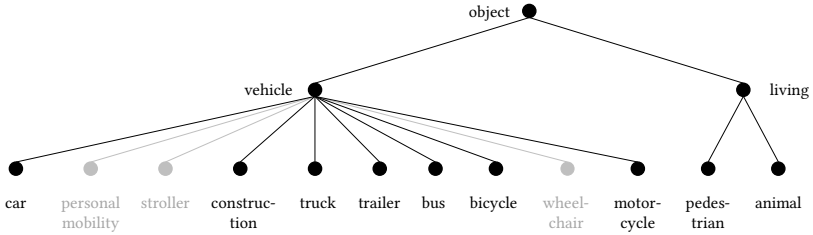


(b) nuImages

Figure B.1: Fine-grained class taxonomies of two datasets that are used in Ch. 6. Unknown classes used for evaluation are grayed out.



(a) Synthetic Dataset



(b) nuImages

Figure B.2: Coarse-grained class taxonomies of two datasets that are used in Ch. 6. Unknown classes used for evaluation are grayed out.

Figures B.1 and B.2 depict the fine- and coarse-grained class hierarchies that used in Ch. 6 for unknown-aware object detection. The coarse-grained class hierarchy eliminate the superclasses that are beyond *level-2*, e.g., *commercial* and *two-wheeled*.

Semantic Segmentation Datasets

For the task of semantic segmentation, five major datasets are utilized in this work, namely, Cityscapes, Mapillary, ACDC, A2D2 and BDD100k. Detailed class taxonomy for Cityscapes [Cor16] is provided in Fig. B.3, which is also shared with ACDC and BDD100k. Fig. B.4 depicts a simplified class hierarchy for Mapillary [Neu17], while Fig. B.5 and Fig. B.6 show the behavior-based class taxonomy and VRU-based class taxonomy that are used in the ablation study from Sec. 7.1.3. The classes under *void* are discarded in Mapillary and relabeled as ignore. Despite the fact that the A2D2 dataset does not provide an

official class hierarchy, it is feasible to group the *car*, *utility vehicle*, *truck*, *tractor*, and *small vehicles* under the category *vehicle* for the comparison, which is similar to the Cityscapes and Mapillary dataset.

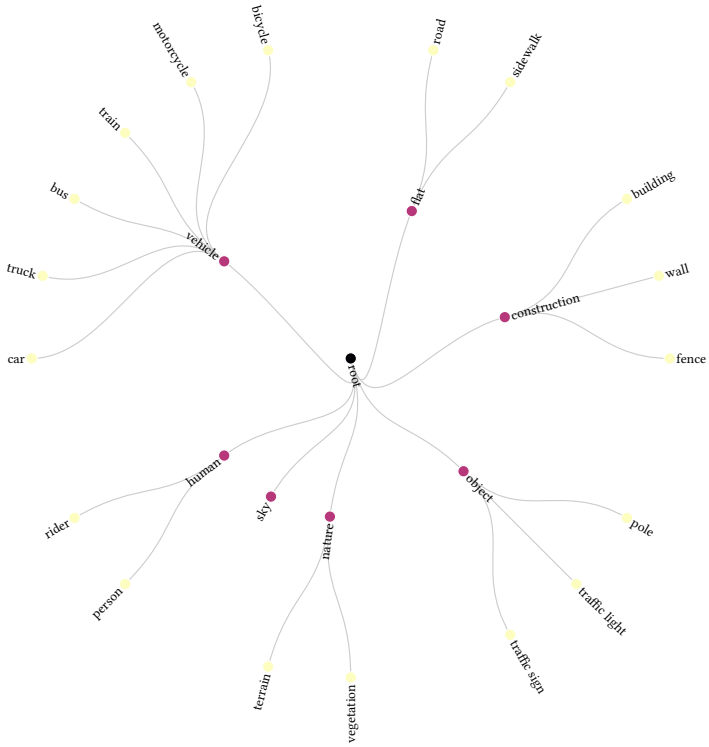


Figure B.3: Class taxonomy of the Cityscapes dataset, which is also adapted by ACDC and BDD100k.

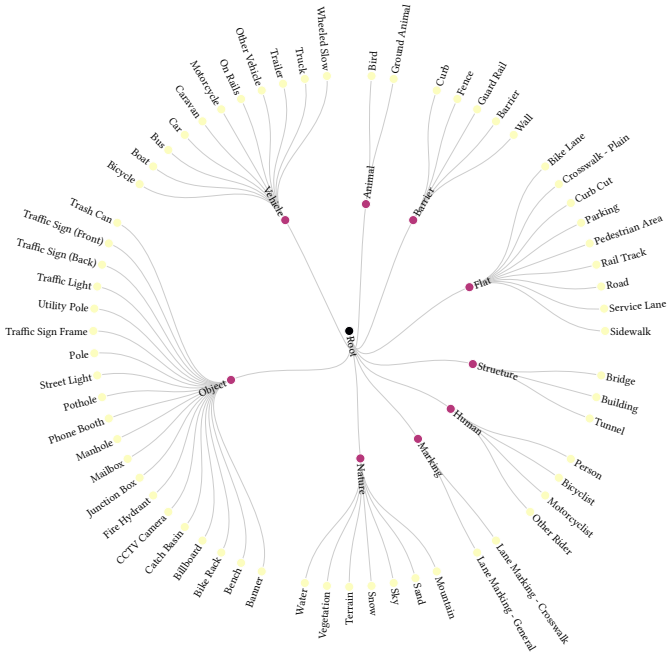


Figure B.4: Simplified class taxonomy of the Mapillary dataset (v1.2); void classes are remapped to ignore during training and validation.

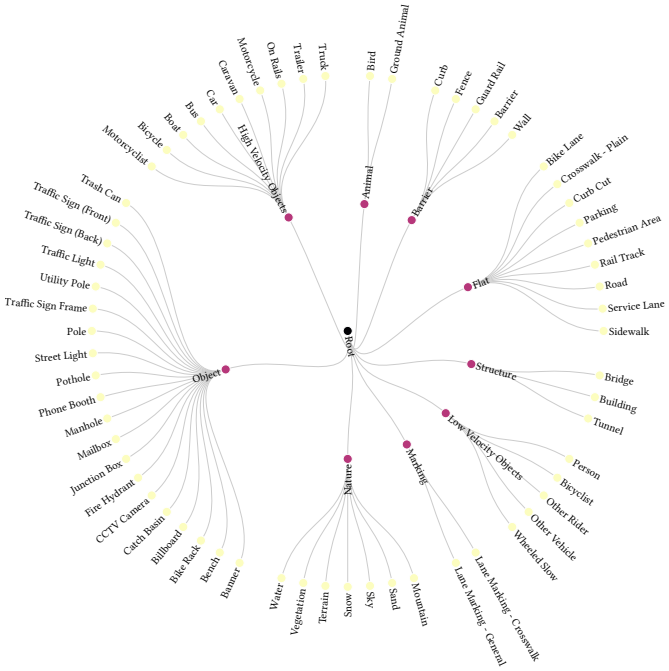


Figure B.5: Behavior-based class taxonomy used for the ablation study from the Mapillary dataset (v1.2); void classes are remapped to ignore during training and validation.

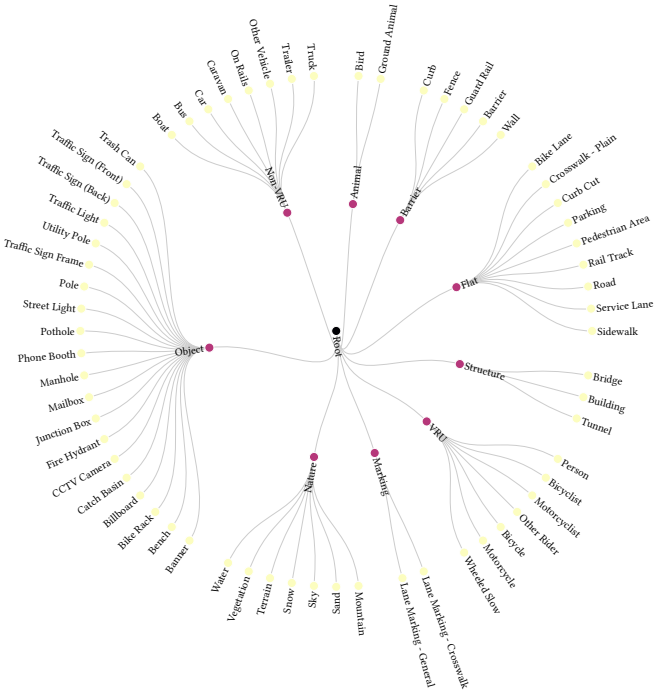


Figure B.6: Class taxonomy that addresses VRUs used for ablation study from the Mapillary dataset (v1.2); void classes are remapped to ignore during training and validation.