PhD-FSTM-2022-112
The Faculty of Science, Technology and Medicine

DISSERTATION

Defense held on 04/10/2022 in Luxembourg

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

EN INFORMATIQUE

by

François ROBINET
Born on 16th October 1993 in Woluwé-Saint-Lambert (Belgium)

MINIMIZING SUPERVISION FOR VISION-BASED PERCEPTION AND CONTROL IN AUTONOMOUS DRIVING

Dissertation Defense Committee

Dr. Djamila AOUADA,
Professor, University of Luxembourg

Dr. Raphaël FRANK, Supervisor,
Professor, University of Luxembourg

Dr. Christian HUNDT,
NVIDIA Germany

Dr. Christian MÜLLER,
Professor, DFKI Saarbrücken

Dr. Radu STATE, Chairman,
Professor, University of Luxembourg

UNIVERSITY OF LUXEMBOURG

Abstract

Interdisciplinary Centre for Security, Reliability and Trust

SEDAN Research Group

Minimizing Supervision for Vision-Based Perception and Control in Autonomous Driving

by François Robinet

The research presented in this dissertation focuses on reducing the need for supervision in two tasks related to autonomous driving: end-to-end steering and free space segmentation.

For end-to-end steering, we devise a new regularization technique which relies on pixel-relevance heatmaps to force the steering model to focus on lane markings. This improves performance across a variety of offline metrics. In relation to this work, we publicly release the RoboBus dataset, which consists of extensive driving data recorded using a commercial bus on a cross-border public transport route on the Luxembourgish-French border.

We also tackle pseudo-supervised free space segmentation from three different angles: (1) we propose a Stochastic Co-Teaching training scheme that explicitly attempts to filter out the noise in pseudo-labels, (2) we study the impact of self-training and of different data augmentation techniques, (3) we devise a novel pseudo-label generation method based on road plane distance estimation from approximate depth maps.

Finally, we investigate semi-supervised free space estimation and find that combining our techniques with a restricted subset of labeled samples results in substantial improvements in IoU, Precision and Recall.

Acknowledgments

In research as in life, it is important to give credit where credit is due.

I would like to start by thanking my PhD advisor, Prof. Raphaël Frank. Over the last four years, Raphaël has provided frequent and precious feedback on my work, he has given me the freedom to explore the academic topics I found most interesting, and he has supported me in teaching initiatives. I would additionally like to thank him for fostering a productive but highly enjoyable work atmosphere within 360Lab.

I also want to express my gratitude to my jury members. Prof. Radu State for leading the great SEDAN research team that welcomed me during these years. Dr. Christian Hundt for his communicative enthusiasm in all things machine learning, for his many insights into my research, and for encouraging me to get certified to teach NVIDIA workshops. Dr. Geoffrey Nichil and Michel Etienne for welcoming me at Foyer. Prof. Djamila Aouada and Prof. Christian Müller for agreeing to review my work.

I am grateful to the Luxembourg National Research Fund, the Interdisciplinary Centre for Security, Reliability and Trust, the University of Luxembourg and Foyer Assurances Luxembourg for funding my research. My gratitude also goes to the High-Performance Computing team of the University of Luxembourg, for allowing researchers to freely use their top-notch computing platform. No GPUs were harmed in making this research!

Contrary to popular belief, research does not have to be a lonely endeavour. I was fortunate enough to meet and work alongside brilliant colleagues and co-authors. Special thanks go to my awesome office mates Mehdi Testouri and Faisal Hawlader, for always coming to work with a smile and for enduring my long-winded rants about C++. I am particularly grateful to Georgios Varisteas for his mentorship when I first joined the University, and to Christian Colot and Gamal Elghazaly for many insightful discussions about life, research and self-driving vehicles. My deepest gratitude goes to Yussef Akl for his unrelenting willingness to learn, improve and joke around: my English banter skills have much improved in your presence!

I also want to thank my co-workers from the SEDAN team for many fun Friday breakfasts. I cannot name all of you for fear of forgetting someone, so I will only express a particular gratitude to Kathya Khramtsova, Ramiro Camino and Farouk Damoun for many great discussions. I am additionally very grateful to Valérie for her help in planning conference trips and for organizing phenomenal team building events.

Finally, I want to thank my family for their support, and for teaching me confidence and perseverance. Last, but certainly not least, I am thankful to Charlotte for loving me, for always being my strongest supporter and for her infinite patience during these years.

Contents

Acronyms

Adam Adaptive Moment Estimation

ADAS Advanced Driver Assistance Systems

CNN Convolutional Neural Network

DARPA Defense Advanced Research Projects Agency

ECU Engine Control Unit

FCN Fully-Convolutional Network

FSE Free Space Estimation

GD Gradient Descent

GNSS Global Navigation Satellite System

GPU Graphics Processing Unit

IMU Inertial Measurement Unit

IoU Intersection-over-Union

MAE Mean Absolute Error

MSE Mean Squared Error

NHTSA National Highway Traffic Safety Administration

PLG Pseudo-Label Generator

RPD Road Plane Distance

SC Superpixel Clustering

SGD Stochastic Gradient Descent

SLAM Simultaneous Localization and Mapping

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Importance of Autonomous Driving Technology

Autonomous driving technology has made impressive progress over the last two decades. One of the first milestones was achieved in 2005 at the Grand Challenge organized by the Defense Advanced Research Projects Agency (DARPA). At this occasion, five autonomous vehicles built by different research teams were able to safely navigate some 212 kilometers of the Mojave Desert [BIS07]. The performance was topped only two years later during the DARPA Urban Challenge, in the much more challenging environment of a mock-up town [BIS09]. In organizing these two competitions, DARPA hoped to spur the development of technologies needed to create the first fully-autonomous ground vehicles [BIS07]. This goal was met, and academic interests has since translated into industrial applications, with many manufacturers currently racing to build and release such vehicles on the market.

Road accidents have become a major health concern world-wide. The Association for Safe International Road Travel estimates that 1.35 million people die in road crashes each year [SIRT]. Up to another 50 million survive an accident with non-fatal injuries, but sometimes have to suffer long-term disabilities. For children and adults below the age of 30, injuries sustained in road crashes are the single greatest cause of death [SIRT]. An extensive survey of the causes of crashes from the National Highway Traffic Safety Administration (NHTSA) in the United States concluded that more than 94% of car accidents are caused at least partly by human error [Adm13]. Analysis of this survey reveals that the leading human-related causes of accidents can be attributed to alcohol consumption (30%), speeding (30%) and distracted driving (21%). Driver drowsiness also accounts for almost 3% of the analyzed accidents [Adm13]. The progressive increase of car autonomy is expected to reduce both the quantity and the severity of these accidents.

Given the challenge presented by full automation and the fact that Europeans drive cars that are on average 12 years old [ACE], car manufacturers are already shipping partial automation features in the form of Advanced Driver Assistance Systems (ADAS). Popular examples of these systems include forward collision warning, automatic emergency breaking, lane keeping assistance, or blind spot detection. The Highway Loss Data Institute and Insurance Institute for Highway Safety have compared police-reported crash rates and insurance claims for vehicles with and without ADAS technologies, and report

decreases in both crashes and claim rates [Fac]. In the case of vehicles equipped with automatic emergency breaking, the study reports a decrease of 50% of front-to-rear crashes and of 27% of accidents involving pedestrians.

The advent of autonomous vehicles may also challenge the insurance industry [Nel21]. While the number and severity of accidents are expected to decrease, the expensive sensor suite on-board self-driving vehicles will likely make damage claims more expensive. There are also major legal liability concerns on the horizon: as driving becomes more automated, the boundary between human and system failure will get blurrier.

Beyond road safety benefits, the availability of autonomous vehicles also has the potential to drastically improve access to transportation for the elderly, or for people suffering from a disability that prevents them from driving. The possibility of a vehicle traveling without a human driver is also likely to enable robo-taxi services at affordable prices, decreasing the need for individual car ownership. Since individual vehicles are idle most of the day, shared mobility would decrease the total amount of vehicles needed, and lower the carbon footprint of the whole industry. Finally, another ecological benefit could come from Vehicle-to-Vehicle communication: once cars start anticipating braking and acceleration decisions they will be able to further optimize their energy consumption [MMB16].

1.2 Autonomous Mobile Robots Design

The study of mobile robotics covers a wide-range of subjects and is by nature interdisciplinary. The traditional robotics view breaks the problem of navigation down into several distinct parts. One common break-down is depicted on Figure 1.1 and consists of the following steps operating in a cycle.

Sensing The sensor-suite embedded in autonomous vehicles usually consists of long- and short-range radars, LiDARs to build a sparse 3D point cloud of the surrounding environments, and cameras to collect high-resolution contextual information. These are accompanied by a precise Global Navigation Satellite System (GNSS) for global positioning and an Inertial Measurement Unit (IMU) to measure the acceleration of the robot.

Perception The Perception module is responsible for analyzing raw data coming from the sensors. Common steps include the tracking or segmentation of objects (pedestrians, other vehicles, traffic signs, lane markings, ...) or the identification of free space where the vehicle may drive, and the detection of landmarks that may be used to localize the vehicle precisely on a map.

Localization & Mapping The landmark features detected from the point cloud or the camera frames are usually matched against a pre-recorded map in order to perform

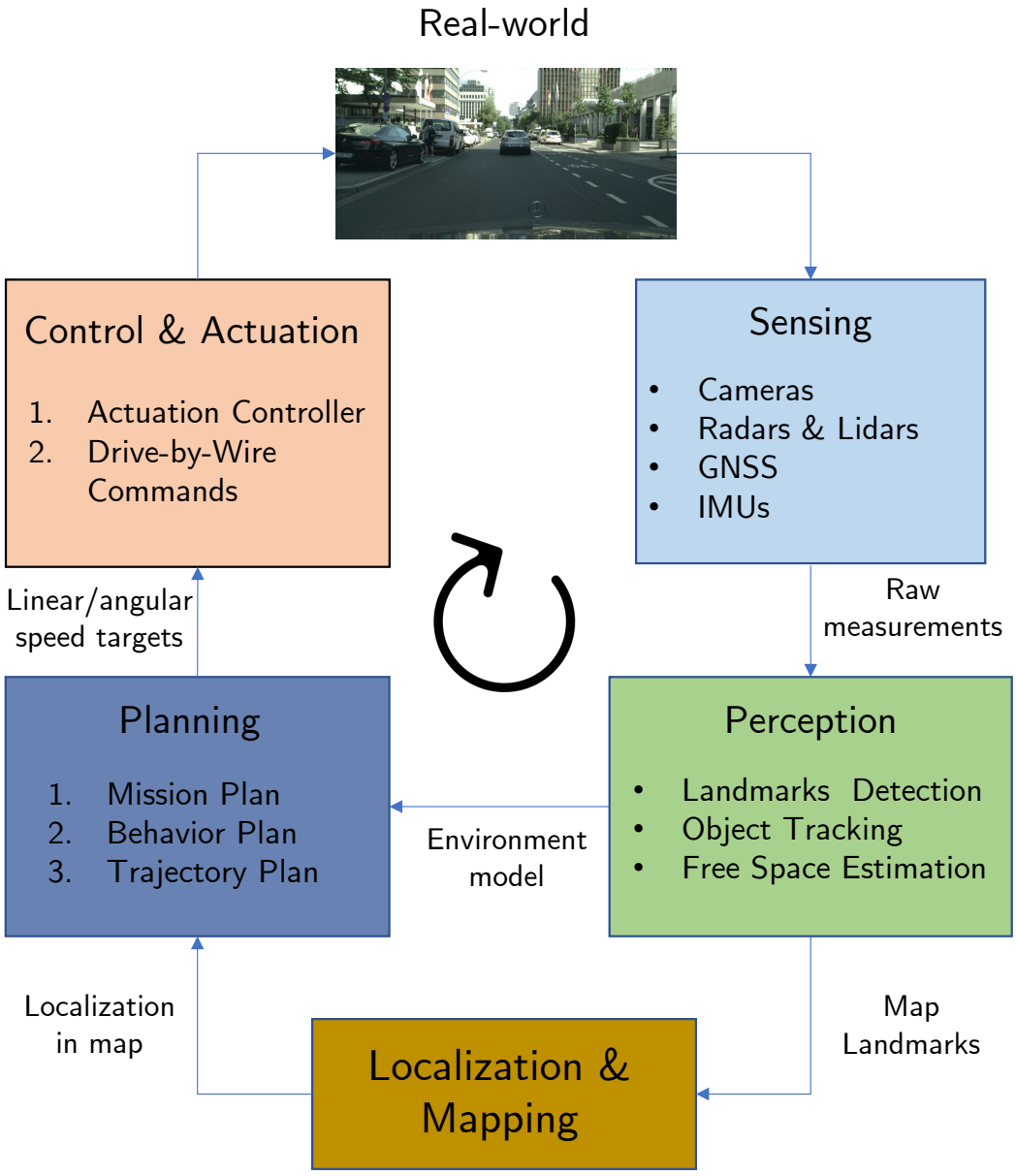


FIGURE 1.1: The processing steps of an autonomous mobile robot.

localization. When relying on a pre-existing map is not possible, another option is to build the map and localize in it at the same time, a task known as Simultaneous Localization and Mapping (SLAM) [SNS11].

Planning Mission-level planning consists in identifying the high-level steps that are needed to reach a set goal, for example which roads to take in order to reach a particular destination. This high-level plan will however not contain the details needed to navigate specific traffic situations, such as lane changing, overtaking or obstacle avoidance decisions. Making these behavior-level decisions hinges on the model of the environment built by the perception module. Finally, the trajectory planner is responsible for converting the behavior plan into a trajectory plan: a series of local positions and speeds to be reached by the vehicle over the next few seconds.

Control & Actuation: Finally, the control module uses a model of the kinematics and dynamics of the vehicle in order to decide which acceleration, steering and braking commands to apply in order to satisfy the trajectory plan. Corresponding commands are sent to the Engine Control Unit (ECU) of the vehicle over a drive-by-wire interface.

Each of the steps of the mobile robotics pipeline is rooted in an extensive body of research. In this thesis, we will focus our attention on two vision-based tasks: end-to-end control and free space perception. We will define these tasks more precisely in Section 1.5, after outlining the challenges of vision-based driving.

1.3 Challenges of Vision-Based Systems

Since humans have evolved to have a sense of sight that is highly adapted to the tasks of our everyday lives, it might be tempting to mistakenly assume that computers can easily be programmed to do the same. A popular anecdote in the community relates that solving computer vision was initially assigned as a student summer project at MIT. In fact, a team of 10 students worked on the Summer Vision Project in 1966 [Pap66]. Their ambitious goal was to design a system capable of performing foreground-background segmentation and object classification. Decades of research have since led to a lot of progress towards this goal, but solving these problems robustly for a variety of image domains remains an open research area.

Three specific categories of computer vision challenges are illustrated on Figure 1.2. First, semantically similar entities may take very different shapes. Second, since digital images are represented as 3D tensors of pixel values, changes to a frame that only slightly affect its semantic content may drastically alter its numerical representation. Finally, contextual understanding is an important part of coping with visual ambiguities. The real stake in computer vision is therefore to process raw pixel values into a different representation that efficiently encodes the semantics and disentangles them such that (1)

slight changes in the semantics result in similar small changes of the representation, and (2) changes that only affect a specific part of the semantics (*e.g.* changing the color of a vehicle) are reflected in only local changes of the representation.

(a) Object variations

(b) Illumination changes

(c) Visual ambiguities

FIGURE 1.2: Some of the challenges that vision-based autonomous systems face. The sources for these images are provided in Appendix A.

1.4 Limitations of Supervised Learning

Given that handcrafting a set of programmable rules to achieve computer vision tasks in their full generality seems impossible, researchers have instead turned to learning-based methods that infer representations from real data. The currently dominant learning methods in the field are based on artificial neural networks, and in particular Convolutional Neural Networks (CNNs). Although the backpropagation-based learning algorithm was already discussed in the 1960s [Sch14] and the CNN architecture was proposed for document recognition in 1998 [Lec+98], the last two decades saw a revolution in the field. This progress was largely made possible because of the combination of three factors.

1. The widespread availability of affordable Graphics Processing Units (GPUs) made the training of neural networks orders of magnitude faster. It also enabled researchers to build larger and more capable models.
2. Many advances were recently made in training algorithms [KB15] and network architecture design [He+16; RFB15]. The availability of software libraries for tensor manipulation and automatic differentiation have also made such design patterns easier to explore and reuse [Aba+15; Pas+19].
3. The advent of the Internet made it possible to crowd-source the curation of ever-growing annotated and raw datasets to learn from.

Although supervised learning has been responsible for many recent achievements, measuring performance using a set of minutely curated and labeled datasets may also give us a false sense of progress. There are limits on how far we can scale supervision, and ever-larger datasets will never suffice to cover the limitless situations that can be encountered at test-time. One could think that perception is easier in the context of autonomous driving, due to a more limited number of distinct object classes and strong priors on their position, but the real-world offers a wide variability. Even the a-priori simple task of detecting stop signs turns out to be challenging when the target accuracy needs to be as close as possible to 100%, as illustrated on Figure 1.3. As another illustration of the variety of situations an autonomous vehicle would need to deal with in the wild, Figure 1.4 shows a number of outliers observed in the real-world data collected by Tesla cars. The fact that labeling data by hand cannot scale forever motivates our research in both unsupervised and semi-supervised scenarios.

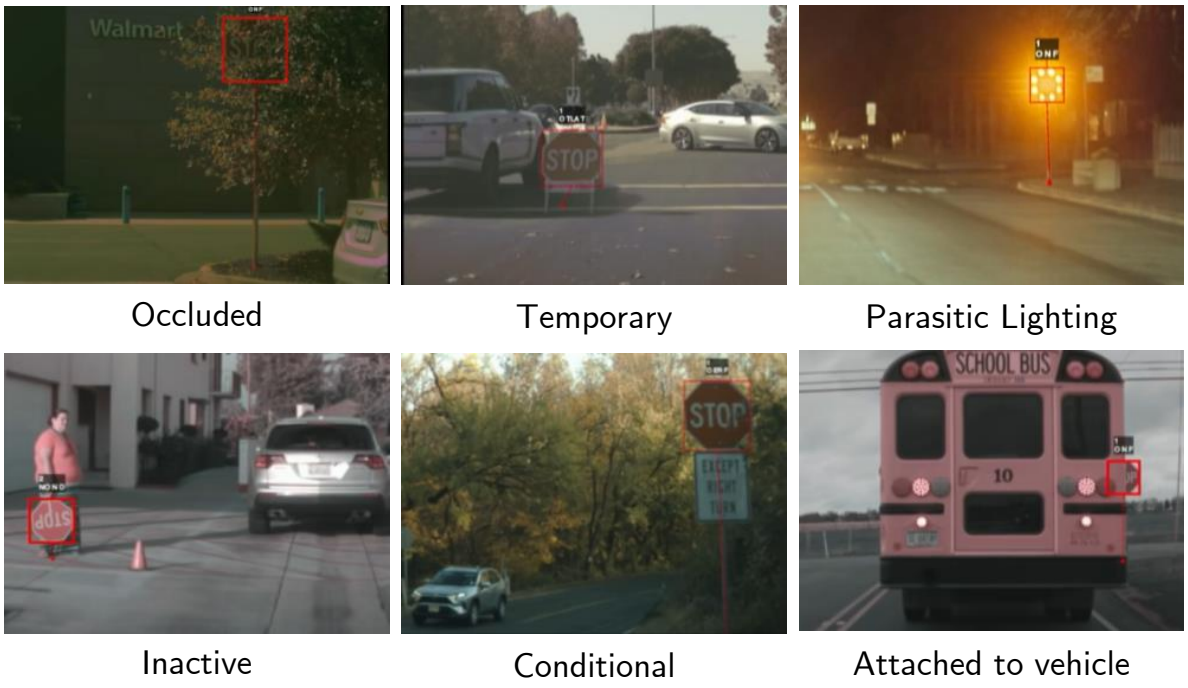


FIGURE 1.3: Even solving the simple task of stop sign detection can prove remarkably challenging when considering edge-cases. Images were obtained from a public presentation by Tesla [Kar20a].

FIGURE 1.4: Real-world cases recorded by Tesla vehicles, illustrating the long tail of the perception problem. Toppled traffic cones on the bottom right were interpreted as red lights before additional data was collected by Tesla to improve the system. These images were extracted from a public video presentation [Kar20b].

1.5 Objectives & Scope

The research presented in this thesis focuses specifically on two tasks related to autonomous driving: end-to-end steering and free space segmentation.

End-to-End Steering consists in creating a model that learns to steer a vehicle using raw camera frames as its sole test-time inputs. This task can be tackled through a form of supervised learning called imitation learning, in which a human demonstrates optimal operation of the vehicle while synchronized camera frames and steering inputs are recorded. Existing systems train to mimic the reactions of the demonstrator given only camera frames. Another option would be to exploit additional knowledge, available only at training time. This Privileged Information paradigm was first introduced by Vapnik and Izmailov [VI15], and prompts our first research question:

Research Question 1: Can imitation learning systems for end-to-end steering benefit from privileged information available at no additional labeling cost?

Free Space Estimation is the task of labeling each pixel from a front-facing camera as either traversable or occupied. As a form of binary segmentation, it has traditionally been tackled through supervised learning [OBB16]. However, a more recent work has shown promising results using unsupervised learning, in an attempt to overcome some of the

limitations of supervised learning described in Section 1.4. The most successful attempts thus far have relied on generating noisy pseudo-labels without supervision, before training models to generalize from them [Tsu+18; MUT18]. In this thesis, we explore different possibilities for improving the performance of these pseudo-supervised approaches, in an effort to answer the following research questions:

Research Question 2: Can the noise present in existing free space pseudo-labels be explicitly taken into account during training in order to improve generalization?

Research Question 3: Which data augmentation strategies are most effective for pseudo-supervised free space segmentation?

Research Question 4: Is it possible to exploit geometrical cues from approximate depth maps to generate more accurate free space pseudo-labels?

Research Question 5: Can using a fraction of ground truth free space frames result in substantial performance gains?

1.6 Organization & Contributions of the Thesis

The organization of the thesis is depicted on Figure 1.5. This section presents each chapter in more details and summarizes their contributions.

Chapter 2 presents the Deep Learning background that is needed to understand subsequent chapters. For conciseness, it reviews only information that is immediately useful in the context of this thesis. This chapter can be skipped by readers familiar with Convolutional Neural Networks and their use for end-to-end steering and semantic segmentation.

Chapter 3 explores a novel regularization technique for end-to-end vehicle steering. We show that privileged pixel-relevance information obtained with VisualBackProp can be exploited during learning to benefit performances across a variety of offline metrics. We propose Distraction Loss, a new regularization term that rewards the model for focusing on lane markings. The proposed method relies on approximate lane information, and can be easily and efficiently implemented. The work presented in Chapter 3 has led to an open-source implementation, the release of our RoboBus dataset, and the following two publications:

François Robinet, Antoine Demeules, Raphaël Frank, Georgios Varisteas, and Christian Hundt. "Leveraging Privileged Information to Limit Distraction in End-to-End Lane Following". In: *2020 IEEE 17th Annual Consumer Communications Networking Conference (CCNC)*. 2020, pp. 1–6. DOI: 10.1109/CCNC46108.2020.9045110

Georgios Varisteas, Raphaël Frank, and François Robinet. "RoboBus: A Diverse and Cross-Border Public Transport Dataset". In: *2021 IEEE International Conference on Pervasive Computing and Communications Workshops*. 2021, pp. 269–274. DOI: 10.1109/PerComWorkshops51409.2021.9431129

Chapter 4 introduces free space estimation and our evaluation protocol. We also review prior work that is relevant to the next chapters. Each one of these subsequent chapters also contains additional information on relevant work specific to that chapter. Chapter 4 ends with an overview of the concepts explored in Chapters 5 to 9, and explanations on how they fit together.

Chapter 5 presents the use of Co-Teaching to explicitly deal with pseudo-labels noise. We adapt Co-Teaching to segmentation and illustrate its effectiveness on the particular case of free space estimation. We also study the impact of the Co-Teaching schedule on performances, and propose a refinement called Stochastic Co-Teaching. Our method improves IoU results compared to standard training and traditional Co-Teaching. The chapter ends with an analysis of the limitations of the proposed approach. Our code and trained models are available online, and the following article was published in relation with the work described in this chapter:

François Robinet, Claudia Parera, Christian Hundt, and Raphaël Frank. "Weakly-Supervised Free Space Estimation Through Stochastic Co-Teaching". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*. Jan. 2022, pp. 618–627

Chapter 6 studies the impact of data augmentation on pseudo-supervised free space segmentation, and uses a recursive training scheme that progressively refines training targets. We show that an appropriate use of the Cutmix augmentation strategy results in improved results over previous efforts, gaining +2.3% in IoU, +2.4% in Precision, and +0.4% in Recall. The limitations of our simple recursive training approach are also discussed. The work presented in this Chapter was presented at the BeneLearn 2021 conference and published in the following journal article:

François Robinet and Raphaël Frank. "Refining Weakly-Supervised Free Space Estimation Through Data Augmentation and Recursive Training". In: *Artificial Intelligence and Machine Learning*. Springer International Publishing, 2022, pp. 30–45. ISBN: 978-3-030-93842-0. DOI: 10.1007/978-3-030-93842-0_2

Chapter 7 takes a different approach and investigates a novel free space pseudo-labels generation technique. We rely on a pre-trained depth estimation network to compute Road Plane Distance maps using the v-disparity algorithm. We fuse geometrical information extracted from these maps with semantical cues obtained from over-segmenting the RGB

input into superpixels in order to obtain pseudo-labels. Unlike prior work, this method does not require the use of stereo-pairs, but still reaches state-of-the-art results.

Chapter 8 builds on the results of Chapter 7 to offer a practical alternative to unsupervised free space estimation. We combine RPD maps with feature maps extracted from a pre-trained depth estimation model, and use them to train a pseudo-label generator with minimal supervision. We show that this method can greatly improve over completely unsupervised approaches, even when using as little as 1% of labeled data, which corresponds to only 29 annotated frames. The results presented in Chapters 7 and 8 have been accepted for publication in the IEEE Robotics and Automation Letters journal (RA-L), and for presentation at the IROS 2022 conference.

François Robinet, Yussef Akl, Kaleem Ullah, Farzad Nozarian, Christian Müller, and Raphaël Frank. "Striving for Less: Minimally-Supervised Pseudo-Label Generation for Monocular Road Segmentation". In: *IEEE Robotics and Automation Letters* (2022), pp. 1–7. DOI: 10.1109/LRA.2022.3193463

Chapter 1: Introduction

Part 1: Supervised Learning with Privileged Information

Chapter 2: Deep Learning Background
Reviews the concepts of Deep Learning necessary to understand subsequent chapters

Chapter 3: Enhancing End-to-End Steering with Privileged Information
Explores how CNNs can learn to steer a vehicle from human demonstrations, and how that process can be improved with a distraction loss if approximate lane information is available.

Part 2: Unsupervised Learning for Free Space Estimation

Chapter 4: Background on Learning-Based Free Space Estimation
Presents the task of free space estimation and our evaluation procedure. Also reviews prior results relevant to subsequent chapters, and outlines how these chapters fit together.

Chapter 5: Free Space Estimation through Stochastic Co-Teaching
Proposes Stochastic Co-Teaching as a new method to deal with pseudo-label noise.

Chapter 6: Data Augmentation and Self-Training for Free Space Estimation
Explores different data augmentation schemes for FSE, as well as the effect of self-training.

Chapter 7: Improving Pseudo-labels Generation for Free Space Estimation
Proposes a novel technique to generate free space pseudo-labels, combining approximate depth information with superpixel segmentation.

Part 3: Semi-Supervised Free Space Estimation

Chapter 8: Minimally-Supervised Pseudo-labels for Free Space Estimation
Builds on the pseudo-labels generation from Chapter 7 with the use of superpixel-level feature extraction and a restricted subset of labeled data at training time. This further improves results while containing labeling costs.

Chapter 9: Conclusion

FIGURE 1.5: Organization of the thesis.

Part I

Supervised Learning with Privileged Information

Chapter 2

Deep Learning Background

This chapter provides the Deep Learning background required to follow subsequent chapters. It is included in this dissertation to make it self-contained, and is meant to be a condensed introduction to the material. The chapter can be skipped entirely by readers familiar with the use of Convolutional Neural Network (CNN) for regression and image segmentation.

2.1 Parametric Learning for Image Understanding

Parametric conditional distribution learning is the most common way to approach predictions problem where a prediction has to be made from one or more input images. Many image understanding problems can be framed as learning the conditional distribution of a random variable Y given X. We denote this distribution $p(y \mid x) \stackrel{\text{def}}{=} P(Y = y \mid X = x)$. In this context, the goal of parametric learning is to model $p(y \mid x)$ as a function $f(x; w)$ parametrized by a set of weights w usually represented as a real-valued vector $w \in \mathbb{R}^m$.

As a concrete example, consider the problem of image classification depicted on Figure 2.1, where an input image has to be associated with one of C predefined classes. In this setting, X consists of RGB pixel values $x \in \mathbb{R}^{3 \times H \times W}$ for some image height H and width W, while $y \in [0, 1]^C$ is the predicted probability of the input image belonging to each class. Note that this formulation is very general and applies to many image understanding problems, including the ones treated in this dissertation. Steering angle regression and free space segmentation can be modeled by simply changing the definition of y and using an appropriate model $f(x; w)$.

2.2 Convolutional Neural Networks

As discussed in Section 1.3, the main challenge of learning-based image understanding is to transform raw pixel values into rich semantic representations. To tackle this problem, researchers have first directed their efforts to manually designing feature extractors

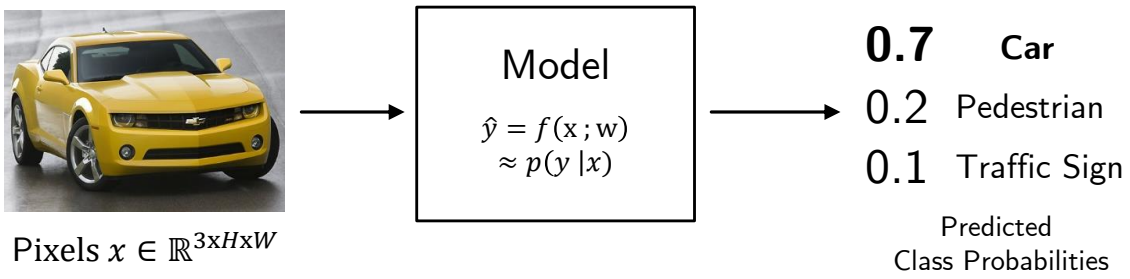


FIGURE 2.1: Toy example for image classification, illustrating parametric conditional distribution learning.

capable of capturing some semantic concepts, such as Histogram of Oriented Gradients (HoG) [DT05], Scale Invariant Feature Transform (SIFT) [Low99], or Speeded Up Robust Features (SURF) [BTVG06]. Rather than focusing on manual feature engineering, efforts in the last decade have turned to learning these automatically, by relying on an operation called "convolution".

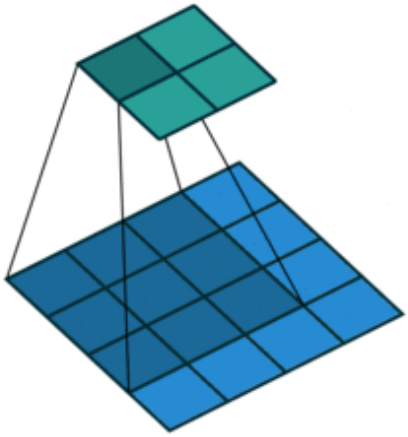
2.2.1 The Convolution Operation

The 2-dimensional convolution operation consists in sliding ("convolving") a 3-dimensional tensor k of dimensions (K_H, K_W, C_{in}) over an input x of dimensions (H, W, C_{in}). The convolved tensor is usually referred to as a "kernel". The spatial dimensions H and W of x are usually much larger than the kernel height and width (K_H, K_W), but the number of channels C_{in} (the depth) must be identical. For every spatial position of k over x, a dot product is computed between k and the subtensor of x at that position. This results in the construction of a feature map f, as illustrated on Figure 2.2a.

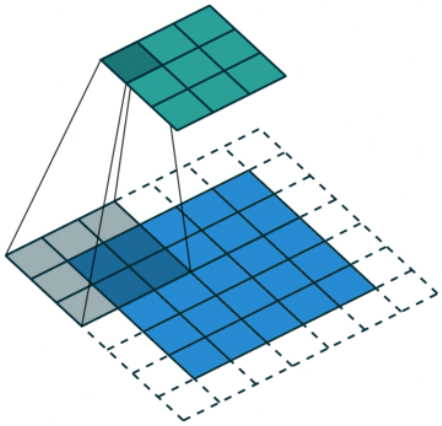
$$f(i,j) = x(i{:}i + K_H, j{:}j + K_W)^T \cdot k \tag{2.1}$$

The convolution operation has many variants. The most common ones are the addition of padding on the input, and the use of a stride, *i.e.* the distance between each kernel positions. Figure 2.2b displays an example of padding and stride being used. Given an input of spatial shape (H, W), a kernel of spatial shape (K_H, K_W), a total padding of P and a stride of S, the feature map resulting from convolving the kernel over the input has a spatial shape (F_H, F_W) given by Equation 2.2.

$$F_H = \left\lfloor \frac{H - K_H - P}{S} \right\rfloor + 1, \quad F_W = \left\lfloor \frac{W - K_W - P}{S} \right\rfloor + 1 \tag{2.2}$$



(A) 3×3 kernel, no padding, stride 1.



(B) 3×3 kernel, padding, stride 2.

FIGURE 2.2: Visualization of the convolution operation. The input is depicted in light blue, the kernel as a shade over the input, and the output feature map is represented in green. Images by Dumoulin and Visin under MIT license [DV16].

Note that the operation presented here and referred to as "convolution" by the computer vision community is in fact more precisely named "cross-correlation" as it is not strictly equivalent to the mathematical definition of the convolution [GBC16]. We will nevertheless follow the literature and refer to it as convolution.

Computing feature maps using Equation 2.1 produces interesting results for particular values of the kernel. Some examples are illustrated on Figure 2.3. The convolution operation allows to compute many interesting feature maps depending on the values of the kernel. Rather than attempting to manually identify and compose relevant kernels for each application, CNNs aim to treat the kernel values as learnable model parameters.

2.2.2 The Convolutional Layer

Since convolution kernels can be seen as learned feature extractors, it is natural to use them as part of a building block for deep network design. A single convolutional layer consists of an arbitrary number of kernels of the same size that are independently convolved over the input. The resulting feature maps are stacked, producing a feature maps tensor of shape (C_{out}, F_H, F_W) where C_{out} is the number of kernels in the layer, and F_H and F_W are computed using Equation 2.2. A bias term is then added to each output. This process is illustrated on Figure 2.4.

Because convolution is a linear operator, stacking convolutional layers (without bias) can only result in a linear transformation. For this reason, non-linear activation functions are traditionally used after each convolution. The most popular choices for hidden layers include the Rectified Linear Unit (ReLU), the hyperbolic tangent (tanh), the Exponential Linear Unit (ELU), or the Sigmoid Linear Unit (SiLU). These activation functions are

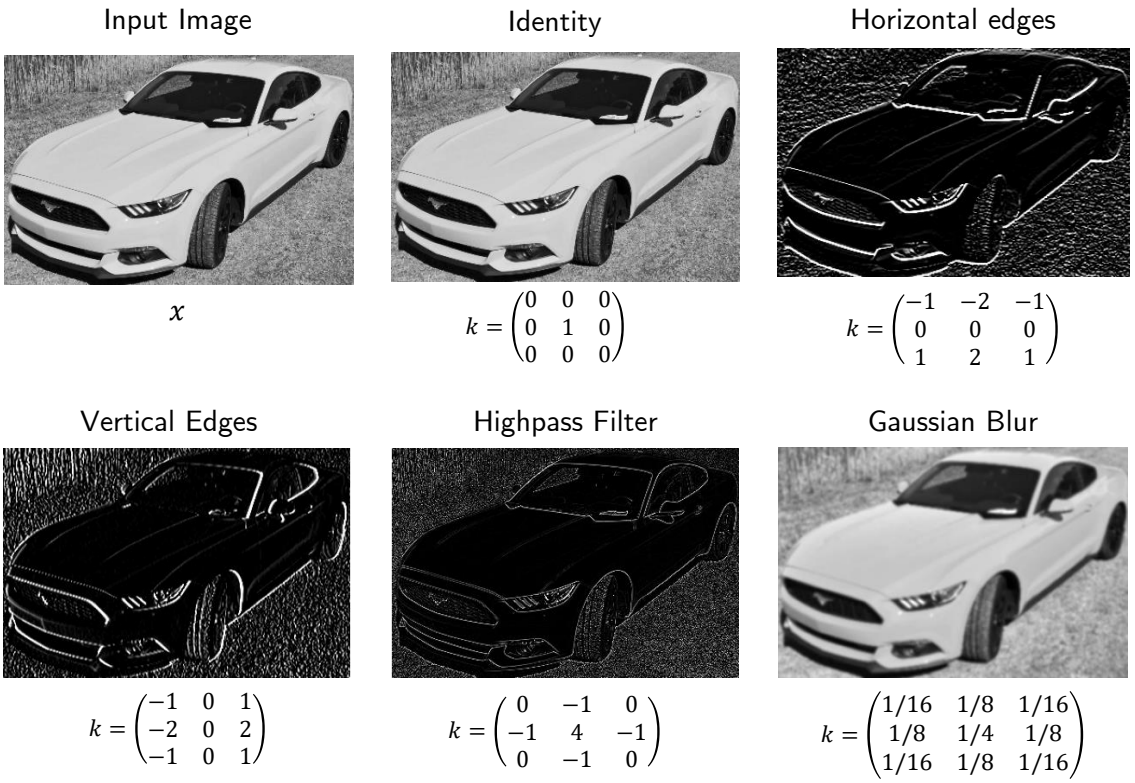


FIGURE 2.3: Feature maps obtained by convolving different 3×3 kernels over the same input image.

depicted on Figure 2.5. An activation can also be present on the final layer of a network, usually to scale the output to some known range or to normalize its components. The softmax function is commonly used for this purpose in classification or segmentation problems:

$$\text{softmax}(z)_i = \frac{e^{z_i}}{\sum_j e^{z_j}} \tag{2.3}$$

2.2.3 Assembling Convolutions: PilotNet

As an example of how convolutional layers can be assembled into a deep network, we will examine the PilotNet architecture. PilotNet has been proposed to tackle the end-to-end vehicle steering problem [Boj+16]. The problem consists in learning to predict human-like steering angles from road-facing camera frames. The PilotNet architecture is illustrated on Figure 2.6. A series of five convolutional layers is followed by three fully-connected layers, in order to compute the steering angle output. A more in-depth treatment of end-to-end steering will be the topic of Chapter 3.

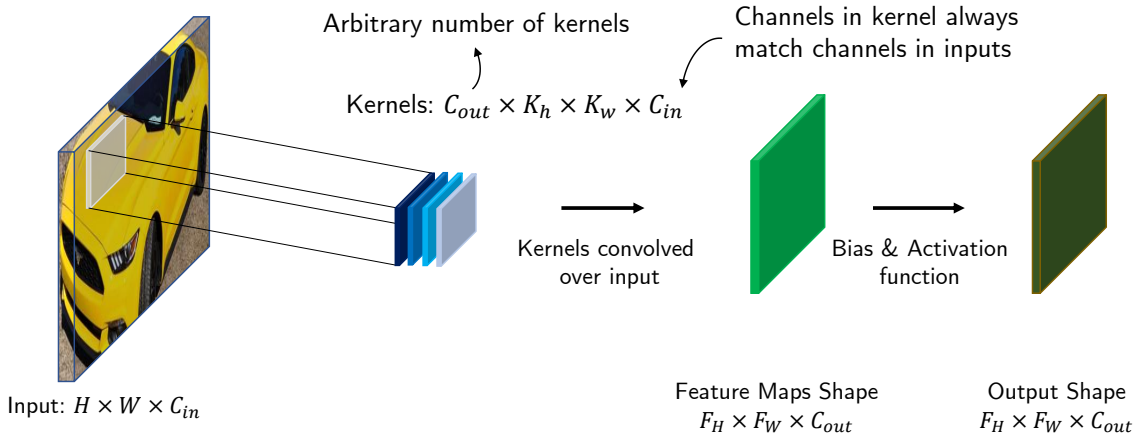


FIGURE 2.4: The convolutional layer, the basic building block of CNNs.

2.3 Loss Functions

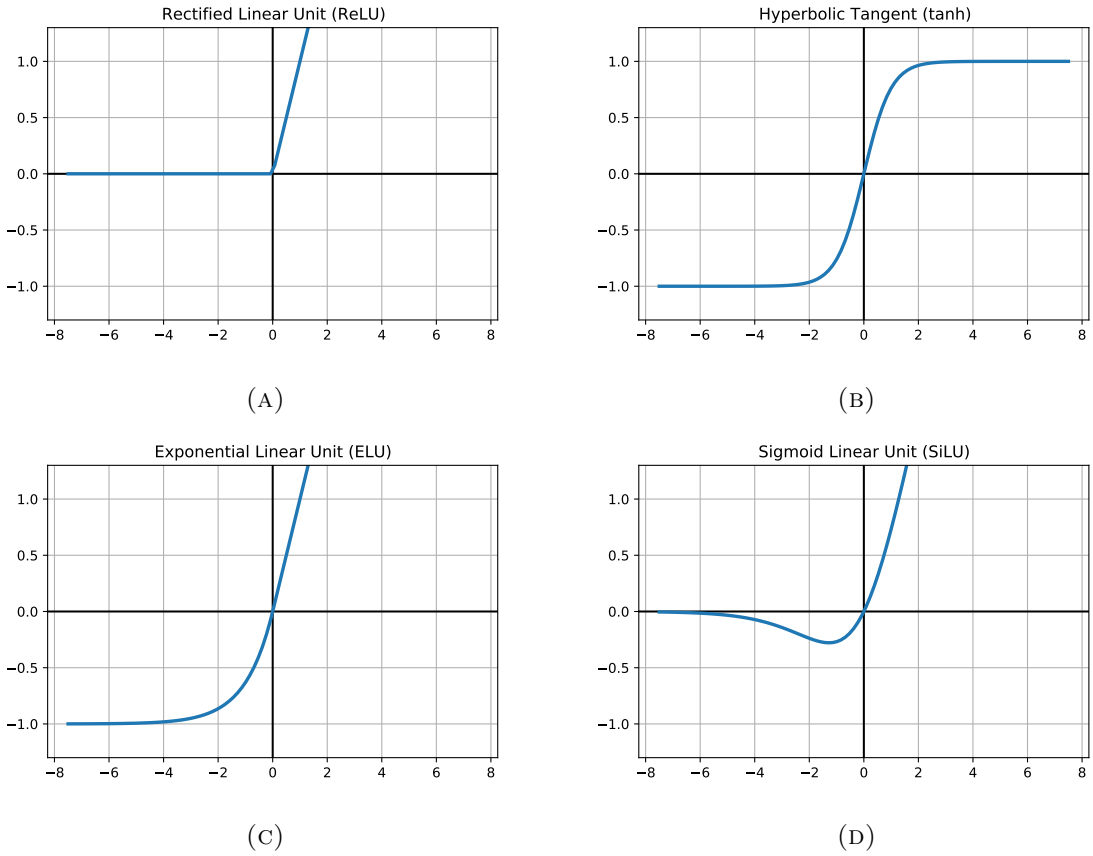
In the context of supervised learning, learning is done using a training dataset $\mathcal{D}_{train} = \left\{(x^{(0)}, y^{(0)}), \ldots, (x^{(n)}, y^{(n)})\right\}$. The samples of $\mathcal{D}_{train}$ are assumed to be independently drawn from the data distribution $p(x, y)$. The expected performance of the trained model on $p(x, y)$ can be approximated after training using a separate, independently drawn test dataset $\mathcal{D}_{test}$.

In order to select the best parameters w for a given task, we must define a criterion for evaluating a set of parameters during training. This is done by choosing a loss function L, comparing the expected outputs with the predicted outputs. Ideally, we should select the best parameter values w^* by minimizing the expected value of L over the true data distribution $p(x, y)$.

$$w^* = \underset{w \in \mathbb{R}^m}{\operatorname{argmin}} \, \mathbb{E}_{(x,y)\sim p} \, L\left(y, f\left(x; w\right)\right) \tag{2.4}$$

Since the underlying data distribution is unknown, we actually minimize the empirical loss computed from the training dataset $\mathcal{D}_{train}$, assuming the training samples were independently drawn.

$$w^* = \underset{w \in \mathbb{R}^m}{\operatorname{argmin}} \, \mathbb{E}_{(x,y)\in\mathcal{D}_{train}} \, L\left(y, f\left(x; w\right)\right) = \underset{w \in \mathbb{R}^m}{\operatorname{argmin}} \prod_{i=0}^{n} L\left(y^{(i)}, f\left(x^{(i)}; w\right)\right) \tag{2.5}$$



(A) (B) (C) (D)

FIGURE 2.5: Common activation functions in hidden layers.

2.3.1 Classification Losses

The choice of the per-sample loss function depends on the application, and is also guided by optimization principles. We will consider classification problems first. In theory, one may want to optimize directly for accuracy, by using a 0-1 loss, which equals zero when the prediction is correct and one otherwise. Optimizing such a 0-1 loss is however NP-hard [NS13], and thus intractable in the context of deep learning, where millions of parameters often have to be learned. The most popular loss for classification problems can be derived by applying the principles of Maximum Likelihood Estimation (MLE). MLE optimizes the weights in order to maximize the likelihood that the model generates the observed data [GBC16]. Using $p_m(y \mid x; w)$ to denote the probability that a model with weights w assigns to class y given input x, we can write

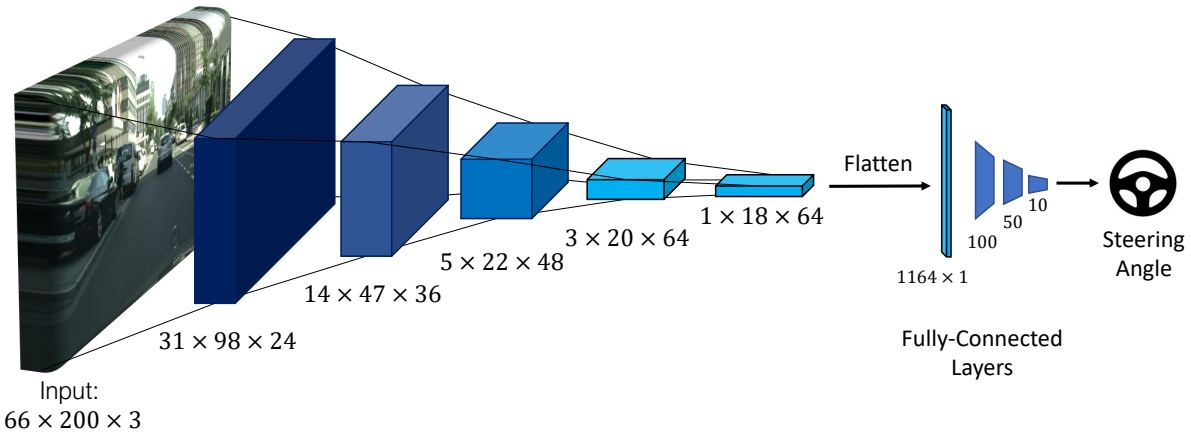


FIGURE 2.6: The PilotNet architecture.

$$
\begin{aligned}
w^* &= \underset{w \in \mathbb{R}^m}{\operatorname{argmax}}\ p_m\left(y^{(0)}, \ldots, y^{(n)} \mid x^{(0)}, \ldots, x^{(n)}; w\right) \\
&= \underset{w \in \mathbb{R}^m}{\operatorname{argmax}} \prod_{i=0}^{n} p_m\left(y^{(i)} \mid x^{(i)}; w\right) \\
&= \underset{w \in \mathbb{R}^m}{\operatorname{argmax}} \sum_{i=0}^{n} \log p_m\left(y^{(i)} \mid x^{(i)}; w\right) \\
w^* &= \underset{w \in \mathbb{R}^m}{\operatorname{argmin}} -\sum_{i=0}^{n} \log p_m\left(y^{(i)} \mid x^{(i)}; w\right)
\end{aligned} \tag{2.6}
$$

Minimizing the right-hand term from Equation 2.6 is equivalent to minimizing the cross-entropy between the training data distribution and p_m, which is why this loss is often referred to as the "cross-entropy loss" [GBC16].

2.3.2 Regression Losses

For regression problems, such as the end-to-end steering task described in Chapter 3, two common losses are the Mean Absolute Error (MAE) and Mean Squared Error (MSE). These two losses are also referred to as L_1 and L_2 losses and are respectively computed using Equations 2.7 and 2.8.

$$w^* = \sum_{i=0}^{n} \left|\left| f(x^{(i)}; w) - y^{(i)} \right|\right|_1 \tag{2.7}$$

$$w^* = \sum_{i=0}^{n} \left|\left| f(x^{(i)}; w) - y^{(i)} \right|\right|_2^2 \tag{2.8}$$

2.4 Learning Parameters

The commonly used losses and layers, including the ones presented in the previous sections, are (sub-)differentiable. This means that we can leverage gradient-based optimization techniques to search for w^*. In its simplest form, Gradient Descent (GD) iteratively optimizes an estimate of the weights starting from an initial guess $w^{(0)}$ and following the direction of steepest descent opposite to the gradient.

$$w^{(k)} = w^{(k-1)} - \frac{\eta}{n} \sum_{i=0}^{n} \frac{\partial L}{\partial w} \left(f(x^{(i)}; w), y^{(i)} \right) \tag{2.9}$$

The value of η is a hyper-parameter called the learning rate, and controls the magnitude of each optimization step. Computing each iteration of Equation 2.9 requires to evaluate the loss gradient over every sample of the training data, which makes it slow when working with large datasets. Note that since our objective is the minimization of the loss function on the underlying data distribution and not the training set, the gradients of Equation 2.9 are only approximations. One way to speed up the computation is therefore to replace the training set gradient by a gradient computed on a smaller "batch" of training samples. This leads to Stochastic Gradient Descent (SGD), which uses a random batch of training data $B \subseteq \mathcal{D}_{train}$ at every iteration.

$$w^{(k)} = w^{(k-1)} - \frac{\eta}{|B|} \sum_{(x,y) \in B} \frac{\partial L}{\partial w} \left(f(x; w), y \right) \tag{2.10}$$

Many refinements to SGD have been proposed over the years, including the popular Adaptive Moment Estimation (Adam) [KB15]. The main goal of these refinements is to automatically adjust the learning rate η during training, by incorporating loss-landscape information collected over the previous iterations.

2.5 The U-Net Architecture for Image Segmentation

As a segmentation task, supervised free space estimation has directly benefited from progress in semantic segmentation. Pixel-level prediction carries a crucial challenge for network design: an optimal prediction can only be achieved by combining fine-grained local information with global contextual cues.

Some architectures, such as Fully Convolutional Networks (FCNs) carry these cues in their encoder-decoder architectures [LSD15]. Building on a similar idea, the U-Net architecture combines entire encoder feature maps with decoder features at each step of the expansion path of the network [RFB15]. This is illustrated on Figure 2.7.

U-Nets have attracted a lot of attention in recent years, and researchers have proposed refinements such as the use of dense connections [Jég+17] and dilated convolutions [Zha+17], the integration of attention mechanisms [Okt+18], or extensions to volumetric images [MNA16]. In subsequent chapters, we will rely on a simple U-Net architecture. Our choice is motivated by a recent finding that many recent architecture improvements are outperformed by a well-tuned vanilla U-Net [Ise+18].

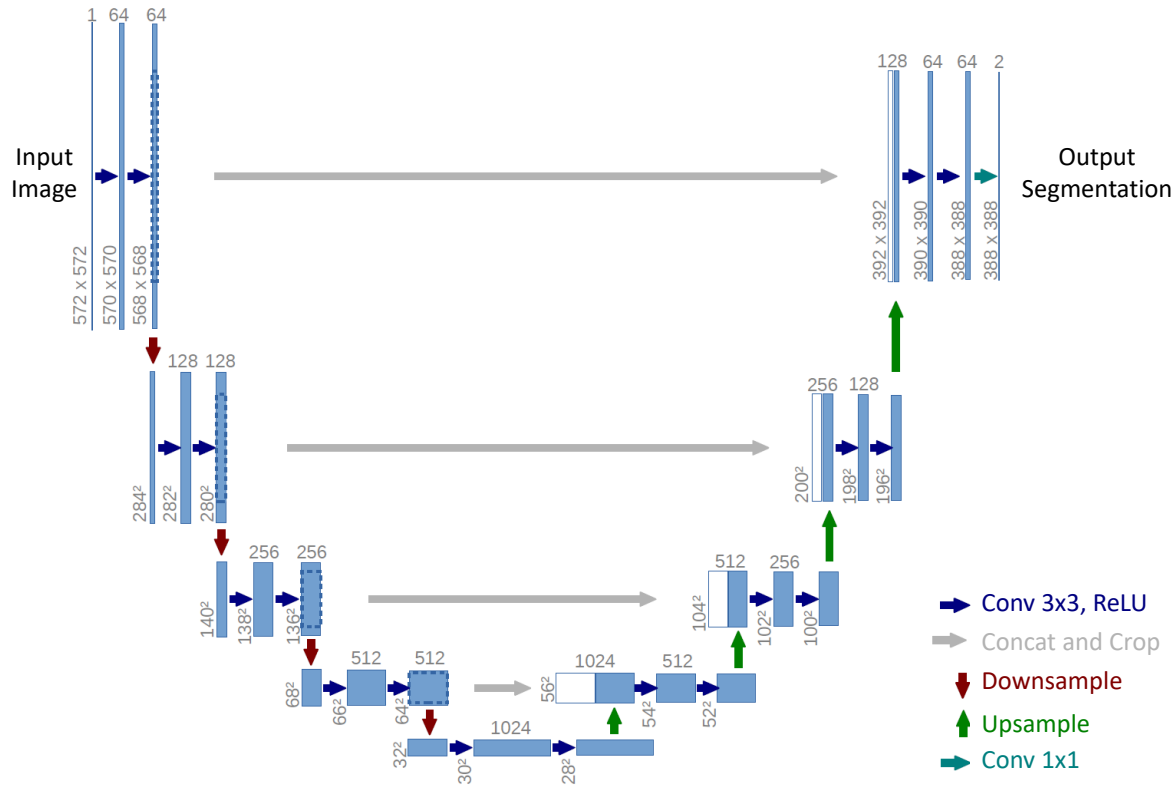


FIGURE 2.7: An example of the U-Net Architecture, reproduced from [RFB15]. Feature maps are gradually downsampled along the encoder path (left) and the decoder path (right) gradually upsamples encoded features.

Chapter 3

Enhancing End-to-End Steering with Privileged Information

3.1 Introduction

In recent years, imitation learning has successfully been applied to learn complex control skills directly from sensor data. One application is end-to-end steering, where steering angles are learned from road facing camera frames recorded during human expert demonstrations [Boj+16]. At the same time, different methods have emerged to visualize how neural networks learn embeddings and make decisions. In particular, a class of these visualization methods generates heatmaps identifying which input pixels are most relevant to determine the output of the model [Bin+16; Boj+18].

In this chapter, we show that pixel-relevance heatmaps for Convolutional Neural Networks (CNNs) can be leveraged during training to improve end-to-end lane following. To achieve this, we propose an ad-hoc loss function that improves the trained model without incurring any noticeable computational overhead. Our loss function includes a regularizer that encourages the model to focus on road markings, which we assume to be relevant for the lane following task. In relation to this work, we also publicly release the RoboBus dataset, which consists of extensive driving data recorded using a commercial bus on a cross-border public transport route at the Luxembourgish-French border.

The work presented in this chapter is based on the following publications:

François Robinet, Antoine Demeules, Raphaël Frank, Georgios Varisteas, and Christian Hundt. "Leveraging Privileged Information to Limit Distraction in End-to-End Lane Following". In: *2020 IEEE 17th Annual Consumer Communications Networking Conference (CCNC)*. 2020, pp. 1–6. DOI: 10.1109/CCNC46108.2020.9045110

Georgios Varisteas, Raphaël Frank, and François Robinet. "RoboBus: A Diverse and Cross-Border Public Transport Dataset". In: *2021 IEEE International Conference on Pervasive Computing and Communications Workshops*. 2021, pp. 269–274. DOI: 10.1109/PerComWorkshops51409.2021.9431129

3.2 Related Work

End-to-end control systems based on neural networks have been proposed in the past. The first attempt, named ALVINN (Autonomous Land Vehicle In a Neural Network) was published three decades ago [Pom89]. The system consisted of a 3-layer back-propagation network designed for the task of lane following. Although it used a simplistic network architecture, and a relatively small synthetic training set, ALVINN was able to accurately drive a path of 400 meters at a speed of half a meter per second.

It took 15 years further until another groundwork relying on the same principles was proposed as part of a DARPA challenge. The project, known as DAVE (DARPA Autonomous Vehicle) [LeC+05], was built around a small-scale radio-controlled car equipped with two cameras. The training set included hours of video data and the steering commands of a human operator. The trained CNN model was able to successfully navigate the vehicle through a previously unseen open terrain while avoiding obstacles.

With the advent of deep learning and large-scale parallel accelerators, the approach has recently been revisited by researchers from NVIDIA [Boj+16]. The system, dubbed PilotNet, employs three cameras and a steering angle recorder to generate a training set from human demonstration. The training data is fed to a deep CNN in order to obtain a model that predicts the correct steering angle from solely one input image. They empirically demonstrate that the CNN is able to learn the task of lane and road following without manual decomposition into human crafted features such as roads or lanes.

In a recent work, we successfully reproduced the results of the NVIDIA paper with our own experimental platform [Var+19]. We made use of the VisualBackProp method [Boj+18] to highlight pixel regions in the input image that contribute most to the prediction of the CNN. We concluded that the network tends to independently learn features such as road boundaries and lanes markings, but vast amounts of diverse data are needed in order to capture changing light and weather conditions that severely impact the performance of the model.

Another approach to implement an autonomous driving software stack is to initially understand the traffic scene before computing the control signals. This is typically achieved by extracting semantics from the raw pixels to detect lanes, traffic signs and other vehicles [Nev+18; Tei+16]. This information is then used by a trajectory planner to compute the low-level control signals.

The idea of this work is to take advantage of the spatial information obtained through semantic segmentation to improve the performance of an end-to-end lane following model. This approach is inspired by the work of Bisla and Choromanska [BC18], who specified a methodology for incorporating privileged information, such as available segmentation masks along with classification labels, into the training stage of a CNN. They rely on the VisualBackProp technique [Boj+18] to guide the network towards specific parts of the input image when forming the prediction. They tested the performance of their approach to classify dermoscopic images to detect skin lesions. Empirically, they find that their

method outperforms common baselines on a variety of learning problems. The use of additional knowledge, available only at training time, is described by the Learning Under Privileged Information (LUPI) paradigm, introduced by Vapnik and Izmailov [VI15].

We apply a similar methodology to enhance the performance of an end-to-end lane following network by incorporating privileged information about the lane geometry to improve the learning process. To the best of our knowledge, such an approach has never been applied in this context before.

We also publicly release our RoboBus dataset, which includes extensive driving data recorded by a public transport bus over four days on a trip at the border between Luxembourg and France. The 8 hours of driving data are divided into 15 trips and include over 1.7 million anonymized images captured by two road-facing cameras, as well as GNSS traces for precise localization, 9-axis IMU measurements, and diagnostics data extracted from the CAN interface. This CAN data includes speed, steering angle, and position of the acceleration and brake pedals, which are important factors for understanding driver intent. The complete dataset size reaches 67GB and is available for download on GitHub [Rob].

3.3 Methodology

Our method attempts to guide the training process by imposing the prior belief that a good lane keeping model should focus a major fraction of its attention on lane markings. To model this prior, we propose a new loss function, which we coin *Distraction Loss D*.

$$D(H, M) = 1 - \frac{||H \odot M||_1}{||H||_1}. \tag{3.1}$$

In Equation 3.1, H denotes a relevance heatmap obtained using VisualBackProp [Boj+18], M is the ground truth lane marking binary segmentation mask, and $\odot$ stands for the component-wise product (also known as the Hadamard product). The numerator of the quotient represents the $L1$-norm of the intersection between the relevance heatmap and the lane markings. The Distraction Loss is thus measuring the fraction of the model focus that is spent outside of lane markings.

Our model is trained using a joint loss that balances the standard Mean Squared Error (MSE) on predicted steering angles with a Distraction Loss term acting as a regularizer.

$$L = ||\hat{y} - y||_2^2 + \lambda D(H, M). \tag{3.2}$$

Because $M \in \{0, 1\}^{h \times w}$, the distraction loss has the useful property of being bounded to $[0, 1]$, which allows complete control over the magnitude of the regularization term by

tuning $\lambda \in \mathcal{R}^+$. Note that setting $\lambda = 0$ disables any distraction penalty.

Although the Distraction Loss expression from 3.1 closely resembles the traditional Intersection over Union metric (IoU), it does not have the effect of forcing the model to spread its focus over the whole lane markings. We argue that such penalty would be unnecessarily restrictive in the context of lane following, because small fractions of lane markings carry enough visual cues to understand the shape of the whole lane. On the other hand, in the presence of high λ values, the model might overfit the regularizing term by focusing on very small parts of the ground truth lanes. The results presented in Section 3.5 show that this can indeed happen in practice.

VisualBackProp has been mathematically proven to appropriately estimate pixel relevance in the general case of convolutional networks [Boj+18]. Although it can be extended for networks that include residual connections [BC18; He+16], this work focuses on the regularization process and therefore relies on a traditional architecture. In particular, we chose to reproduce the PilotNet architecture [Boj+16].

The implementation of VisualBackProp is illustrated on Figure 3.1. Given that all its operations are differentiable and use constant weights, this method can be embedded in our architecture without adding any trainable weights or noticeable computational overhead.

Figure 3.2 presents an overview of the complete architecture. During training, a forward pass proceeds as follows: a batch of camera frames is fed through a sequence of convolutional layers to generate stacks of feature maps. The feature maps of the last layers are forwarded through a sequence of dense layers to predict the steering angle. In the second branch of the network, all convolutional feature maps are used by VisualBackProp to generate heatmaps H. These heatmaps highlight the pixels which are most relevant to the network's learned representation. The joint loss is then computed as described in Equation 3.2. During backpropagation, the gradient flows through both the VisualBackProp and angle prediction branches. The learned representation of the model is therefore adjusted to increase focus on lane markings while correctly predicting steering angles. Since VisualBackProp does not add any trainable parameters to the model, it acts as a constant operation with respect to the computation of the gradient. Moreover, generating the pixel-relevance heatmaps during the same forward pass as the steering angles allows for a computationally efficient training procedure.

Under the LUPI framework, we are only using relevance heatmaps to guide the learning process. They can be discarded once training is over. During inference, this shrinks the model to only the first row of Figure 3.2.

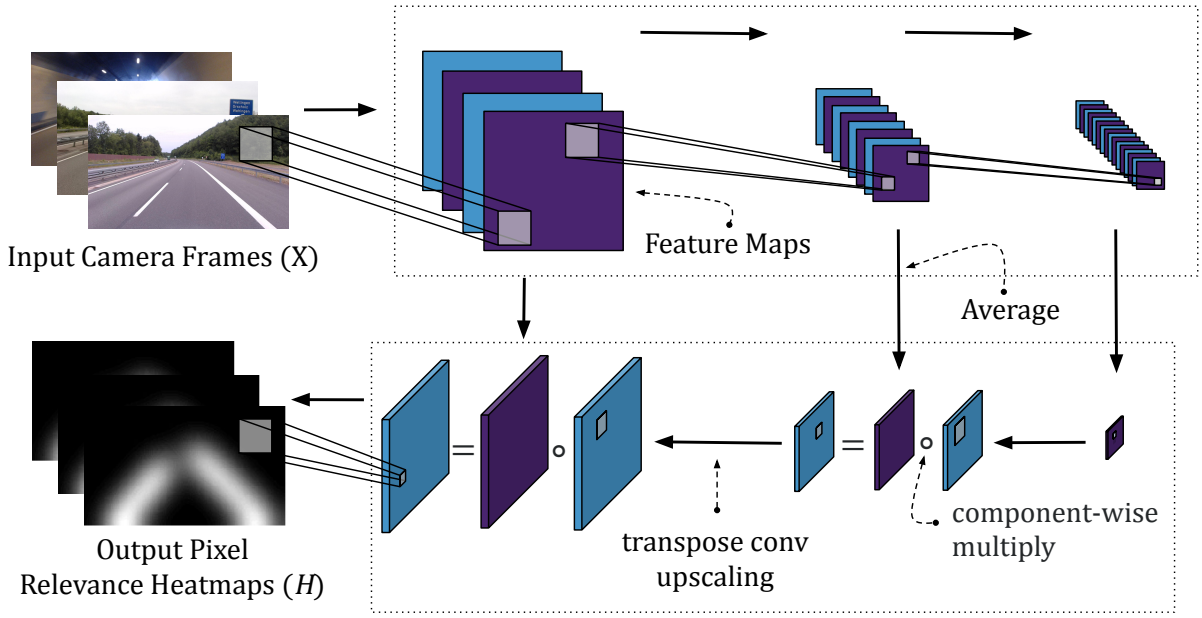


FIGURE 3.1: The VisualBackProp method for generating pixel relevance heatmaps for 3 convolutional layers [Boj+18]. The top-right block encompasses all convolutional layers, while the VisualBackProp operations are represented in the bottom-right block. The arrows indicate the flow of a forward pass.

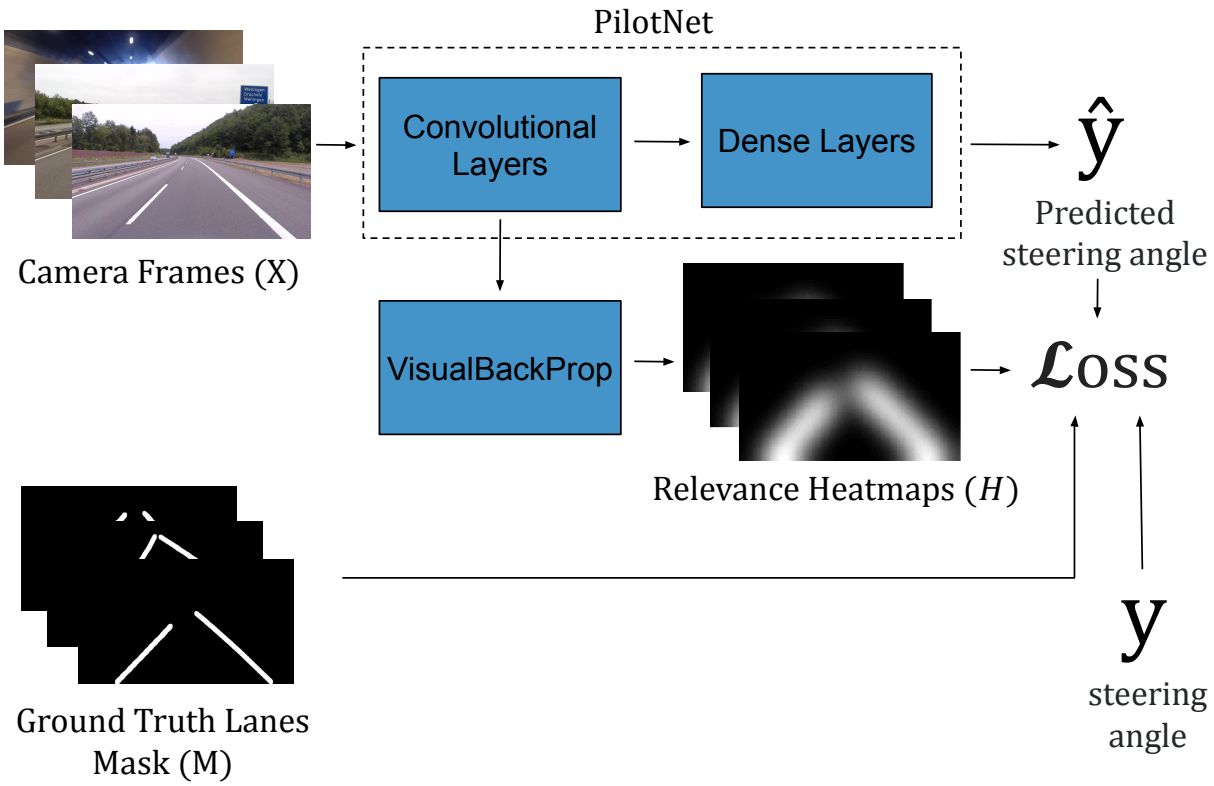


FIGURE 3.2: Complete architecture overview. The network extends PilotNet with a VisualBackProp branch whose output is used to compute a joint loss.

3.4 Experimental Setup

3.4.1 Dataset

We collected data on Luxembourgish highways using an off-the-shelf RGB camera mounted at the front of our experimental vehicle, recording in 720 × 1280 resolution at 30 frames per second. The recording occurred over multiple days to capture a broad variety of daytime weather conditions as well as different road portions, spanning a total of 2 hours of driving. The ground truth steering angles were also read at 30 Hz from the vehicle's CAN bus. In the context of our lane following experiment, the recorded data contains no lane changes, but does include curvy road sections as well as highway exits.

In order to include the distraction loss term from Equation 3.1 into the loss, ground truth lane marking segmentation is required. In the absence of annotated lanes in our dataset, we used a LaneNet model that was pre-trained on the TuSimple dataset [Nev+18; Tus]. This yielded visually accurate lane masks on our data. The results from Section 3.5 show that the training process can extract useful information even from such imperfect lane marking ground truth.

In the machine learning literature, test data is commonly obtained by uniformly sampling a fraction of the available data. However, in the case of end-to-end steering, there exists a high correlation between temporally close frames. Using the classical sampling strategy would leak information from the training set into the test set, and yield a poor estimate of the true capabilities of trained models. Instead, we form a test set by selecting whole stretches of highways never seen in the training data, accounting for 20% of the total data. The test clips are selected randomly while ensuring that the steering angle distribution stays similar to the one from the training data.

The preprocessing steps are straightforward. Similarly to the original PilotNet, the top 150 pixels corresponding to the sky are cropped. Input frames are then normalized and the result is resized using Lanczos interpolation [Duc79], producing a final 66x200 input resolution.

3.4.2 Evaluation

The evaluation of end-to-end steering models is a complex open problem. Ideally, these models should be evaluated in the real-world by running them in one or more test vehicles. In this online evaluation context, several performance metrics have been proposed, such as the fraction of time that the model is able to safely and autonomously steer the vehicle, or the average time between human interventions [Boj+16; Cod+18].

However, online evaluation schemes are often not practical for academic labs, which have to resort to evaluating models offline, using an annotated test dataset collected in

Name	Per Sample Expression
Mean Squared Error	$(\hat{y}_i - y_i)^2$
Absolute Error	$\lvert\hat{y}_i - y_i\rvert$
Quantized Classification Error	$1 - \delta\left(Q_\sigma(\hat{y}_i)), Q_\sigma(y_i)\right)$

TABLE 3.1: Per sample arithmetic expressions for the metrics used to evaluate our models on the test set. A prediction is denoted with $\hat{y}_i$, Q_σ is a quantization function and $\delta(i,j) = \mathbb{1}_{ij}$ is Kronecker's delta. A precise definition of Q_σ is given in the text.

conditions similar to the training set. Models are usually trained by minimizing the MSE of steering angle predictions, but can be evaluated using other metrics. Codevilla et al. have proposed different options and used the CARLA simulator [Dos+17] to show that some of them correlate better with online driving abilities than the angle MSE [Cod+18].

The metrics used to evaluate our models are presented in Table 3.1. Evaluating with MSE is known to overemphasize outliers since it heavily penalizes the largest errors compared to the absolute error. The quantized classification error introduces the idea that driving performances do not really depend on exactly predicting the steering angle. Instead, it is more important to predict the general direction correctly. Using a threshold σ, a quantization function Q_σ classifies steering angles as going left, right or straight. The classification error is then computed using the quantized angle classes and subsequently used to measure the model's abilities.

$$Q_\sigma(x) = \begin{cases} 1 & \text{if } x > \sigma \text{ (right)} \\ 0 & \text{if } -\sigma \leq x \leq \sigma \text{ (straight)} \\ -1 & \text{if } x < -\sigma \text{ (left)} \end{cases}$$

3.5 Results

We train many models to minimize the joint loss introduced in Equation 3.2. The training is run over 70 epochs on batches of 256 examples, using the Adam optimizer [KB15]. The learning rate is initially set to 0.001 and decayed to 0.0005 after 40 epochs. In order to account for the stochasticity of the learning process, 20 independent training runs are considered for each value of λ. Let $\mathcal{M}_\lambda^i$ denote the ith model trained with regularization coefficient λ. For any given training run $i \in \{0, ..., 20\}$, all models $\mathcal{M}_\lambda^i$ are trained on the same batches of data in the same order. Note that we present results obtained for different arbitrary values of λ, without tuning it to the test set. Determining an optimal value for λ and its expected generalization error would

Error Type		mean	std	best run
Mean Squared Error	$\lambda = 0$	0.3961	0.0228	0.3503
	$\lambda = 1$	0.3494	**0.0218**	0.3221
	$\lambda = 5$	**0.3331**	0.0228	**0.2920**
	$\lambda = 25$	0.4741	0.0339	0.4283
Mean Absolute Error	$\lambda = 0$	0.4431	0.0249	0.3987
	$\lambda = 1$	0.4111	**0.0103**	0.3945
	$\lambda = 5$	**0.3966**	0.0109	**0.3769**
	$\lambda = 25$	0.4956	0.0196	0.4752
Quantized Classification Error	$\lambda = 0$	0.1853	0.0097	0.1694
	$\lambda = 1$	0.1676	**0.0061**	0.1538
	$\lambda = 5$	**0.1634**	0.0067	**0.1519**
	$\lambda = 25$	0.1927	0.0117	0.1735

TABLE 3.2: Aggregated test results for each metric and different regularization strengths. For each value of λ, 20 models are trained over 70 epochs to minimize Equation 3.2.

In Figure 3.3, we investigate the effect of increasing λ on the Distraction Loss D. Unsurprisingly, larger λ values result in a steeper decrease of D. Interestingly, D settles to the same range of values for all $\lambda > 0$. This indicates that all models eventually prefer to pay the cost of sometimes looking outside of lane markings in order to further decrease their steering errors.

Table 3.2 presents statistics of the performance distributions obtained for all trained models $\mathcal{M}_\lambda^i$. Using the joint loss with reasonable λ values yields performance improvements on average, but also results in the best models overall. Guiding the training with lane geometry information also benefits the stability of the learning process, as shown by smaller standard deviations across different runs.

Lastly, Figure 3.4 illustrates the impact of tweaking λ on both VisualBackProp heatmaps and predicted steering angles. Note that lane markings emerge as relatively prominent centers of attention even when $\lambda = 0$. This result was already observed by Bojarski et al. [Boj+17]. As expected, increasing λ results in models placing a sharper focus on these regions of the input. Although this can help models to avoid distractions such as overtaking vehicles, λ values that are too high also end up increasing steering errors. This is the case in the second and third rows of the figure, where $|\Delta y|$ grows larger when $\lambda = 10$.

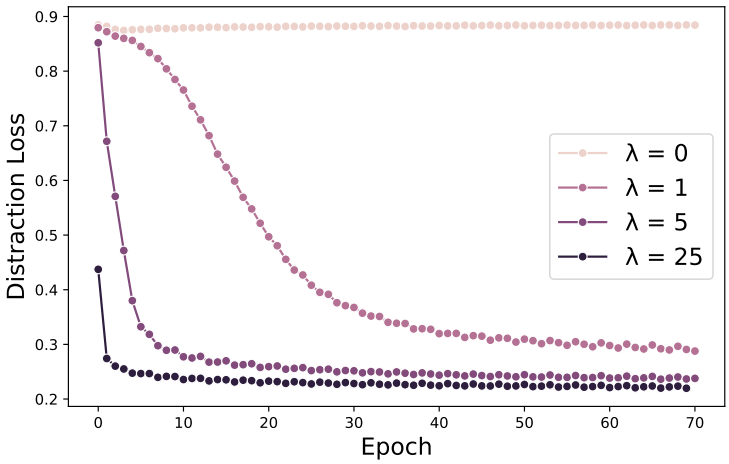


FIGURE 3.3: Training learning curves for the Distraction Loss term from Equation 3.2. Each experiment is run 20 times for each value of λ and mean curves are reported.

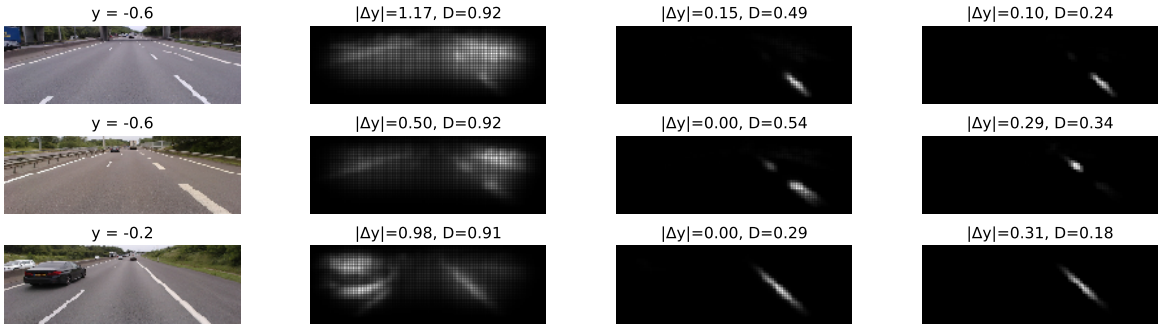


FIGURE 3.4: Relevance heatmaps obtained for selected input frames and for different values of λ. The leftmost column shows the rescaled input that is fed to the model, and the three columns to its right respectively correspond to $\lambda = 0$, $\lambda = 5$ and $\lambda = 10$. For each input and λ values, the absolute angle prediction error $|\Delta y|$ and the value of the Distraction Loss D are reported.

3.6 Conclusion

In this work, we have shown that privileged pixel-relevance information can be exploited during learning to benefit performances across a variety of offline metrics. The additional knowledge available during training doesn't need to be perfectly accurate to provide useful insights. In our case, lane marking ground truth was approximated using a pre-trained network without fine-tuning. The proposed method can be easily and efficiently implemented, without adding any learnable parameter to the architecture. Our approach was tested with real-world data and improves on the offline metrics that have the highest correlations with real-world driving performance. We leave the investigation of actual steering capabilities and the extension to larger datasets to future work.

Part II

Unsupervised Learning for Free Space Estimation

Chapter 4

Background on Learning-Based Free Space Estimation

This chapter serves as an introduction to Part II. In Section 4.1, we introduce the task of free space estimation in the context of perception for autonomous vehicles. Existing learning-based approaches using different degrees of ground truth supervision are reviewed in Section 4.2. Section 4.3 details Superpixel Clustering, an existing pseudo-labeling technique that will be used in Chapters 5 and 6. The evaluation procedure that will be followed in Chapters 5 to 8 is detailed in Section 4.4, and results for existing work and baselines are reported in Section 4.5. Finally, Section 4.6 concludes this chapter with an overview of Chapters 5 to 8.

4.1 Introduction

Autonomous navigation is one of the key problems in modern robotics. Before being able to safely plan and execute its motion, an autonomous vehicle should perceive its environment and identify drivable free space in an accurate manner. In this context, free space can be defined as road surfaces that are not occupied by other objects such as vehicles, traffic signs, road dividers or pedestrians [Jan+20]. Since collision avoidance requires a fine-grained understanding of the scene, our work aims to classify every pixel as belonging to either free space or occupied space. This task is commonly referred to as Free Space Estimation (FSE).

In this thesis, we focus our attention on systems using only a single road-facing camera. Although free space segmentation can be approached using classical supervised semantic segmentation techniques, they require large quantities of annotated images. While bounding-boxes for object detection can be relatively cheap to obtain, studies have shown that pixel-level annotations are significantly more time consuming [Lin+14]. In addition to the 1.5 hour labor cost associated with labeling every pixel in a single frame [Cor+15], a wide variety of environmental and weather conditions need to be captured. This creates a need for very large datasets, and renders fully-supervised semantic segmentation solutions impractical. This motivates more recent approaches that focus on unsupervised, pseudo-supervised or semi-supervised learning techniques.

4.2 Related Work on Free Space Estimation

The work presented in this document builds mainly on recent advances in network architectures for segmentation, and on pseudo-supervised techniques for generic semantic segmentation or specific to free space estimation. In this section, we present an overview of the landscape of free space estimation techniques, examining techniques that use different sensors or varying degrees of supervision. Each subsequent chapter also includes details on prior work specific to the ideas presented in that particular chapter.

Over the last decades, free space estimation has been approached with methods that leverage a wide variety of sensors, *e.g.* GNSS [Lad+16], LiDAR [Xia+15] or cameras [OBB16]. In this thesis, we place a particular focus on recent camera-based learning methods that use Convolutional Neural Networks.

4.2.1 Supervised Learning for Free Space Estimation

Varied Representations Supervised monocular free space estimation has been approached in many different ways that differ in the representation they use. A popular representation is the stixel world, which approximates the ground plane and represents obstacles as vertical sticks [BFP09; Cor+17], but ignores free space lying behind obstacles. Another possibility is to represent free space as a single horizontal curve lying on the ground plane [Yao+15]. These approaches are however not directly comparable to our work, since they do not annotate attempt to label every pixel.

Supervised Segmentation When framed as a pixel segmentation task, supervised free space estimation has directly benefited from progress in semantic segmentation. Pixel-level prediction carries a crucial challenge for network design: an optimal prediction can only be achieved by combining fine-grained local information with global contextual cues. Fully Convolutional Networks rely on skip connections to carry these cues in their encoder-decoder architecture [LSD15], while SegNets ease the upsampling task by reusing encoder max-pooling indices in the decoder [BKC17]. Building on similar ideas, U-Nets combine entire encoder feature maps with decoder features at each step of the expansion path of the network [RFB15]. The basic U-Net architecture, which we described in more details in Section 2.5, has attracted a lot of attention in recent years. Researchers have proposed refinements such as the use of dense connections [Jég+17] and dilated convolutions [Zha+17], the integration of attention mechanisms [Okt+18], or extensions to volumetric images [MNA16]. Interestingly, recent work has shown that many proposed U-Net architecture improvements are outperformed by a well-tuned vanilla U-Net [Ise+18], motivating our adoption of a vanilla U-Net architecture in our own work.

4.2.2 Unsupervised Free Space Segmentation

The efficiency of deep networks for free space segmentation has already been demonstrated in the fully-supervised context [OBB16]. One major drawback of these techniques is their reliance on extensive human-annotated datasets. The cost of labeling is particularly important in segmentation tasks, where the total time required to annotate every pixel in a single frame can reach 1.5 hours in some cases [Cor+15]. The reuse of models pre-trained on very large datasets such as ImageNet [Den+09] partially alleviates this problem, but several thousands of training images are still routinely needed to reach adequate performance. In recent years, researchers have devised strategies to reduce or eliminate the need for human annotations during training.

Synthetic Data In the complete absence of pixel-wise ground truth labels, researchers have proposed to use domain adaptation to transfer knowledge gained from synthetic datasets to the real-world [Hof+16; Che+19b]. Domain adaptation consists in reducing the drop in performance observed when the training and test distributions differ significantly, which is the case when training on synthetic data and running inference on real-world images. For a comprehensive overview of this topic, we refer the reader to the survey in [Csu17].

Weakly-Supervised Segmentation When obtaining pixel-level ground truth is not an option, existing techniques rely on coarser labels that can be annotated much faster. Examples include supervision derived from bounding boxes [DHS15; Ker+20; Kho+17; Xie+20], image-level labels [PC15; Dur+17; TSK17], class activation maps [Cha+20], single points [Bea+16], or scribbles [Lin+16].

Pseudo-Supervised Segmentation Our work presented in Part II explores another avenue: we learn dense free space from single images using approximate masks that can be generated without requiring any supervision. One way of generating such *pseudo-labels* is to obtain depth information from stereo pairs and to extract a ground plane estimate, often using the v-disparity algorithm [LAT02; HAS16; MUT18]. Another possibility is to exploit strong road texture and location priors, by dividing the input into superpixels and clustering them based on saliency maps [TSK17] or semantic features [OBB16]. We stress that relying on pseudo-labels differs from approaches based on coarse labels as in pseudo-supervised learning, since the generated masks contain both false positives and false negatives. Indeed, bounding-boxes contain no false negative, and scribble or point supervisions do not include any false positive. Additional information on how the training strategy can be adapted to take pseudo-label noise into account is available in Section 5.2, and Chapter 6 covers data augmentation and self-training approaches.

4.2.3 Semi-Supervised Free Space Segmentation

Cases where fine-grained annotations are available for only a subset of the data fall in the realm of semi-supervised learning. Two main categories of approaches have emerged for semi-supervised learning in segmentation problems: consistency regularization and pseudo-labeling.

Consistency regularization builds on the concept that learned representations should be similar in some way for different perturbations of the same input. Since perturbing a given input does not require supervision, this idea can be used to learn robust representations on unlabeled data, which can then be fine-tuned using a small sample of labeled data [Zbo+21; Che+20b; Xie+19; Miy+17; Yun+19].

Pseudo-labeling takes the opposite route: a model is trained on the labeled data, and is then used to generate approximate pseudo-labels for the unlabeled data. A model can then be trained on the whole dataset, using both ground truth labels and pseudo-labels [Tsu+18; MUT18; Che+20a; Hoy+21; TV17; Rob+22a].

A more thorough introduction to these techniques is presented in Section 8.2.

4.3 Pseudo-labeling with Superpixel Clustering

This section gives an overview of the Superpixel Clustering approach introduced by Tsutsui et al. in order to generate free space pseudo-labels. We rely on this technique to generate the pseudo-labels used in Chapters 5 and 6.

The method can be summarized in three steps, illustrated on Figure 4.1:

1. *Superpixel Segmentation*: Pixels in the frame are grouped into superpixels based on their location and RGB values.
2. *Superpixel Features Extraction*: A single feature vector is computed for each superpixel.
3. *Superpixel Clustering*: Superpixels are clustered based on their feature vectors and a cluster is selected to represent the road.

4.3.1 Superpixel Segmentation

Similar color-space regions are aggregated into superpixels using the graph-based segmentation algorithm from Felzenszwalb and Huttenlocher [FH04]. This produces a rough segmentation and the method will proceed by clustering superpixels into four clusters and classifying one cluster as the road.

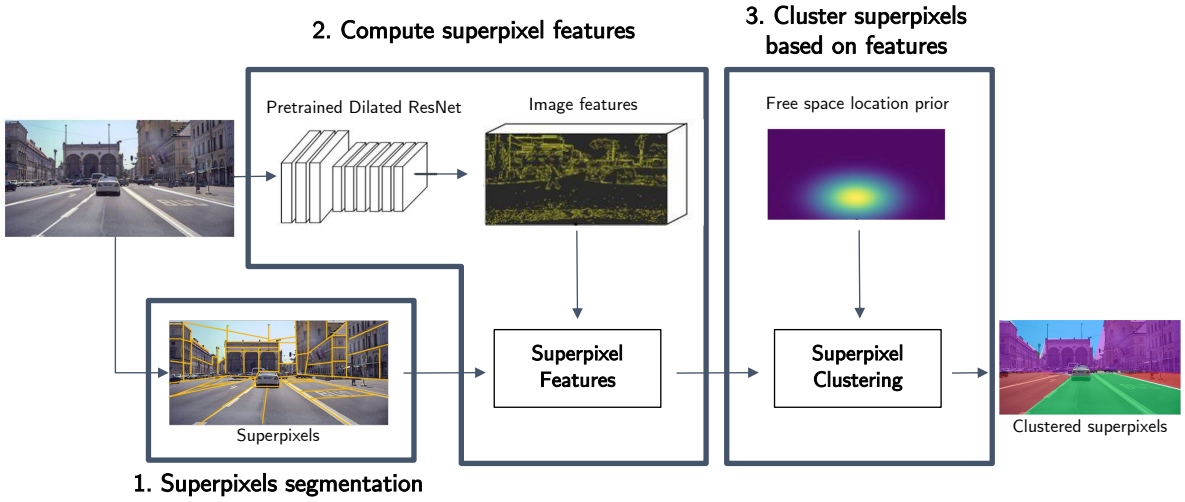


FIGURE 4.1: Pseudo-labels generation through Superpixel Clustering [Tsu+18]. The superpixels and clusters are simplified for illustration clarity.

4.3.2 Superpixel Features Extraction

In order to cluster superpixels, a feature vector is computed for each of them. Feature maps are extracted from a dilated ResNet that was pre-trained on ImageNet. Since the method aims at classifying superpixels, it needs to aggregate feature vectors over entire superpixels. Because the spatial dimensions of the feature maps computed in the previous step do not match the input shape, the authors propose a superpixel alignment scheme. Inspired by the ROIAlign procedure used in Mask-RCNN [He+17], they compute feature vectors for each superpixel for a random sample of pixels in each superpixel by bilinearly interpolating feature maps. The superpixel feature vector is then obtained by averaging the feature vector of each sample.

4.3.3 Superpixel Clustering

Superpixels are grouped into four clusters based on their feature vectors, as well as the spatial position of their centroids. Although any standard clustering algorithm can be used for this step, an important problem is to determine which cluster corresponds to free space. The authors take advantage of the fact that, in the Cityscapes dataset, the road surface is commonly located right above the hood of the ego-vehicle. They obtain four superpixel clusters using a batched weighted K-Means scheme that takes a gaussian location prior for the road into account. The road cluster is identified as as the closest match to the road prior.

4.4 Evaluation

In this section, we detail the dataset and evaluation metrics used to benchmark the free space estimation methods introduced in Chapter 5 to 8.

4.4.1 The Cityscapes Dataset

The Cityscapes dataset provides pixel-wise annotations for 30 visual classes in 5000 frames [Cor+15]. Since the test set has no public annotation, and to enable comparison with existing work, we treat the 500 frames of its validation set as our test set and randomly split the Cityscapes training set into 2380 training and 595 validation frames. In the context of autonomous robot navigation, we consider free space to correspond to the *road* object class. Cityscapes also contains 1.6% of frames where no pixel is labeled as *road*. For these frames only, we use the *ground* class to denote free space. Visual inspection confirmed that *ground* corresponds to free space in these frames. Finally, the semantic labels include 6 *void* classes such as *unlabeled*, *out of the region of interest* or *ego-vehicle*. Following Cityscapes semantic segmentation benchmarks, pixels that correspond to such classes are ignored at evaluation time using a mask $m \in \{0,1\}^{H \times W}$.

4.4.2 Evaluation Metrics

Our evaluation relies on three metrics: Intersection-over-Union (IoU), Precision and Recall of the free space class. IoU reflects the overall quality of the prediction, but does not immediately capture the fraction of pixels that are labeled as part of the road when they are actually occupied. Since these *false free space positives* are extremely harmful to a robot navigation scenario, we also emphasize the importance of measuring the Precision of our predictions, *i.e.* the fraction of our free space prediction that is indeed free space. Although it is also interesting to monitor Recall, we note that missing free space has less impact than false positives in an autonomous driving scenario.

The computation of IoU, Precision and Recall can be visualized for a Cityscapes sample frame on Figure 4.2. Given a single free space prediction $\hat{y}$, ground truth y, and evaluation mask m, the metrics for a single frame of shape $H \times W$ are computed with Equations 4.1 to 4.3, where $\hat{y},\ y,\ m \in \{0,1\}^{H \times W}$. Note that since the output of our models are real-valued in the $[0,1]$ interval, a thresholding operation is needed. Unless otherwise stated, results are reported for the standard threshold $\tau = 0.5$.

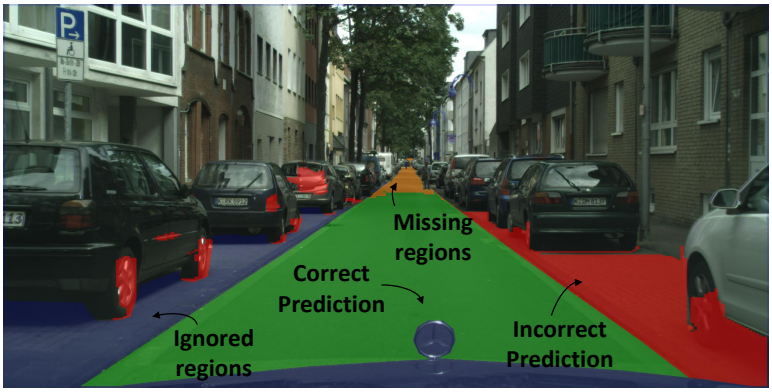


FIGURE 4.2: Visualization of IoU, Precision and Recall using a sample from Cityscapes. Pixels labeled as one of the void classes (*e.g. ego-vehicle*, *out of ROI* or *unlabeled*) are ignored at evaluation time.

$$IoU = \frac{\sum_i \hat{y}_i y_i m_i}{\sum_i (\hat{y}_i + y_i - \hat{y}_i y_i) m_i} \tag{4.1}$$

$$Precision = \frac{\sum_i \hat{y}_i y_i m_i}{\sum_i \hat{y}_i m_i} \tag{4.2}$$

$$Recall = \frac{\sum_i \hat{y}_i y_i m_i}{\sum_i y_i m_i} \tag{4.3}$$

4.5 Existing Results & Baselines

This section presents existing results that will put the results from the methods presented in subsequent chapters in perspective. The results from Table 4.1 are split into four categories, which are detailed in separate subsections: (1) supervised U-Net results, (2) the Bottom-Half baseline, (3) unsupervised baselines based on SC pseudo-labels, and (4) other unsupervised approaches. Figure 4.3 illustrates the output and IoU value for the most relevant baselines on a specific Cityscapes test sample.

4.5.1 Supervised Results

Since Cityscapes provides human annotations for all of the data, it is natural to compare our unsupervised approach with its supervised counterpart. To this end, we train a *Supervised U-Net* using the ground truth labels and observe that it is able to reach high IoU (94.12%), Precision (97.26%), and Recall (0.9727). Since this is the only method that uses the ground truth labels for training or validation, we expect it to provide an upper-bound for unsupervised or semi-supervised results.

	Training Labels	Test IoU	Test Precision	Test Recall
Supervised U-Net	ground truth	0.9412	0.9726	0.9727
Bottom Half	no training	0.7550	0.7798	0.9616
Raw SC Pseudo-labels [Tsu+18]	no training	0.7900	0.8778	0.8924
SC SegNet	SC Pseudo-labels	0.8130	0.8936	0.9015
SC U-Net	SC Pseudo-labels	0.8152	0.8854	0.9138
PACA [Hof+16]	synthetic data	0.7040	not reported	not reported
Distant Supervision [TSK17]	DS Pseudo-labels	0.8000	not reported	not reported
Stereo v-disparity [MUT18]	v-disparity reprojection	0.8001	0.9283	0.8529

TABLE 4.1: Quantitative results on the Cityscapes validation set, which we treat as our test set to enable comparison with existing work.

(A) Supervised U-Net (97.38% IoU)

(B) Bottom-Half (70.70% IoU)

(C) Superpixel Clustering Pseudo-label (87.73% IoU)

(D) SegNet trained on SC Pseudo-labels (55.35% IoU)

(E) U-Net trained on SC Pseudo-labels (82.97% IoU)

FIGURE 4.3: Baseline outputs and IoU for a single Cityscapes test sample. Superpixel Clustering pseudo-labels are obtained using our own implementation following [Tsu+18].

4.5.2 Bottom-Half Baseline

In order to compare our approach to other algorithms, we use two simple methods that do not need training and should act as lower bounds. The *Bottom-Half* model is a trivial baseline that classifies the entire lower half of the image as free space. This process is illustrated on Figure 4.3b. Bottom-Half is able to reach a decent IoU of 75.50% and a high Recall, which is not surprising since free space indeed covers a large portion of the lower half of most frames. The Precision of this model is however only of 77.98%, which is poor compared to other less trivial baselines.

4.5.3 Unsupervised Baselines based on Superpixel Clustering

Raw SC Pseudo-labels We generate approximate labels without supervision using the Superpixel Clustering technique described in Section 4.3 [Tsu+18]. Evaluating these raw pseudo-labels, we obtain an IoU of 79.00%, a Precision of 87.78% and a Recall of 89.24%.

SegNet and U-Net trained on SC Pseudo-labels A common way to improve over results obtained by raw pseudo-labels is to train a predictive model to generalize beyond the noise in these labels. This was already attempted using the SegNet architecture in [Tsu+18]. Since the authors of [Tsu+18] do not provide pre-trained weights with their implementation[1], we have reimplemented their method using Pytorch [Pas+19] in order to compare it to our own methods. Our SegNet baseline is able to improve results over raw pseudo-labels in IoU (+2.30%), Precision (+1.58%) and Recall (+0.91%). Our re-implementation reaches slightly better results than the original Chainer implementation distributed by the authors [AFS17], presumably due to minor differences in Pytorch layer implementations. We also train a U-Net architecture based on a ResNet-18 backbone to perform the same task. Since SegNet yields slightly worse IoU results than U-Net and is slower at training and inference time, we focus our experiments on U-Nets in subsequent chapters. We stress the fact that both the SegNet and U-Net models do not use any human-annotated ground truth at any point during training or validation. Although training from pseudo-labels results in an overall improvement compared to the raw pseudo-labels, Figure 4.3d shows that it is not always the case for specific samples.

4.5.4 Other Unsupervised Approaches

Competing unsupervised approaches often tackle the more general problem of semantic segmentation, for which other datasets are preferred to Cityscapes [DHS15; Xie+20; PC15; Dur+17; Cha+20]. Furthermore, papers that use the Cityscapes benchmark seldom report road-class IoU. Recent unsupervised free space estimation works also use varied

[1] https://github.com/pfnet-research/superpixel-align

datasets [HAS16; Yao+15]. Two exceptions are *Distantly Supervised Road Segmentation* [TSK17] and *Pixel-level Adversarial and Constraint-based Adaptation* [Hof+16], which report Road IoU but not Precision or Recall.

Pixel-level Adversarial and Constraint-based Adaptation (PACA) PACA trains an FCN for semantic segmentation by leveraging both annotated frames from a synthetic 3D-rendered dataset and unannotated ones from the Cityscapes training set. The method therefore uses ground truth frames from a source domain where they are readily available, in order to transfer knowledge to a target domain where only unannotated frames can be used. The main challenge of this simulation-to-reality transfer is the global domain shift due to synthetic frames looking different than real ones overall. To address this issue, the authors propose a domain adaptation loss that forces the representations learned from the target domain to be close to the ones from the source domain. Although the authors address the generic problem of segmenting frames into 19 different classes, they also share results for the road class specifically, where they obtain an IoU of 70.40%.

Distantly Supervised Road Segmentation Distantly Supervised Road Segmentation [TSK17] is another example of pseudo-supervised learning, where the authors propose to derive pseudo-labels from saliency maps obtained from an ImageNet-trained classifier [Den+09]. Images containing roads are obtained by searching ImageNet for road-related labels to train a CNN to distinguish road images from negative samples. Global average pooling [Zho+16] is then used to extract saliency maps from the last convolutional layer of this trained classifier. The authors obtain distantly-supervised (DS) pseudo-labels by computing the intersection between a saliency map and a graph-based superpixel segmentation, before thresholding to classify highly salient superpixels as part of the road. A Fully-Convolutional Network (FCN) [LSD15] is then trained under pseudo-supervision and reaches an IoU of 80.00%.

Stereo v-disparity Pseudo-Supervision Mayr, Unger, and Tombari have proposed the use of stereo frames at training time, in order to generate depth-based pseudo-labels [MUT18]. They estimate sparse depth maps using stereo reconstruction, and identify planar surfaces in disparity-space using the v-disparity algorithm. Pseudo-labels are obtained by labeling any pixel that is part of the dominant planar surface as part of the road. The authors use an evaluation procedure similar to the one described in Section 4.4, but their definition of the road also includes the *parking* and *ground* classes. The authors report an IoU of 80.01%.

4.6 Overview of the Next Chapters

The next four chapters will exploit a number of concepts for improving over the baselines described in Section 4.5. Figure 4.4 illustrates how these different research avenues are explored in subsequent chapters.

Superpixel Clustering Pseudo-labels This pseudo-labeling method has already been detailed in Section 4.3. We have re-implemented this technique and use the generated pseudo-labels in both Chapter 5 and 6.

Accounting for label noise during training Some training techniques are explicitly designed to take label noise into account during the training process, resulting in increased robustness to mistakes in the pseudo-labels and better generalization on test data. Co-Teaching is one such technique that has been proposed for classification problems. We adapt it for segmentation and propose a Stochastic Co-Teaching variant in Chapter 5.

Self-training Since training a model on noisy pseudo-labels results in predictions that are more accurate than the original pseudo-labels, one can also recursively train a new model on the output of the first one. This idea is explored in Chapter 6.

Data Augmentation & Self-training Beyond the training algorithm itself, we also analyze different data augmentation strategies in Chapter 6. The Cutmix augmentation strategy is also used in the methods proposed in Chapters 7 and 8.

Pseudo-labels based on v-disparity Chapter 7 proposes a different way of generating free space pseudo-labels using disparity information obtained from a depth estimation network, the v-disparity algorithm and a novel filtering technique. We show that replacing the SC pseudo-labels used thus far with the v-disparity pseudo-labels results in performance improvements.

Using a restricted subset of ground truth Chapters 5 to 7 strictly focus on learning to estimate free space without leveraging any annotated frame. In Chapter 8, we relax this constraint and take a more pragmatic approach by accepting the use of a restricted subset of ground truth. The proposed method builds on the v-disparity pseudo-labels proposed in Chapter 7 and explores the use of 1% or 10% of labeled data at training time.

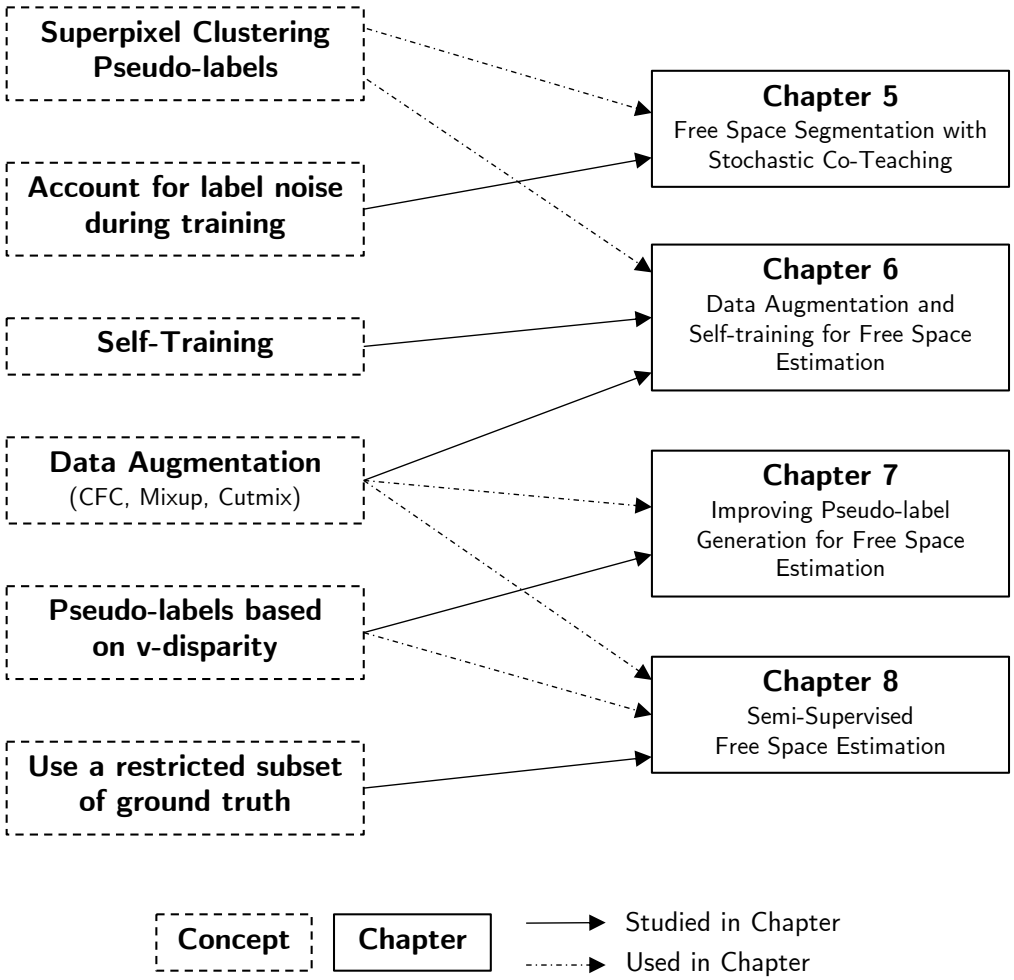


FIGURE 4.4: Overview of the concepts introduced or used in the next chapters.

Chapter 5

Free Space Estimation through Stochastic Co-Teaching

5.1 Introduction

Free space estimation is an important problem for autonomous robot navigation. As explained in Chapter 4, traditional camera-based approaches train a segmentation model using an annotated dataset. However, the training data needs to capture the wide variety of environments and weather conditions encountered at runtime, making the annotation cost prohibitively high.

In this chapter, we propose a novel approach for obtaining free space estimates from images taken with a single road-facing camera. We rely on the technique presented in Section 4.3 to generate approximate free space labels without any supervision [Tsu+18]. These pseudo-labels are then used as ground truth to train a segmentation model for free space estimation.

This chapter is based on the following published article:

François Robinet, Claudia Parera, Christian Hundt, and Raphaël Frank.
"Weakly-Supervised Free Space Estimation Through Stochastic Co-Teaching". In:
Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops. Jan. 2022, pp. 618–627

Our work differs from prior attempts by explicitly taking label noise into account through the use of Co-Teaching. Since Co-Teaching has traditionally been investigated in classification tasks, we adapt it for segmentation and examine how its parameters affect performances in our experiments. In addition, we propose *Stochastic Co-Teaching*, which is a novel method to select clean samples that leads to enhanced results.

We achieve an IoU of 82.6%, a Precision of 90.9%, and a Recall of 90.3%. Our best model reaches 87% of the IoU, 93% of the Precision, and 93% of the Recall of the equivalent fully-supervised baseline while using no human annotations. To the best of our

knowledge, this work is the first to use Co-Teaching to train a free space segmentation model under explicit label noise, and these results constituted the state-of-the-art at the time of their submission. We have since improved them using the techniques described in Chapters 6 and 7.

The contributions of the work presented in this chapter can be summarized as follows: 1) we adapt Co-Teaching for segmentation tasks and illustrate its effectiveness on the particular case of free space estimation, 2) we study the impact of the Co-Teaching schedule on performances, 3) we propose a refinement called *Stochastic Co-Teaching*, 4) we compare Stochastic Co-Teaching to standard training and traditional Co-Teaching and observe improvements in both IoU and Precision, and 5) we analyze the limitations of the proposed approach. We also make our code and models available online.

The remainder of this chapter is organized as follows: In Section 5.2, we review the recent literature for both free space estimation and pseudo-supervised segmentation. In Section 5.3, we introduce our pseudo-supervised Co-Teaching approach to free space estimation and describe the baseline methods used for benchmarking. In Section 5.4, we detail the experimental setup of this study. In Section 5.5, we carry out experiments, detail the qualitative and quantitative results achieved, analyze the limitations of our approach, and share further research directions. Finally, we conclude with a summary of our contributions.

5.2 Related Work on Segmentation under Label Noise

Section 4.2 has already covered learning techniques for supervised and unsupervised semantic segmentation, as well as segmentation techniques specific to free space. Rather than repeating these references, this section specifically focuses on techniques for robustly training under label noise.

Recent research has shown that it is possible to train over-parametrized models to generalize past some of the label noise using Stochastic Gradient Descent (SGD) schemes combined with early stopping [LSO20]. However, approaches that explicitly deal with noisy labels can further improve performances, and have become an important research focus over the past few years. Solutions to this problem include label cleaning [Chi+19], noise-aware network architectures [Suk+15], or noise reduction through robust loss functions [MES08; Lu+17; Rob+20]. Another line of research proposes to adapt the training procedure itself. Curriculum learning [Ben+09] is based on training a model on samples of increasing difficulty, which can correspond to different noise levels [Jia+18; Guo+18]. Knowledge distillation [HVD15] is another procedure that can cope with noise by training the teacher model on a relatively clean subset of the data, and using it to guide the training of the student model on the whole dataset [Li+17]. Decoupling [MSS17] and Co-Teaching [Han+18] are two other approaches where two models are trained simultaneously. Decoupling trains both models only on data where their outputs disagree, while Co-Teaching trains each model on the fraction of the data that the other considers to be

clean. For a more comprehensive overview of techniques that cope with noisy labels in image analysis, we refer the reader to the survey in [Kar+20].

The work presented in this chapter builds on supervised segmentation research through its use of the U-Net architecture described in Section 2.5. We address label noise using a Co-Teaching training scheme that we adapt for segmentation tasks. We choose Co-Teaching because it has been shown to perform well under moderate amounts of noise [Che+19a]. We present our method in detail in the next section.

5.3 Methodology

In this section we describe the main steps of our pseudo-supervised approach for the free space estimation task using Co-Teaching. We present Co-Teaching and its adaptation for a segmentation task, and we introduce our Stochastic variant. Since we focus on improving the training aspect, we use the pseudo-labels described in Section 4.3 as targets during training. We benchmark the performances of (Stochastic) Co-Teaching against a fully-supervised model, as well as against unsupervised and pseudo-supervised baselines in Section 5.5.

5.3.1 Co-Teaching for Segmentation

The intuition for Co-Teaching is based on memorization properties of deep neural networks trained with a variant of SGD [Han+18]. Although these networks are capable of overfitting random noise in their training set [Zha+16], they also learn patterns from clean data first [LSO20]. To exploit this property, Co-Teaching proposes to train two separate student networks f and g, and to have each one select clean instances for the other to train on. Since models tend to learn from clean data first, standard Co-Teaching selects clean labels as the ones with the lowest loss. The main additional meta-parameter of Co-Teaching is a schedule $R(t)$ that defines the fraction of the data that is considered clean and should be used for training at any given iteration t. In early iterations, models have not learned enough to identify noise in the training data, and we should generally set $R(t) \approx 1$. As training goes on, $R(t)$ should tend towards the expected noise rate τ, such that all the noise and only the noise gets discarded. In a pseudo-supervised setting, τ is unknown and one must either estimate it or try different schedules. Although it has traditionally been used in classification tasks, Co-Teaching has also recently seen some success when training an object detector from noisy bounding boxes [CN20]. To the best of our knowledge, this work is the first to adapt it to a segmentation task. Our adaptation is straightforward: we consider each pseudo-labeled pixel as an independent label, and therefore train each network on a sample of all pixels in a batch. The entire procedure is detailed in Algorithm 1.

5.3.2 Stochastic Co-Teaching

We further propose a refinement named *Stochastic Co-Teaching*. Rather than systematically selecting the subset of labels that incur the least loss, Stochastic Co-Teaching samples them with weights that are inversely proportional to their loss. With this change, when labels in a batch incur similar losses, they are all similarly likely to be selected for training, regardless of the schedule $R(t)$. We are trusting that low-loss labels are more likely to be clean, but we accept that some higher-loss samples can also be selected with lower probability. Rather than being noisy, some of these high-loss labels may correspond to harder examples that should not be systematically discarded. Figure 5.1 illustrates the whole process: (a) An identical batch of B images with resolution $H \times W$ is fed to independent networks f and g, (b) pixel-wise losses $L^{(f)}$ and $L^{(g)}$ are computed using the noisy labels for each image, (c) clean indices $I^{(f)}$ and $I^{(g)}$ are independently selected for each student network by sampling a fraction $R(t)$ of indices without replacement and with probability inversely proportional to their loss, (d) networks exchange their clean indices and use them to sub-sample their own losses and obtain $\bar{L}^{(f)}$ and $\bar{L}^{(g)}$, (e) each network only learns from pixels that the other student deems clean. Note that nothing prevents the use of networks f and g with completely different topologies. In this work, students share the same architecture, but their weights are randomly initialized independently from each other.

Algorithm 1: (Stochastic) Co-Teaching for Segmentation

Inputs: Models f and g, data generator $\mathcal{G}$, loss $\mathcal{L}$, max iterations T and schedule $R(t)$

forall $t \in \{1, \dots, T\}$ **do**
- Obtain next batch $(x, y_{weak}) \in \left(\mathbb{R}^{B\times H\times W\times 3} \times \mathbb{R}^{B\times H\times W}\right)$ from training data $\mathcal{G}$
- Compute per pixel losses $L^{(f)} = \mathcal{L}\left(f(x), y_{weak}\right)$ and $L^{(g)} = \mathcal{L}\left(g(x), y_{weak}\right)$
- Compute $n = R(t) \times B \times H \times W$, the number pixel samples to keep
- **if** *stochastic co-teaching* **then**
  - Let $\mathcal{S}(w, k)$ randomly sample k unique indices of w, using values of w as weights
  - Sample n clean indices $I^{(f)} = \mathcal{S}\left(1/L^{(f)}, n\right)$ and $I^{(g)} = \mathcal{S}\left(1/L^{(g)}, n\right)$
- **else**
  - Let $TopK\left(z, k\right)$ select the indices of the k largest elements of z
  - Compute n clean indices $I^{(f)} = TopK\left(1/L^{(f)}, n\right)$ and $I^{(g)} = TopK\left(1/L^{(g)}, n\right)$
- Compute clean losses $\bar{L}^{(f)} = \left\{L_i^{(f)} : i \in I^{(g)}\right\}$ and $\bar{L}^{(g)} = \left\{L_i^{(g)} : i \in I^{(f)}\right\}$
- Update f using $\nabla \bar{L}^{(f)}$ and g using $\nabla \bar{L}^{(g)}$

5.4 Experimental Setup

Dataset & Evaluation We report IoU, Precision and Recall results on the Cityscapes dataset [Cor+15]. Complete descriptions of the Cityscapes dataset and of the evaluation metrics were respectively given in Sections 4.4.1 and 4.4.2.

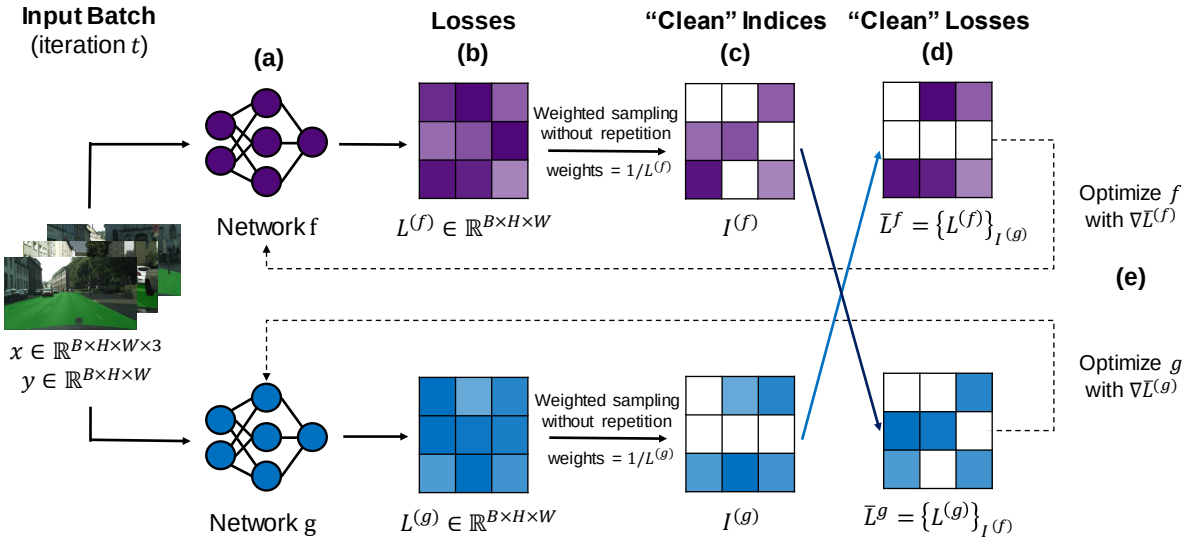


FIGURE 5.1: Stochastic Co-Teaching. Student networks compute their own pixel-wise loss, and randomly select a subset of clean pixels to train on, based on loss values of the other student.

Network architectures Following recent research that shows that a well-tuned vanilla U-Net can outperform many variants on segmentation tasks [Ise+18], we opt for a U-Net structure based on a ResNet18 backbone (14.3M parameters) [RFB15; He+16; Yak19]. To compare with prior art, we also implement and train the SegNet model described by Tsutsui et al. [Tsu+18]. For computational reasons, we use a 512×1024 input resolution in all experiments. Outputs are however re-scaled using nearest neighbor interpolation in order to compute metrics in the original 1024×2048 resolution.

Training procedure Since Tsutsui et al. do not share trained weights for their SegNet architecture, all models are trained from randomly initialized weights to allow for fair comparison. We use the PyTorch framework [Pas+19] and train models with the Adam optimizer [KB15], a batch size of 6 and a learning rate of 0.001. The models are trained on a single NVIDIA K80 until the training loss plateaus, which occurs after 25 epochs for U-Nets and 50 epochs for SegNets. We keep all intermediate models and perform model selection post-training.

Model selection In the context of pseudo-supervised learning, we must be careful when performing model selection. This is especially important since Cityscapes provides ground truth annotations for all training and validation frames used in this study. We stress that these frames are never used for training, picking hyper-parameters, or to perform early stopping. We therefore evaluate ground truth IoU, Precision and Recall only once on the test set, after all these steps have been performed. Models trained with standard training loops are selected to minimize the validation loss. This approach has to be slightly adapted when using Co-Teaching, since the scale of the loss varies with the value of $R(t)$, the fraction of data to be considered clean over time. As explained in Section 5.3,

	Training/Validation Labels	Test IoU	Test Precision	Test Recall
Supervised U-Net (standard training)	ground truth	0.9412	0.9726	0.9727
Supervised U-Net (Co-Teaching ensemble)	ground truth	0.9311	0.9621	0.9646
Supervised U-Net (Stochastic Co-Teaching ensemble)	ground truth	0.9360	0.9664	0.9655
Bottom Half	no training	0.7550	0.7798	0.9616
Raw Pseudo-labels [Tsu+18]	no training	0.7900	0.8778	0.8924
Unsupervised Domain Adaptation [Hof+16]	synthetic data	0.7040	not reported	not reported
Distant Supervision [TSK17]	image labels	0.8000	not reported	not reported
Standard SegNet	pseudo-labels	0.8130	0.8936	0.9015
Standard U-Net	pseudo-labels	0.8152	0.8854	**0.9138**
Co-Teaching U-Net (best student)	pseudo-labels	0.8214	0.8995	0.9074
Stochastic Co-Teaching U-Net (best student)	pseudo-labels	0.8237	0.9076	0.9017
Co-Teaching U-Net (ensemble)	pseudo-labels	0.8219	0.9028	0.9047
Stochastic Co-Teaching U-Net (ensemble)	pseudo-labels	**0.8261**	**0.9093**	0.9027

TABLE 5.1: Quantitative results on the Cityscapes validation set, which we treat as our test set.

$R(t)$ usually starts at 1, before decreasing to a minimum, and again increasing to plateau at $R(t) = \tau$. When $R(t)$ is smaller, a larger fraction of high-loss samples is discarded and the loss value is deflated. To account for this, we only select models in the *final plateau* of our schedules, where $R(t) = \tau$. For more information about the schedules $R(t)$ we consider, see Section 5.6.

5.5 Results

This section outlines the set of experiments carried out to benchmark our proposed method, using IoU, Precision and Recall. Five main types of approaches were tested: 1) Fully-supervised upper-bounds, 2) unsupervised baselines, 3) standard training, 4) (Stochastic) Co-Teaching, and 5) Ensembling (Stochastic) Co-Teaching students. The quantitative results are summarized in the five categories of Table 5.1. In this section, we present results for each category and qualitative results for our best model.

Fully-Supervised Upper-bound Since Cityscapes provides human annotations for all of the data, it is natural to compare our unsupervised approach with its supervised counterpart. To this end, we train a *Fully-Supervised U-Net* using the ground truth labels and observe that it is able to reach high IoU (0.9412) and Precision (0.9726). Since this is the only method that uses the ground truth labels for training or validation, we expect it to provide an upper-bound for unsupervised results. To account for the effect of potential noise in the ground truth, we also train the same network using (Stochastic) Co-Teaching. Since ground truth data is assumed to contain only a small amount of noise, we use a specific $R(t)$ schedule that trains on the whole training data for one epoch, before progressively discarding up to 4% of the training data and slowly incorporating 3% back in to finish training with $R(t) = 0.99$. Examples of similar schedules are illustrated in Section 5.6. We observe that both Co-Teaching and Stochastic Co-Teaching result in

degraded performance in the fully-supervised case. We suggest two hypotheses to explain this. The most likely scenario is that we are discarding valuable data and that ground truth label noise, although always present in pixel-wise annotations, is likely negligible for our purposes. A second option is that the ground truth label noise present in training data is also present in test data, making exact evaluation unreliable.

Unsupervised Baselines In order to compare our approach to other algorithms, we use two simple methods that do not need training and should act as lower bounds. The *Bottom Half* model is a trivial baseline that classifies the entire lower half of the image as free space. Bottom-Half is able to reach a decent IoU of 0.7550 and a high Recall, which is not surprising since free space indeed covers a large portion of the lower half of most frames. The Precision of this model is however only of 0.7798, which is poor compared to the 0.8778 achieved by our second unsupervised baseline, the raw *Pseudo-labels* from [Tsu+18]. This second baseline also yields a large IoU improvement, reaching 0.7900. Competing unsupervised approaches often tackle the more general problem of semantic segmentation, for which other datasets are preferred to Cityscapes [DHS15; Xie+20; PC15; Dur+17; Cha+20]. Furthermore, papers that use the Cityscapes benchmark seldom report road-class IoU. Recent unsupervised free space estimation works also use varied datasets [MUT18; HAS16; Yao+15]. Two exceptions are presented in [TSK17] and [Hof+16], which respectively obtain an IoU of 0.8 and 0.704, but do not report Precision and Recall.

Standard Training We train both our own U-Net and the SegNet model described in [Tsu+18], using the pseudo-labels as targets in a standard training loop. We stress the fact that these models do not use any human-annotated ground truth at any point during training or validation. As previously observed in [LSO20], standard training is robust to noise to some degree, and our models are able to generalize beyond the noise in their training targets. Compared to raw pseudo-labels, U-Net is able to improve in IoU (+2.52%), Precision (+1.58%), and Recall (+2.34%). Since SegNet yields slightly worse IoU results than U-Net (+2.3%) and is slower at training and inference time, we focus our Co-Teaching experiments on U-Nets.

(Stochastic) Co-Teaching Training We first report results from the best student models, which are selected as having the lowest validation loss among the two students of each Co-Teaching experiment. Note that all models trained with Co-Teaching in the 4th and 5th sections of Table 5.1 use a tuned $R(t)$, whose effect on the performance is explored in Section 5.6. Co-Teaching is able to improve over standard training in both IoU (+0.62%) and Precision (+0.76%), while our stochastic variant results in an additional improvement of 0.23% in IoU and 0.81% in Precision.

Ensembling Models trained with (Stochastic) Co-Teaching To assess their convergence, we run Co-Teaching trained students over the training data and we observe a high agreement over the free space predictions (99.2% of pixels are predicted the same),

and over which pixels should be considered clean for training (99.4% agreement). Due to its additional sampling step, we observe a slightly lower convergence when Stochastic Co-Teaching is used (97.6% agreement on predictions, 97.8% on clean indices). Since the students exhibit similar validation losses but are not completely equivalent, it is natural to ensemble them by averaging their confidence outputs before thresholding for a prediction. We obtain our best model using this strategy with the Stochastic Co-Teaching students, yielding an IoU of 82.61%, a 0.24% improvement over the best student, and a 0.42% improvement over the Co-Teaching equivalent. These results amount to 87% of the IoU, 93% of the Precision, and 93% of the Recall of the fully-supervised baseline while not using any human labels.

Qualitative Results Figure 5.2 shows test set predictions of our Stochastic Co-Teaching U-Net, and compares them against the Cityscapes ground truth and raw pseudo-labels. The first three columns of images illustrate the higher Precision of our learned model. It is able to classify regions such as cars or pedestrians as occupied, even though pseudo-labels mark them as free space. Improvements in Precision happen at the cost of Recall, and our predictions tend to be less homogeneous than pseudo-labels. This can be explained by the fact that pseudo-labels are obtained by clustering superpixels and therefore profit from the homogeneity of the superpixel segmentation. Finally, the last row illustrates a failure case: the pseudo-labels are almost flawless but the model fails to segment free space correctly.

5.6 Co-Teaching Schedule Impact

The most important hyper-parameter of Co-Teaching is the schedule $R(t)$, which controls the fraction of the training data that should be considered clean at any epoch t. Following recent research, our $R(t)$ starts at one, decrease to a minimum and then increase to plateau at a final value $R(T)$ [Yao+20]. We choose piecewise linear schedules, and vary the length of their *warmup phase* where $R(t) = 1$, their minimum and their final value. Figure 5.3 illustrates the schedules we consider, and the two parts of Table 5.2 presents the corresponding test set metrics for the Co-Teaching U-Net model presented in Section 5.5. In the first part of Table 5.2, we alter both the minimum and final values of $R(t)$. The 70%-85% schedule discards so much data that many clean labels are also ignored. This lowers the IoU to 0.3749, while the decreased noise allows the Precision to rise to 0.9352. As more data is kept in rows 2 to 4, IoU increases, while Precision slightly decreases. Since our goal is to balance IoU with Precision, we select 90%-95% for further investigation. In the second part of Table 5.2, we keep the schedule bounds fixed to 90%-95% and vary the length of the initial warm-up phase, where $R(t) = 1$ and all the data is kept. We observe little impact on IoU, but Precision gradually rises with shorter warm-up phases. This indicates that few iterations are enough for the student models to identify and discard noise, and we select the 90%-95% with a single warm-up epoch as our best $R(t)$ schedule, for which we reported our results in Section 5.5.

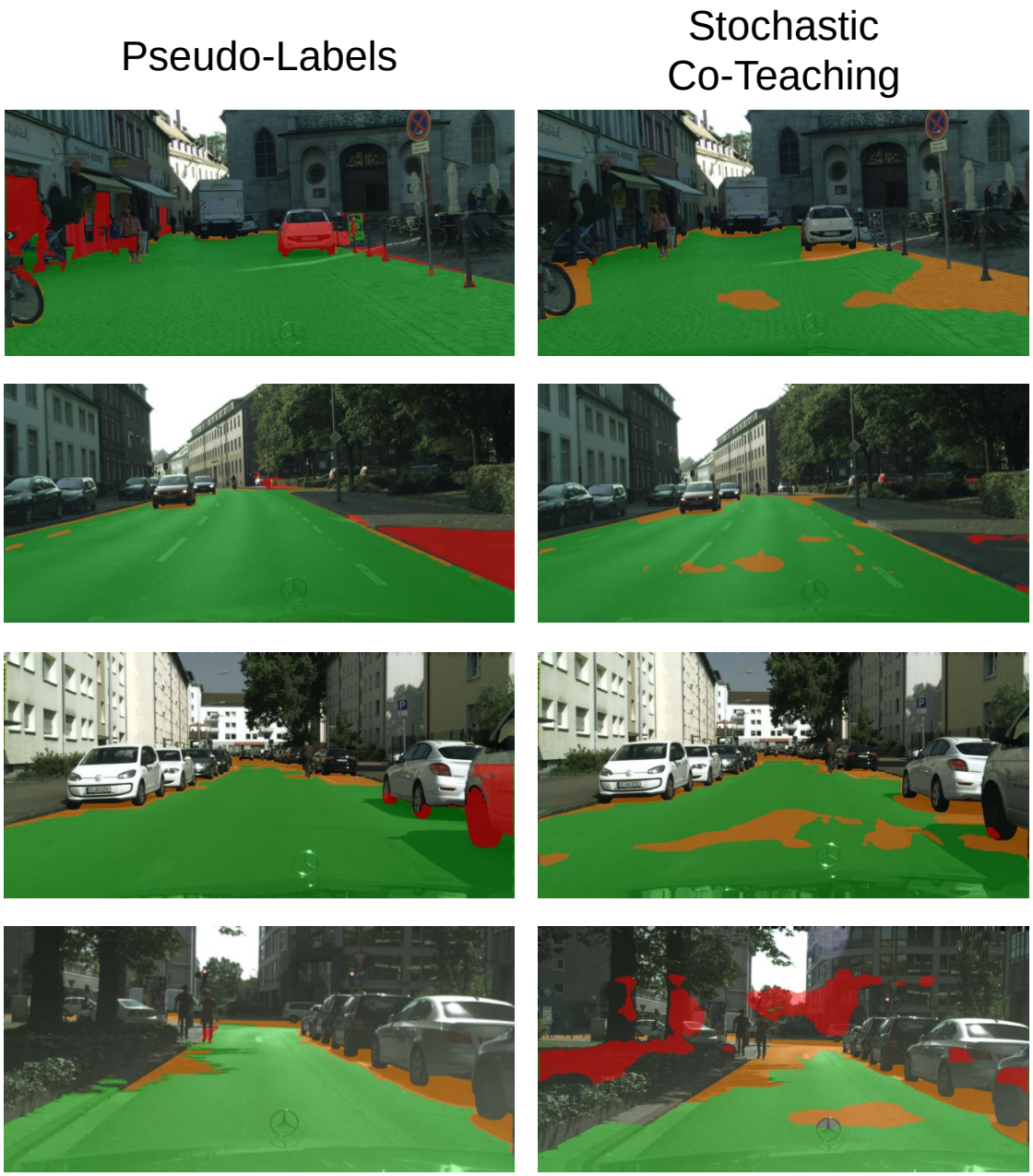


FIGURE 5.2: Qualitative results from the test set. Green, red and orange respectively indicate correct, incorrect and missing free space predictions. Note that we display the raw outputs of our model, without masking the ego-vehicle or *void* classes discussed in Section 4.4.1.

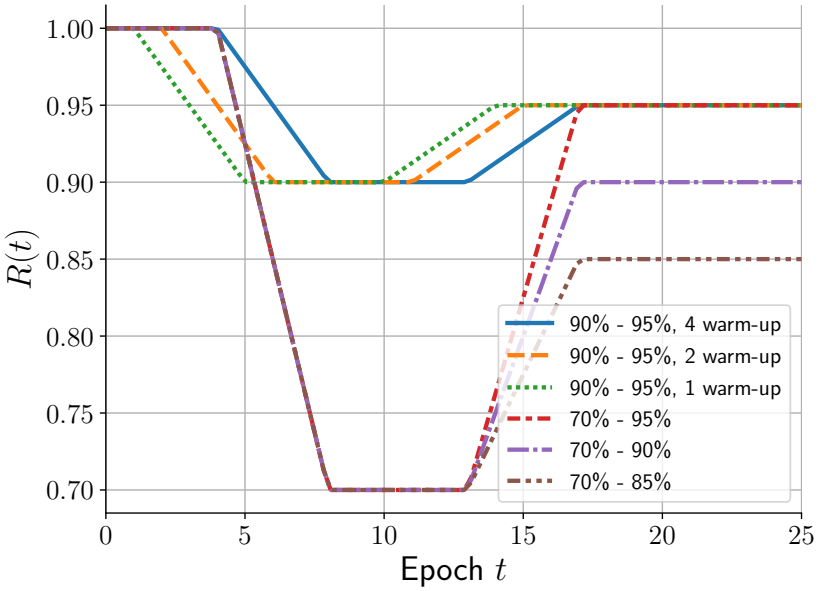


FIGURE 5.3: Tested $R(t)$ schedules.

$R(t)$ Bounds		Warm-up	Test IoU	Test Precision
70%	85%	4	0.3749	**0.9352**
70%	90%	4	0.8081	0.9021
70%	95%	4	0.8079	0.8961
90%	95%	4	**0.8214**	0.8940
90%	95%	2	0.8210	0.8970
90%	95%	1	**0.8214**	**0.8995**

TABLE 5.2: Co-Teaching U-Net results.

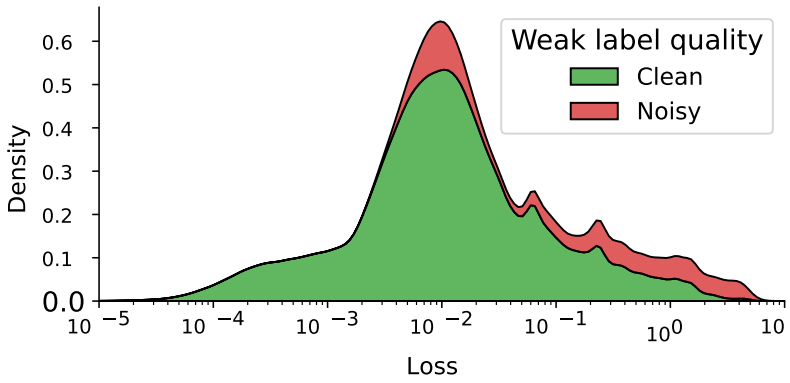


FIGURE 5.4: Distribution of U-Net pixel-wise loss after 2 training epochs on pseudo-labels.

5.7 Limitations of (Stochastic) Co-Teaching

The introduction of sampling during the training process in the stochastic variant results in performance gains. However, these gains are limited to a few percentage points. To understand why, we take advantage of the availability of ground truth labels in training data from Cityscapes. We train the same U-Net model used in previous experiments for 2 warm-up epochs using the entire pseudo-labels and depict the distribution of pixel-wise losses on Figure 5.4. This allows us to observe the distribution of noise with respect to the loss at the beginning of training, and analyze the impact of applying different training strategies in subsequent epochs.

Figure 5.4 illustrates an absence of noisy labels at low loss values, and a much larger proportion of wrong labels at high loss values. The Co-Teaching assumption that almost all noisy labels incur high loss values is not completely respected in this case. Indeed, non-negligible noise is also observed at median loss values. This empirical example validates the idea of sampling rather than using a fixed loss cutoff to reject likely noisy samples. Table 5.3 presents the noise statistics for training a third epoch using different strategies. When discarding 5% of pixels with the highest loss in each batch, Co-Teaching is able to discard 3.3% of the noise, while only removing 1.7% of the clean data. By sampling the training losses 10000 times and reporting mean noise statistics along with their standard deviation, we observe that Stochastic Co-Teaching is able to discard slightly more noise.

Using the knowledge that our training pseudo-labels contain 15.67% of noise, we show that using $R(t) = 0.85$ rejects a larger fraction of the noise, but also rejects more clean data. The fact that both Co-Teaching methods invariably sacrifice a small fraction of clean data explains why their Recall results are slightly worse than for Standard training in Table 5.1.

We remind the reader that our previous experiments were conducted without any use of the ground truth on the training and validation data, which prevents the use of such noise level estimates to set optimal co-teaching schedules in practice.

Training Strategy	Discarded Noise	Discarded Clean
Standard	0%	0%
Co-Teaching ($R(t) = 0.95$)	3.3%	1.7%
Stochastic Co-Teaching ($R(t) = 0.95$)	$3.6\% \pm 0.04\%$	$1.4\% \pm 0.04\%$
Co-Teaching ($R(t) = 0.85$)	7.2%	7.8%
Stochastic Co-Teaching ($R(t) = 0.85$)	$7.5\% \pm 0.11\%$	$7.5\% \pm 0.11\%$

TABLE 5.3: Noise statistics using different training strategies.

5.8 Conclusion

In this chapter, we introduce a novel approach for training a neural network to predict free space from images taken with a single road-facing camera. We train our models using pseudo-labels that are generated without expensive human annotations, and adapt Co-Teaching to our segmentation task in order to cope with label noise. To the best of our knowledge, our method is the first free space estimation approach to explicitly take label noise into account during training by using an adaptation of Co-Teaching. We also propose *Stochastic Co-Teaching*, a refinement that allows us to improve over results obtained with standard training and classical Co-Teaching procedures. By ensembling students trained with Stochastic Co-Teaching, we improve over standard training in both IoU (+1.1%) and Precision (+2.4%). Our best model reaches 87% of the IoU and 93% of the Precision of the fully-supervised competitor that trains from ground truth pixel-wise labels.

Chapter 6

Data Augmentation and Self-Training for Free Space Estimation

6.1 Introduction

Perception is the first step towards autonomous robot navigation. To be able to safely act in the world, a robot needs to perceive its environment and identify traversable free space. As explained in Chapter 4, in the context of autonomous driving, free space is usually defined as road areas that are not occupied by either static objects such as traffic signs and road dividers, or by dynamic entities such pedestrians and cars [Jan+20]. Since collision-free planning requires a fine-grained understanding of the environment around the vehicle, we attempt to label each pixel of a front-facing camera as traversable or not.

As detailed in Section 4.2, the large labor costs entailed by labeling of ground truth frames and the wide variety of environments and lightning conditions encountered at runtime make supervised segmentation difficult to scale for this task. Instead, we tackle it in a different way: relying on a method that generates noisy free space annotations without any supervision [Tsu+18], we train a neural network to generalize past the pseudo-label noise using data augmentation and recursive training. This approach is similar to the Co-Teaching experiments presented in Chapter 5 in that it uses the same pseudo-labels and focuses on improving training aspects. Rather than tweaking the training loop itself with Co-Teaching, this chapter explores different data augmentation scenarios, and analyzes how self-training impacts performances.

This chapter is based on the following published article:

François Robinet and Raphaël Frank. "Refining Weakly-Supervised Free Space Estimation Through Data Augmentation and Recursive Training". In: *Proceedings of BNAIC/BeneLearn 2021*. 2021

Following its conference presentation, this article was also selected by the organizing committee for publication in the following journal issue:

François Robinet and Raphaël Frank. "Refining Weakly-Supervised Free Space Estimation Through Data Augmentation and Recursive Training". In: *Artificial Intelligence and Machine Learning*. Springer International Publishing, 2022, pp. 30–45. ISBN: 978-3-030-93842-0. DOI: 10.1007/978-3-030-93842-0_2

The contributions presented in this chapter can be summarized as follows: (1) we study the impact of data augmentation on pseudo-supervised free space segmentation, (2) we propose a recursive training scheme that uses a progressively refined ground truth, (3) we improve existing results for unsupervised free space estimation on the Cityscapes dataset over previous efforts, gaining +2.3% in IoU, +2.4% in Precision, and +0.4% in Recall, (4) we discuss the limitations of our simple recursive training approach, and (5) we release our code and models to facilitate reproduction and further work.

To the best of our knowledge, these results established the state-of-the-art for monocular unsupervised free space estimation at the time of their submission. We later proposed the techniques discussed in Chapter 7 to further improve these results.

The remainder of this chapter is organized as follows: In Section 6.2, we review the recent literature for free space estimation, data augmentation in the context of semantic segmentation, and recursive training. In Section 6.3, we introduce our data augmentation and recursive training schemes. In Section 6.4, we describe our use of the Cityscapes dataset [Cor+15] and detail the experimental setup of this study. In Section 6.5, we carry out experiments and present the qualitative and quantitative results achieved. Finally, we summarize our contributions.

6.2 Related Work on Training Strategies for Pseudo-Supervised FSE

Related work on Free Space Estimation (FSE) has already been presented in Section 4.2, and an overview of methods that deal with label noise at training time is available in Section 5.2. Rather than repeating this information, this section focuses on regularization through data augmentation and self-training strategies.

Recent research shows that it is possible to train over-parametrized models to generalize past some of the label noise using Stochastic Gradient Descent (SGD) schemes combined with early stopping [LSO20]. Dealing with label noise at training time has become an important research area over the past few years. Solutions to this problem include label cleaning [Chi+19], noise-aware network architectures [Suk+15], or noise reduction through robust loss functions [MES08; Lu+17; Rob+20].

Besides work on training algorithms themselves, researchers have also largely explored regularization through data augmentation in unsupervised settings. Traditional augmentation strategies (scaling, color jittering, flipping, cropping, *etc.*) change pixel values in a single input image without altering its semantic content. More recently, researchers have proposed augmentations that combine several images and their labels. Two notable examples are MixUp [Zha+18] and CutMix [Yun+19]. MixUp is a method that augments the training set using convex combinations of image pairs and labels, while CutMix overlays random crops of other samples on top of original frames.

Since the output of models trained on pseudo-labels can be seen as stronger labels, another interesting research direction uses recursive training to iteratively refine segmentation results [DHS15; Kho+17].

6.3 Methodology

In this work, we train U-Net models to predict dense free space from RGB images by learning on approximate labels that can be generated without any supervision. Since our focus is on improving training aspects rather than on improving pseudo-labels generation, we will reuse the pseudo-labels from [Tsu+18]. We look at improving training across two dimensions: data augmentation and recursive training.

6.3.1 Data Augmentation

We study the impact of data augmentation on pseudo-supervised free space estimation. We cover both traditional augmentation techniques that operate on single images, as well as MixUp and CutMix, which are more recent and combine multiple samples.

Color-Flip-Crop To represent traditional augmentation techniques, we use a combination of color jittering, horizontal flips and random cropping, which we will refer to as *Color-Flip-Crop* or *CFC* in the remainder of the text. Each augmentation is independently applied with a 50% probability. The color jittering randomly affects brightness, contrast, saturation, and hue using the bounds defined in the Torchvision implementation [Tor]. In order to preserve most of the original image, cropping is performed with a randomly chosen rectangle that occupies between 25% and 50% of the image area. The aspect ratio is also randomly chosen, with the constraint that the height is at least 10% of the height of the original image. Figure 6.1 shows some examples of the effect of CFC on a single randomly chosen training image.

MixUp Rather than augmenting isolated images, Mixup trains models on convex combinations of samples [Zha+18]. By training on synthesized samples that lie between

FIGURE 6.1: Seven possible Color-Flip-Crop augmentations on a random training sample. The original sample is on the top-left. We show ground truth mask for illustration purposes, they are not used during training.

FIGURE 6.2: MixUp augmentation combining two random samples (a) and (b) from the training set. The convex combination using $\lambda = 0.5$ is shown as (c). We show ground truth mask for illustration purposes, they are not used during training.

the original training samples, MixUp encourages the network to exhibit a linear behavior between samples and helps preventing memorization. During training, each sample (x_1, y_1) is combined with another random sample (x_2, y_2) from the batch using Equations 6.1 and 6.2, where we sample λ uniformly in $[0, 1]$. The effect of combining input samples is illustrated on Figure 6.2.

$$x_{mixup} = \lambda x_1 + (1 - \lambda) x_2 \tag{6.1}$$

$$y_{mixup} = \lambda y_1 + (1 - \lambda) y_2 \tag{6.2}$$

CutMix Similar to Mixup in spirit, CutMix also combines two random input samples (x_1, y_1) and (x_2, y_2) from the same batch [Yun+19]. Rather than combining them over the entire image, CutMix overlays a crop of x_2 over x_1, and the same crop of y_2 over y_1. Equations 6.3 and 6.4 formalize this process using a random binary mask $M \in \{0, 1\}^{H \times W}$ to denote the cropped area ($\circ$ denotes the element-wise product). Like for the CFC augmentation, the cropping mask M occupies between 25% and 50% of the image area with a random aspect ratio. Figure 6.3 illustrates four different instances of CutMix augmentation on a chosen training sample. CutMix generates more natural images than MixUp and allows the network to learn more localizable features since the transformation is only applied to a fraction of the input image.

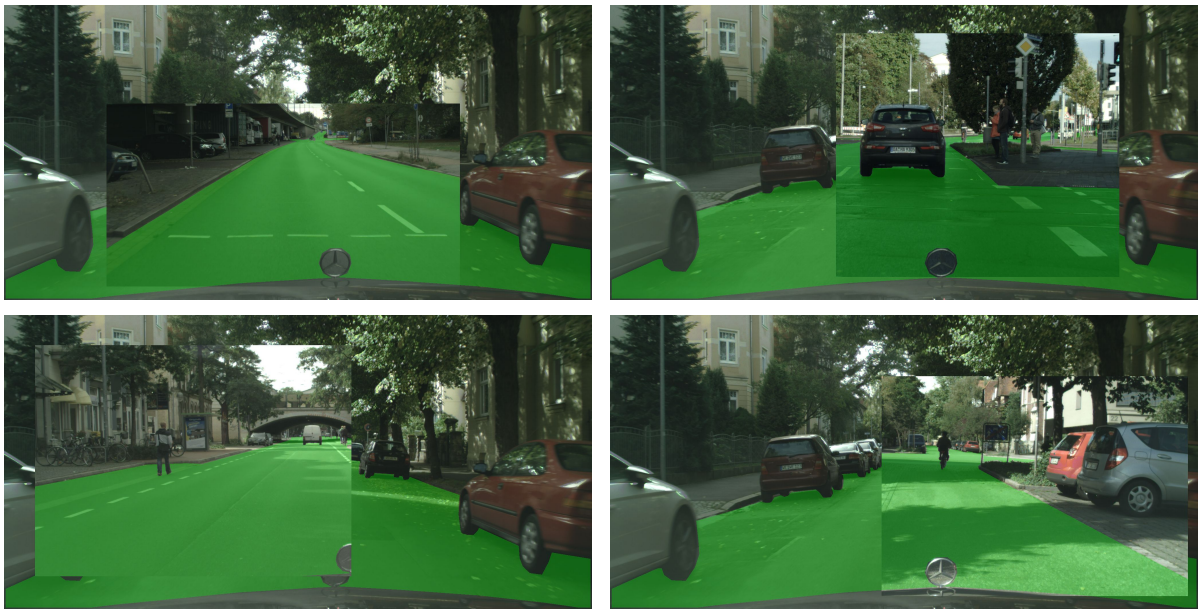


FIGURE 6.3: Four instances of the CutMix augmentation on a random training sample. We show ground truth mask for illustration purposes, they are not used during training.

$$x_{cutmix} = (1 - M) \circ x_1 + M \circ x_2 \tag{6.3}$$
$$y_{cutmix} = (1 - M) \circ y_1 + M \circ y_2 \tag{6.4}$$

6.3.2 Recursive Training

We are training neural networks to estimate free space by learning on approximate labels y_{weak}. Since neural networks trained with SGD variants are partially robust to noise in their training targets [LSO20], the outputs y will tend to approximate the unknown ground truth y^* better than y_{weak}. Assuming the outputs y are better estimates of free space than y_{weak}, it is natural to treat them as cleaner targets for a second round of training. This process can in principle be iterated to obtain progressively cleaner outputs y_2, y_3, *etc.* This approach was already attempted in the context of pseudo-supervised free space segmentation [TSK17], but we revisit its impact in the presence of data augmentation and with different pseudo-labels. Figure 6.4 illustrates the process for a given training round.

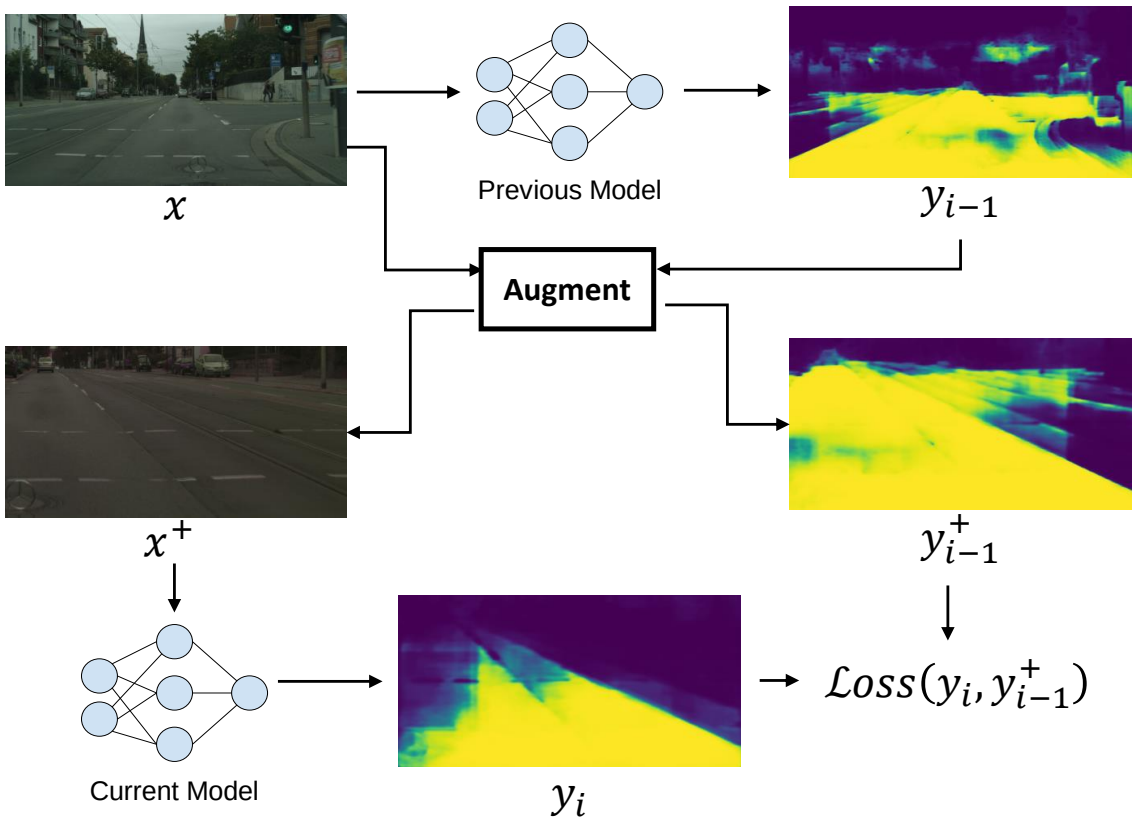


FIGURE 6.4: Recursive training procedure. The current model is trained on augmented outputs from the model obtained at the previous training round. In this example, CFC is used for augmentation. The process is similar for other augmentation strategies.

6.4 Experimental Setup

6.4.1 Network Architectures

Following recent research that shows that a well-tuned vanilla U-Net can outperform many refined variants on most segmentation tasks [Ise+18], we opt for a U-Net structure based on a ResNet18 residual network backbone [RFB15; He+16; Yak19]. To allow for comparison with prior art, we also implement and train the SegNet model described in [Tsu+18]. For computational reasons, we use a 512×1024 input resolution in all experiments. Outputs are however re-scaled using nearest neighbor interpolation in order to compute IoU and Precision in the original 1024×2048 resolution.

6.4.2 Training Procedure

We use the PyTorch framework [Pas+19] and train randomly initialized models to minimize a binary cross-entropy loss using the Adam optimizer [KB15], a batch size of 8 and an initial learning rate of 0.001. We train our models on single NVIDIA V100 for up to 200 epochs, with an early stopping strategy that halts training when the validation loss has not improved by at least 10^{-4} for 50 consecutive epochs. For each experiment, we select the model that minimizes the validation loss.

6.4.3 Use of Ground Truth Data

The Cityscapes dataset provides ground truth annotations for all training and validation frames used in this study. We stress that these annotations are only used to train the fully-supervised baseline for comparison with our pseudo-supervised approach. Outside of the fully-supervised experiment, ground truth labels are never used for training, hyper-parameter tuning, or to perform early stopping. Ground truth IoU, Precision and Recall are computed only once on the test set, after all these steps have been performed.

6.5 Results

This section describes the experiments carried out to benchmark our proposed method, using Precision, IoU and Recall. We present results for three main categories of models: 1) a fully-supervised upper-bound, 2) unsupervised and pseudo-supervised baselines, and 3) U-Nets trained on the pseudo-labels using recursive training and different augmentation strategies. The quantitative results for each category are summarized in Table 6.1. In this section, we analyze the results of each category, discuss the limitations of recursive training, and present qualitative results.

6.5.1 Fully-Supervised Results

Since Cityscapes provides pixel-wise ground truth annotations for our training and validation data, we use it to train a fully-supervised U-Net for comparison with its unsupervised counterpart. When trained on ground-truth labels, our U-Net model reaches high IoU (94.12%), Precision (97.26%) and Recall (97.27%). Since this fully-supervised model is the only one that uses ground truth labels at any point during training and validation, it is expected to produce an upper-bound for our unsupervised experiments. When using the CutMix augmentation strategy, IoU is slightly improved and reaches 94.54%.

6.5.2 Unsupervised and Pseudo-Supervised Baselines

Competing unsupervised approaches are often focused on generic semantic segmentation rather than free space estimation, and use other datasets than Cityscapes as benchmarks [DHS15; Xie+20; PC15; Dur+17; Cha+20]. Among unsupervised approaches that tackle free space estimation [HAS16; TSK17; Hof+16], only two publish results for Cityscapes. *Distant Supervision* [TSK17] and *Unsupervised Domain Adaptation* [Hof+16] respectively obtain an IoU of 80% and 70.4%, but do not report Precision or Recall values.

We generate approximate labels without supervision using the technique described in [Tsu+18]. Evaluating these raw pseudo-labels, we obtain an IoU of 79%, a Precision of 87.78% and a Recall of 89.24%. These results can be further improved by training a neural network to generalize beyond the noise in these labels. This was already attempted using the SegNet architecture in [Tsu+18], which we also implement and train for comparison. SegNet is able to improve results over raw pseudo-labels in IoU (+2.3%), Precision (+1.58%) and Recall (+0.91%).

6.5.3 Data Augmentation & Recursive Training

We train the same U-Net model using different data augmentation strategies. Since the outputs of our different augmented U-Nets are better than the initial pseudo-labels, we use them as target for a second round of training. We iterate this recursive training process four times for each of the data augmentation strategies under study. We limit training to four rounds for computational reasons and because it is enough for IoU values to reach their peak.

No Augmentation We start by training a U-Net with the pseudo-labels as targets and without any data augmentation. We observe that it compares favorably with the results from SegNet, reaching an IoU of 81.85%, a Precision of 90.65%, and a Recall of 89.76%. Without resorting to data augmentation, recursive training over several rounds is unable to meaningfully improve IoU, and slightly decreases Precision in favor of Recall.

MixUp Applying MixUp allows to improve Precision compared to not using data augmentation by 0.5% in the first training round. IoU is maintained, but Recall decreases by 0.45%. Iterative training is however not effective when combined with MixUp, since we observe a drop in Precision after each round. As discussed in Section 4.4.2, free space IoU and Precision are more important than Recall in an autonomous navigation scenario. In this case, increases in Recall are not enough to compensate this effect, and we observe a steady decrease in IoU.

Color-Flip-Crop Traditional data augmentation consisting of color jittering, horizontal flips and random cropping is able to improve IoU over not using augmentation and over using MixUp. After a single training round, CFC allows to reach an IoU of 81.99% through increasing Recall by 1.47% compared to the first round without augmentation. Subsequent training rounds are able to improve both Precision and IoU. After 3 iterations, the model reaches an IoU of 82.34% and a Precision of 90.75%.

CutMix The CutMix augmentation can be seen as providing the advantages of cropping and MixUp. Like MixUp, it synthesizes new input samples by combining pairs of existing ones. However, CutMix produces more natural images and its effect is localized since it only affects the area of a random crop. The locality of CutMix has been shown to allow models to learn more localizable features in classification scenarios [Yun+19], and it is not surprising that such features are helpful in this segmentation context. Indeed, models trained with CutMix augmentation outperform all other models by a wide margin. After a single training round, CutMix improves over not using augmentations in IoU (+1.2%), Precision (+0.5%), and Recall (+0.26%).

Since our application scenario favors Precision over Recall, our best overall model is obtained after the fourth training round, reaching an IoU of 83.64% and a Precision of 91.75%. Compared to the prior state-of-the-art results from SegNet [Tsu+18], it improves IoU by 2.3%, Precision by 2.4% and Recall by 0.4%. Although our model does not rely on any human-annotated ground truth, its relative performance compared to the fully-supervised variant is impressive: we reach 88.8% of its IoU, 94.3% of its Precision, and 93.1% of its Recall.

6.5.4 Limits of Recursive Training

While CutMix results are impressive, we note that the success of recursive training is limited. When not applying data augmentation or when using MixUp, recursive training does not improve on IoU or Precision. In the case of CFC and CutMix augmentations, results are more encouraging, but the improvements are limited to three rounds of training. Starting with the fourth round of training, IoU results start to degrade, sometimes getting worse than those obtained after a single round of training. Explaining this effect is not straightforward: given that target labels on round 4 are superior to those used on round 3 in both IoU and Precision, we would expect to either observe improved or plateauing

	Training/Validation Labels	Test IoU	Test Precision	Test Recall
Fully-Supervised U-Net	ground truth	94.12%	97.26%	97.27%
Fully-Supervised U-Net + CutMix	ground truth	94.54%	98.37%	94.27%
Unsup. Domain Adaptation [Hof+16]	synthetic data	70.40%	not reported	not reported
Distant Supervision [TSK17]	image labels	80.00%	not reported	not reported
Superpixel Clustering [Tsu+18]	no training	79.00%	87.78%	89.24%
SegNet (repr. from [Tsu+18])	pseudo-labels [Tsu+18]	81.30%	89.36%	90.15%
U-Net (no augmentation)				
Round 1	pseudo-labels [Tsu+18]	81.85%	90.65%	89.76%
Round 2	output of round 1	81.79%	89.53%	90.80%
Round 3	output of round 2	81.86%	90.15%	90.27%
Round 4	output of round 3	81.82%	90.11%	90.25%
U-Net + MixUp				
Round 1	pseudo-labels [Tsu+18]	81.89%	91.14%	89.31%
Round 2	output of round 1	81.97%	90.89%	89.60%
Round 3	output of round 2	81.62%	90.13%	89.97%
Round 4	output of round 3	81.45%	89.91%	90.02%
U-Net + Color-Flip-Crop				
Round 1	pseudo-labels [Tsu+18]	81.99%	88.80%	91.23%
Round 2	output of round 1	82.12%	89.71%	90.64%
Round 3	output of round 2	82.34%	90.75%	90.69%
Round 4	output of round 3	81.91%	90.21%	90.27%
U-Net + CutMix				
Round 1	pseudo-labels [Tsu+18]	83.05%	91.19%	90.51%
Round 2	output of round 1	83.58%	91.20%	91.12%
Round 3	output of round 2	**83.77%**	91.23%	**91.29%**
Round 4	output of round 3	83.64%	**91.75%**	90.62%

TABLE 6.1: Results on the Cityscapes validation set, which we treat as our test set. The best results for a given data augmentation strategy are underlined, and the best overall results are reported in bold.

results. Such recursive training strategy has been successfully used in foreground class segmentation contexts with results improving over more than 10 rounds [Kho+17]. As opposed to our completely unsupervised approach, the authors of [Kho+17] could exploit coarser ground truth in the form of bounding boxes in order to refine predictions after each round. We postulate that the absence of such refinement step in our approach is the reason we are unable to further leverage recursive training. Designing such a prediction refinement step will be the topic of future work.

6.5.5 Qualitative Results

We compare the free space estimates from pseudo-labels with the predictions of our best model on test set samples on Figure 6.5.

The ability of our learned model to generalize past some of the noise present in the pseudo-labels that were used during training is clearly visible in the first two rows of Figure 6.5. Indeed, the cars and side walks that were wrongly considered free space in the pseudo-labels are correctly predicted by our trained model. In addition to its higher

Precision, our model also has higher IoU and Recall, as illustrated by the near-absence of orange areas in its predictions.

The third row shows a more contrasted situation. Although our model is able to cover more free space, it still shows some signs of overfitting to noise in the pseudo-labels. Shadows are especially problematic because they are likely to impact the superpixel segmentation that the pseudo-labels are based on, resulting in missed free space areas such as the one present in front of the cyclist. Since this effect happens fairly consistently over the training set, our model is incapable of completely addressing it.

Finally, the fourth row illustrates another partial failure of our model in a particularly crowded scene. Compared to the corresponding pseudo-labels, the trained model correctly rejects pedestrians, but is unable to produce a clean segmentation around them and considers the pavement as occupied space. Although the prediction still contains errors, we note that red areas in our prediction are much more acceptable from a semantics point-of-view than the ones from the corresponding pseudo-labels.

6.6 Conclusion

In this work, we investigate different pseudo-supervised training strategies for teaching a neural network to predict free space from images taken with a single road-facing camera. Our models are trained using pseudo-labels that are generated without human intervention, and we investigate the impact of recursive training with several data augmentation schemes. We show that the CutMix augmentation is particularly efficient for free space estimation, especially when combined with recursive training. We benchmark our results on the Cityscapes dataset and improve over unsupervised and pseudo-supervised baselines, reaching 83.64% IoU (+2.3%), 91.75% Precision (+2.4%) and 91.29% Recall (+0.4%). Our best model obtains 88.8% of the IoU, 94.3% of the Precision and 93.1% of the Recall of the fully-supervised competitor that trains from expensive pixel-wise labels without CutMix. Finally, we show that simple recursive training is limited in its ability to increase performances, and suggest directions to improve the approach.

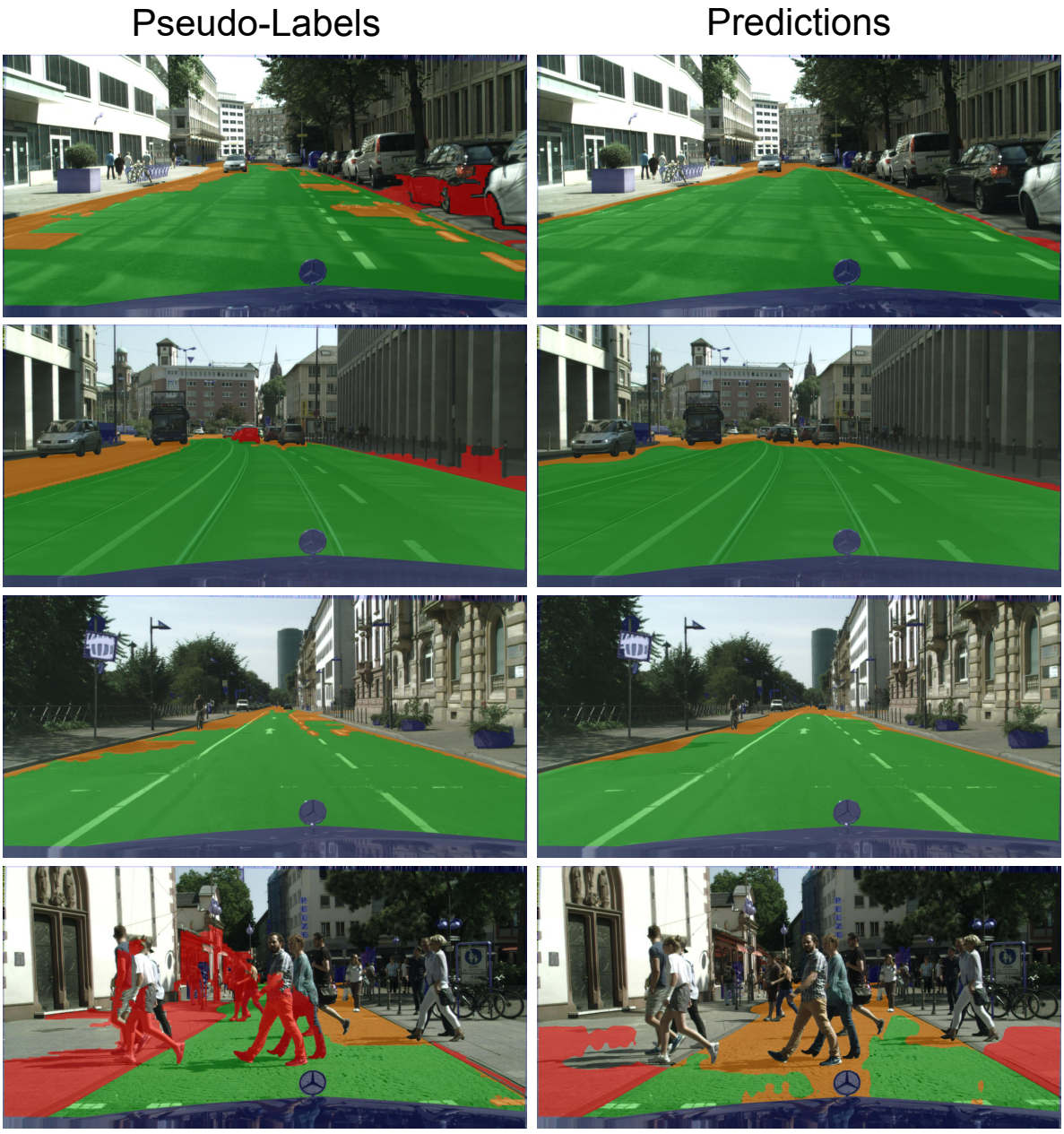


FIGURE 6.5: Qualitative results from the test set obtained from a U-Net trained with CutMix for 4 rounds. Predictions are color-coded using the ground truth: green and red respectively corresponds to correct and incorrect predictions, orange represents missing free space, and areas that are ignored at evaluation time are denoted in blue (see Section 4.4.2).

Chapter 7

Improving Pseudo-labels Generation for Free Space Estimation

7.1 Introduction

This chapter presents the first-half of the following article, published in the IEEE Robotics and Automation Letters, and presented at the IEEE IROS 2022 conference. The second half is covered in Chapter 8.

François Robinet, Yussef Akl, Kaleem Ullah, Farzad Nozarian, Christian Müller, and Raphaël Frank. "Striving for Less: Minimally-Supervised Pseudo-Label Generation for Monocular Road Segmentation". In: *IEEE Robotics and Automation Letters* (2022), pp. 1–7. DOI: 10.1109/LRA.2022.3193463

Like the previous two chapters, this chapter studies pseudo-supervised monocular free space segmentation. In Chapter 5, we examined how the training strategy could be adapted to use (Stochastic) Co-Teaching in order to explicitly cope with noise present in pseudo-labels. In Chapter 6, we proposed to combine the Cutmix and CFC data augmentation strategies with a recursive training scheme in order to further improve performance. While Chapters 5 and 6 relied on approximate labels obtained by superpixel clustering as described in Section 4.3, this chapter introduces a novel way of computing free space pseudo-labels.

We propose a practical and generic approach based on task-specific feature extraction. Building on recent advances in monocular depth estimation models, we process predicted dense depth maps to estimate pixel-wise road-plane distance maps. These maps are then used to generate pseudo-labels for a road segmentation scenario. This pseudo-labeling pipeline reaches state-of-the-art IoU (0.8529), while reducing complexity and computations compared to existing approaches. Although we test our work on road segmentation only, the proposed method is generic enough to apply to other less constrained-settings, such as indoor and outdoor ground robot navigation.

7.2 Related Work on Pseudo-labels Generation

Related work on free space segmentation has already been presented in Section 4.2. In this section, we specifically focus on methods that rely on generating approximate pseudo-labels using prior knowledge about road geometry or semantics. These approximate road masks can then be used as targets to train a statistical model to generalize past the noise that they contain.

One successful example, and an inspiration for this work, is the use of the v-disparity algorithm [LAT02]. This method uses disparity maps to estimate a flat road plane by making the generic assumption that the road-camera distance linearly increases as the road recedes to the horizon. Two main drawbacks prevent the direct use of the v-disparity approach in practical applications: (1) the assumption of a perfectly planar road is sometimes violated in practice and, (2) the reliance on very high quality disparity maps makes this method very sensitive to errors in these inputs. Similar to prior work, we use this technique to generate approximate road masks, before training a deep neural network to generalize past the noise they contain [HAS16; MUT18].

Our work differs in that we propose to approximate these disparity maps using a monocular depth estimation network which can be trained without stereo cameras or supervision [God+19]. More direct use of depth measurements can also be beneficial, as shown through the use of RGB-D inputs to improve performance in indoor perception [WSL19]. Also exploiting geometric information present in depth maps, work from Seichter et al. uses a trained network specifically designed to extract features from RGB-D images by fusing depth feature maps at various stages of encoding [Sei+21].

Rather than explicitly relying on scene geometry, another successful method is to identify the road by over-segmenting frames into superpixels and extracting feature vectors for each superpixel using a network trained for generic image classification. Superpixels can then be clustered in feature-space before using a spatial prior to identify the cluster corresponding to the road [Tsu+18; Rob+22a; RF22]. This *Superpixel Clustering* approach was covered in greater details in Section 4.3.

Our unsupervised approach unifies the geometrical and semantic approaches by combining features extracted from a v-disparity representation with semantic cues obtained from over-segmenting the RGB space into superpixels.

7.3 Methodology

In this section, we start by detailing how we obtain our dense Road Plane Distance (RPD) maps. We then show how we leverage these estimated distances to obtain pseudo-labels, which can then be used to train predictive models.

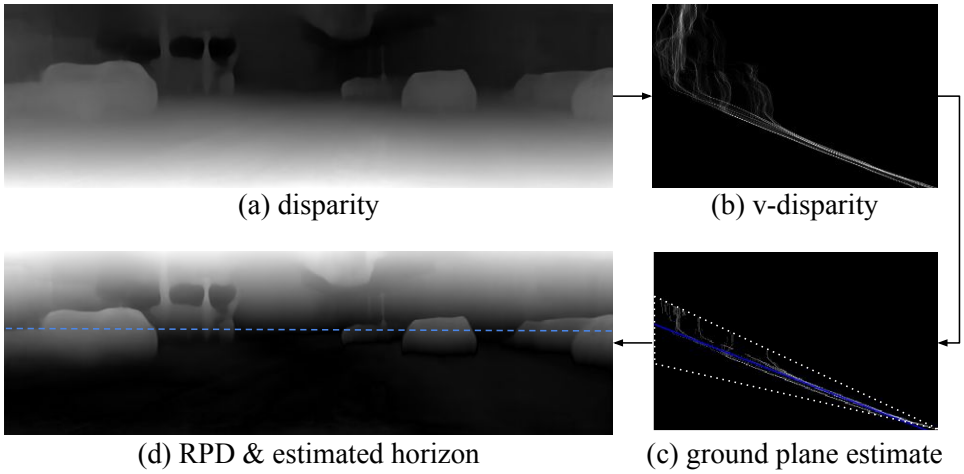


FIGURE 7.1: Estimation of Road Plane Distance (RPD) from dense disparity maps using robust line fitting in v-disparity space.

7.3.1 Estimating Road Plane Distance (RPD)

To estimate the road plane, we rely on the v-disparity algorithm, which was first proposed in the context of obstacle detection [LAT02]. The algorithm does not operate on raw RGB images, but instead takes dense disparity maps as inputs. Rather than relying on depth reconstruction using stereo pairs [Hir08; MUT18], we propose to use a monocular depth estimation network to estimate such disparity maps from a single view of the scene. For this purpose, we have chosen a Monodepth2 network [God+19] that was trained on the KITTI dataset [GLU12] without any ground truth labels, using Structure-from-Motion [Zho+17].

Starting from a disparity map with dimensions (H, W) in Figure 7.1(a), a v-disparity map with dimension (H, B) is obtained by creating a B-bin histogram of disparity values in each row, as shown on Figure 7.1(b). The road detection procedure builds on the following intuition: oblique planes in the disparity input are mapped to straight lines in v-disparity space. Assuming that free space is the dominant planar region in the disparity input, one can therefore approximate it through line-fitting in v-disparity space. This process is illustrated on Figure 7.1(c). Since obstacles appear as vertical lines in v-disparity space, it is important to filter data before attempting line fitting. We follow [HAS16] and only keep bin values belonging to the 95% percentile in each row. Since the road plane maps to a line in v-disparity space, the intercept of that line corresponds to an estimate of the horizon line in the original image. Exploiting prior knowledge about camera placement, we further restrict the area of interest to only match common horizon lines appearing on the frame between 0.8 and 0.4 relative heights. We fit the filtered v-disparity points using RANSAC linear regression [FB81].

As opposed to prior work [LAT02; MUT18; HAS16], we do not back-project the fitted plane to the disparity image by reverting the histograms to obtain a road plane estimate. Rather, for each pixel in the disparity space, we estimate its elevation relative to the ground plane by computing the horizontal distance between its v-disparity projection and

the fitted ground line. We call the result a Road Plane Distance map (RPD) and represent it on Figure 7.1(d).

7.3.2 From RPD Maps to Pseudo-labels

In this section, we explain our use of RPD maps to generate approximate pseudo-labels of the road class. Our unsupervised pseudo-label generation method is illustrated on Figure 7.2.

Superpixel cues Since object parts that touch the ground will contain low RPD values near the ground, directly thresholding RPD values cannot result in precise class boundaries. Instead, we rely on superpixel segmentation in RGB space to recover crisp boundaries. For a given RGB input, we generate a normalized RPD map and compute a superpixel segmentation using the Felzenszwalb method with a scale of 50 and a minimal size of 500 pixels [FH04]. We then aggregate the RPD values over each RGB superpixel using its 90% RPD quantile. We select the 90% quantile rather than the maximum value in an effort to capture the highest RPD values without being overly affected by possible outliers. An analysis of the impact of quantile choice is provided in Section 7.5.3.

Adaptive RPD thresholding In order to obtain pseudo-labels, we apply a threshold on superpixel aggregated RPD maps. Rather than choosing a fixed threshold to apply over all frames, we propose an adaptive procedure based on the distribution of RPD values in each frame. We make the assumption that most road pixels lie in the lower half of the frame, and compute a frame-specific threshold based on their distribution. We compute a smooth approximation of the distribution of RPD values in the lower portion of the frame using gaussian kernel density estimation. Since road superpixels have uniform and low RPD values, they will tend to form the largest peak in this distribution, and we identify the threshold as the first local minimum following this peak. In order to obtain our final pseudo-labels, we also remove any pixel lying above the estimated horizon from the road prediction.

Pseudo-supervised training We use the pseudo-labels as targets to train a model to segment the road directly from an RGB input. Recent work has shown that deep neural networks trained via stochastic gradient descent exhibit surprising robustness to noise in their training targets [LSO20], and such procedure has repeatedly been proven beneficial to road segmentation [MUT18; Tsu+18; RF22; Rob+22a].

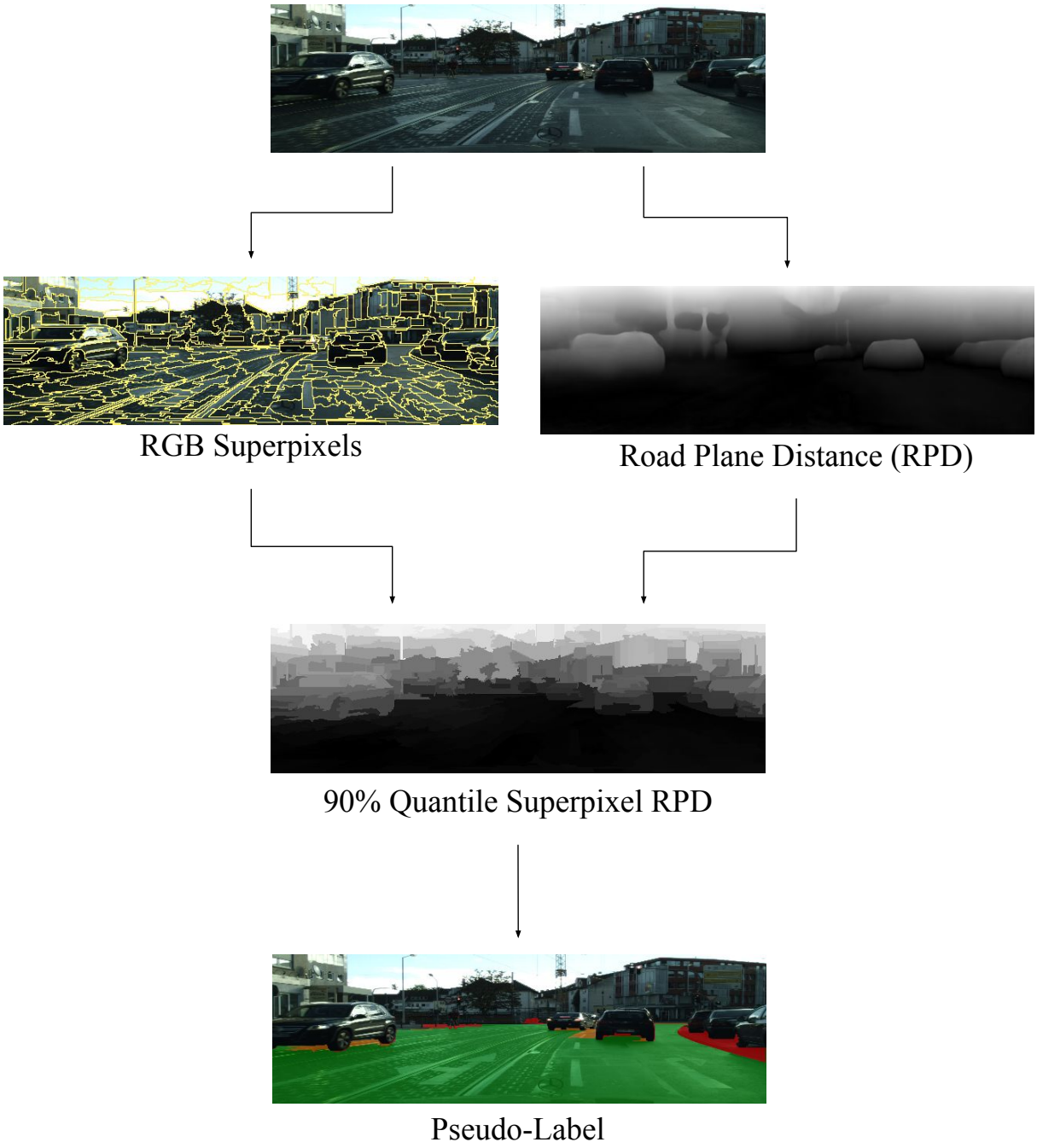


FIGURE 7.2: Unsupervised pseudo-labels generation. RPD quantiles are computed over each RGB superpixels, and the result is thresholded and filtered to obtain a road estimate.

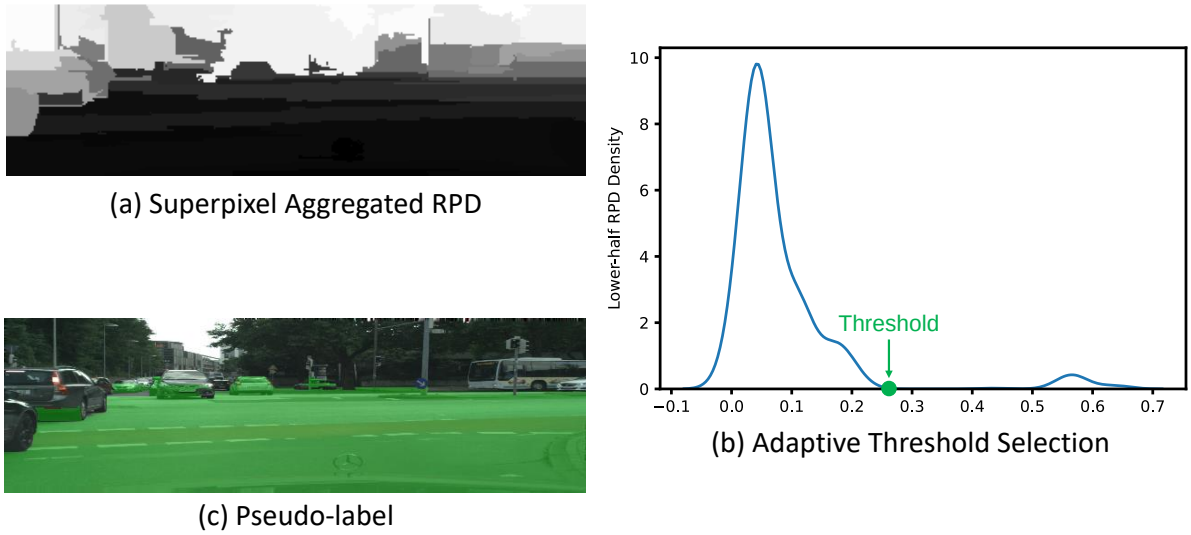


FIGURE 7.3: Adaptive thresholding procedure.

7.4 Experimental Setup

To allow for fair comparison and analysis of the impact of using our pseudo-labels compared to existing ones, all our experiments use the same architecture as in our Co-Teaching and data augmentation experiments from Chapters 5 and 6. We again rely on a U-Net architecture based on a ResNet18 residual network backbone [RFB15; He+16]. The unsupervised and fully-supervised models are trained to predict road masks from RGB frames and all have 14.3M parameters.

We train randomly initialized models to minimize a binary cross-entropy loss using a batch size of 4. We set an initial learning rate of 10^{-4} and decay it by half when the training loss plateaus for at least 25 epochs. We train our models on a single NVIDIA V100 for up to 500 epochs, with an early stopping strategy that halts training when the validation loss has not improved by at least 0.0003 for 75 consecutive epochs.

We use the Cutmix data augmentation strategy detailed in Section 6.3.1 for all of our models. For models that use RGB as inputs, we also apply the Color-Crop-Flip (CFC) augmentation strategy, also described in Section 6.3.1. CFC applies a color jitter, takes a random crop of appropriate aspect ratio, and randomly perform an horizontal flip of the input. Each augmentation occurs with 50% probability.

For each experiment, we select the model that minimizes the validation loss. For computational reasons and to match the Monodepth2 input shape, we use a 192×640 input resolution in all experiments. Outputs are however re-scaled using nearest neighbor interpolation in order to evaluate metrics in the original 1024×2048 resolution.

7.5 Evaluation & Results

Like in previous chapters, we evaluate our method using the Cityscapes dataset described in Section 4.4.1 to be able to compare it to existing work. To obtain a complete picture of prediction quality, we measure Intersection-over-Union (IoU), Precision and Recall. As in the official Cityscapes benchmark, the metrics are computed using an evaluation mask that ignores pixels labeled as *void*. The complete evaluation procedure is detailed in Section 4.4.2. This section outlines the set of experiments carried out to benchmark the proposed method.

7.5.1 Results

Road Classes: only road	**IoU**	**Precision**	**Recall**
Superpixel Clustering [Tsu+18; RF22]	0.8152	0.8854	0.9138
Co-Teaching Superpixel Clustering [Rob+22a]	0.8261	0.9093	0.9027
Cutmix Superpixel Clustering [RF22]	0.8377	**0.9193**	0.9129
Ours (unsupervised)	**0.8529**	0.8827	**0.9623**
Road Classes: road, ground & parking	**IoU_3**	**Precision_3**	**Recall_3**
Stereo v-disparity [MUT18]	0.8001	**0.9283**	0.8529
Ours (unsupervised)	**0.8600**	0.8943	**0.9595**

TABLE 7.1: Test set results for unsupervised road segmentation. The second part of the table evaluates using a different definition of road that includes the *road*, *parking* and *ground* classes to allow comparison with [MUT18]. Metrics are suffixed to emphasize that three classes are used as part of the road in that second scenario.

In Table 7.1, we compare our model trained on unsupervised pseudo-labels (described in Section 7.3.2) to other recent unsupervised methods. All approaches use a similar strategy of generating pseudo-labels before training a neural network to generalize part of the label noise away.

The Stereo v-disparity method [MUT18] is closest in spirit to our approach since it also exploits disparity maps. The notable differences are that they obtain disparity maps through stereo depth reconstruction rather than from a monocular depth estimation network, and they simply threshold in v-disparity space to obtain a road mask, while we use cues from the RGB image in the form of superpixel segments. The other three methods [Tsu+18; Rob+22a; RF22] rely on the Superpixel Clustering pseudo-labeling strategy, and were respectively presented in Sections 4.3, 5.3 and 6.3.

Our method does not need stereo pairs, but still achieves the highest IoU (0.8529) and Recall (0.9623). The large increase in Recall however comes at the cost of Precision, as expected for a method that detects the ground plane rather than the road. The source of imprecision is indeed mostly attributable to ground-level pixels being wrongly classified as *road* when they are actually *sidewalks* or *parking*. When including such *flat* classes

Superpixel aggregation	Adaptive thresholding	IoU	Precision	Recall
–	–	0.7456	0.7535	**0.9875**
✓	–	0.8432	0.8683	0.9664
✓	✓	**0.8529**	**0.8827**	0.9623

TABLE 7.2: Ablation study of trained models test results.

from Cityscapes as part of our definition of the road, the Precision rises to over 98%. This shows the potential of our approach for different robotics applications where any flat surfaces can be traversed.

To identify the contributions of superpixel aggregation and adaptive thresholding, Table 7.2 presents an ablation study. In the absence of adaptive thresholding, a single fixed threshold is selected for all the frames by visually examining RPD maps for 10 random training frames. The manual threshold is set to 0.075 when no superpixel aggregation is used, and to 0.1 when it is enabled. Note that although this manual selection technically introduces human-supervision, it is only used for investigating the impact of adaptive thresholding. Due to noise in the estimated depth and imprecisions in the road plane fit used to build RPD maps, the absence of superpixel aggregation causes a severe lack of Precision.

7.5.2 Inference Time

The models used in our unsupervised and semi-supervised methods are also computationally efficient, since they use the same architecture but a lower input resolution of 192×640 compared to the 512×1024 inputs used in other approaches [Tsu+18; Rob+22a; RF22; MUT18]. Table 7.3 illustrates the differences in both inference time and Multiply-Accumulate operations (MACs) required in a forward pass for a single input frame. The inference times include the copy of the frame to GPU memory and of the result back to CPU memory.

Input Resolution	Inference Time	GMACs
192×640	$3.56\ \text{ms} \pm 5.65\ \mu\text{s}$	10.16
512×1024	$10.42\ \text{ms} \pm 4.29\ \mu\text{s}$	43.37

TABLE 7.3: Inference times and Mutiply-Accumulate operations on a NVIDIA Tesla V100. For time measurements, we report mean and standard deviation over 1000 runs.

7.5.3 Impact of RPD Quantile Choice

The unsupervised pseudo-label generation described in Section 7.3.2 relies on RGB superpixel cues to reach the best performance. The values of RPD maps are aggregated over

RGB superpixels using a quantile function in order to discard some of the noise. In Table 7.4, we study the impact of changing the quantile value on IoU, Precision and Recall. Note that these results are for raw pseudo-labels. They can therefore not be directly compared to the IoU results presented in Table 7.1 and Table 7.2, which correspond to a trained model. They are however indicative of relative performance between quantile choices. Since evaluating different quantile choices on the test set technically constitutes a fit, this analysis is only provided for additional insights. The decision to use the 90% quantile in our methodology was made prior to such evaluation, by visually inspecting the RPD maps of a few random training frames to decide on an appropriate value.

The exact choice of the quantile does not have a large impact on IoU, but influences the Precision-Recall trade-off, with lower values favoring Recall since more superpixels tend be classified as part of the road. The choice of the 100% quantile is equivalent to aggregating a superpixel by its maximum RPD value, and is detrimental to IoU since it is highly sensitive to large RPD outliers.

RPD Quantile	IoU	Precision	Recall
50%	0.7836	0.8140	**0.9532**
60%	0.7890	0.8283	0.9422
70%	0.7939	0.8307	0.9471
80%	**0.7984**	0.8357	0.9475
85%	0.7983	0.8441	0.9360
90%	0.7957	0.8548	0.9214
95%	0.7963	0.8653	0.9104
100%	0.7625	**0.8661**	0.8779

TABLE 7.4: Test results for unsupervised raw pseudo-labels using adaptive thresholding with different quantile values.

7.6 Conclusion

This chapter investigates a novel unsupervised pseudo-labeling strategy in order to train a neural network to segment the road using images captured by a single road-facing camera with no manual annotations.

In order to minimize the labeling efforts required to train these models, we devise a novel feature extraction method based on the v-disparity algorithm. To the best of our knowledge, our approach is the first to compute approximate v-disparity maps using a depth prediction network, as well as to use them to compute dense Road Plane Distance (RPD) maps.

Our approach fuses the geometrical information from RPD maps with semantical cues obtained from over-segmenting the RGB input into superpixels. We obtain pseudo-labels by combining RPD maps and RGB superpixels, and these are then used as targets to

train a neural network in a completely unsupervised way. Unlike previous work, our unsupervised method does not require the use of stereo-pairs but still reaches state-of-the-art results while also being less computationally-intensive.

The lack of Precision of RPD-based pseudo-labels is attributable to the fact that not all flat surfaces (*e.g.* sidewalks, parkings, ...) should be considered free space in an autonomous driving scenario. We expect that this shortcoming would not be so important in other scenarios such as indoors perception, or robotic navigation in less unconstrained outdoor environments such as buildings, parks, fields or forests.

While this chapter discusses the direct use of RPD maps for unsupervised road segmentation, Chapter 8 will show how they can be used as task-specific features in a semi-supervised setting, and explain how the addition of a minimal amount of labels can help overcome limitations in Precision.

Part III

Semi-Supervised Free Space Estimation

Chapter 8

Minimally-Supervised Pseudo-Labels for Free Space Estimation

8.1 Introduction

This chapter presents the second-half of the following article, accepted for publication in the IEEE Robotics and Automation Letters, and for presentation at the IEEE IROS 2022 conference. The first half was covered in Chapter 7.

François Robinet, Yussef Akl, Kaleem Ullah, Farzad Nozarian, Christian Müller, and Raphaël Frank. "Striving for Less: Minimally-Supervised Pseudo-Label Generation for Monocular Road Segmentation". In: *IEEE Robotics and Automation Letters* (2022), pp. 1–7. DOI: 10.1109/LRA.2022.3193463

To be able to safely navigate the world, autonomous robots have to be capable of perceiving their environment to detect traversable free space. Like in previous chapters, we focus here on road detection for vehicles equipped with a single forward-facing camera. While this task has traditionally been solved using supervised segmentation methods, these techniques suffer the drawbacks of laborious labeling cost (1.5h/frame for fine annotations [Cor+15]), as well as test-time distribution shift due to the wide variety of environments and weather conditions [Tre+18]. To alleviate these issues, the community has recently focused on unsupervised and semi-supervised alternatives.

Although unsupervised approaches lead to respectable results, they are not reliable enough to enable the safe operation of autonomous agents [HAS16; Tsu+18; MUT18; Rob+22a; RF22]. Semi-supervised methods can offer a practical compromise by approaching the performance of full supervision while training on restricted subsets of manually-labeled data. Such performance is usually achieved by exploiting unlabeled data through self-supervised consistencies or pseudo-labeling [Che+20a; Hoy+21; TV17].

The work presented in this chapter explores this trend by expanding on the RPD pseudo-labeling framework presented in Chapter 7, in order to incorporate a restricted subset of ground truth labeled samples. Our method reduces the quantity of labeled data

required by exploiting prior knowledge of road semantics as well as recent advances in monocular depth estimation [God+19].

Relying on the RPD maps described in Chapter 7, we design a semi-supervised extension, which uses as little as 1% (respectively 10%) of ground truth data while reaching 96.1% (resp. 97.8%) of the IoU obtained by a comparable fully-supervised model.

As discussed in Section 7.5, the unsupervised RPD-based pseudo-labels described in the previous chapter are only able to evaluate whether space is physically traversable, but cannot deal with traffic rules. For example, ground-level sidewalks or vegetation are often mislabeled as the road in our unsupervised pipeline. This type of error is largely fixed by labeling a small amount of ground truth samples in the semi-supervised approach described in this chapter, which results in a Precision increase of +9.08% when labeling only 29 frames.

While our work focuses on urban scenes, the use of an unsupervised monocular depth network and RPD maps are generic enough to allow the same techniques to be used in other robotic navigation scenarios, such as indoors or less-constrained outdoor environments.

The content of this chapter is organized as follows: In Section 8.2, we review existing works on semi-supervised road segmentation. In Section 8.3, we introduce our method for computing our semi-supervised extension to the RPD-based method proposed in Chapter 7. Our experiments are detailed in Section 8.4 and we analyze their results in Section 8.5. Finally, we summarize our contributions in Section 8.6.

8.2 Related Work on Semi-Supervised FSE

This section presents a brief overview of road segmentation approaches that learn pixel-wise free space representations from data. While related works rely on video sequences to learn sparse point clouds [Dav+07], or to segment obstacle footprints using structure-from-motion [Wat+20], we focus our attention on unsupervised and semi-supervised methods that learn road masks using single images.

Recent advances in semi-supervised segmentation can mainly be divided in two categories of approaches: consistency training and self-training using pseudo-labeling.

Consistency Regularization Consistency regularization obtains a loss by making the assumption that the output of a model should be similar for perturbations of the same input that do not affect its semantics. Proposed approaches differ in the perturbations of the input that they use, or in the definition of similarity that they optimize for. Barlow Twins [Zbo+21] is a recent method that attempts to make learned embeddings for two distinctly distorted inputs be similar but non-redundant, by making their cross-correlation matrix be close to the identity matrix. Another similar approach is SimCLR, where two

random augmentations are computed for the same input, but a contrastive loss is used for training the features encoder [Che+20b]. Other examples that fall into this category are Unsupervised Data Augmentation [Xie+19], Virtual Adversarial Training [Miy+17] and Cutmix [Yun+19].

Pseudo-labeling The semi-supervised part of this work adopts the second approach of self-training with pseudo-labels, in particular through the use of a mix of ground truth labels and noisy pseudo-labels that are generated for unlabeled data [Che+20a; Hoy+21]. Recent research has shown that over-parameterized neural networks can generalize past some of the label noise using Stochastic Gradient Descent (SGD) schemes combined with early stopping [LSO20], but more advanced schemes have been devised to explicitly deal with this noise, such as Mean Teacher [TV17] and Co-Teaching [Han+18]. For a comprehensive overview of semi-supervised techniques that cope with noisy labels in image analysis, we refer the reader to the survey in [Kar+20].

Note that these two categories of approaches are not necessarily mutually exclusive. FixMatch is a recent attempt at combining the ideas of consistency regularization and pseudo-labeling [Soh+20].

8.3 Methodology

Although RPD maps are useful to detect the ground plane, the road class is more restrictive since it should not contain things like sidewalks or grass patches, even though they might lie in the same plane. Since RPD maps do not contain this information, we propose to combine them with RGB frames and to learn from a minimal number of ground truth annotations.

8.3.1 Pseudo-Label Generator (PLG)

To generate pseudo-labels in the semi-supervised scenario, we train a PLG network to predict road masks from RPD maps and RGB frames. We concatenate them and use them as inputs to our PLG, which will allow it to learn to segment the road from only dozens of ground truth samples. This process is illustrated on Figure 8.1.

8.3.2 Training from Semi-Supervised Pseudo-Labels

As in the unsupervised case from Section 7.3.2, an additional model can be trained using the semi-supervised pseudo-labels as targets in order to obtain road masks from RGB inputs. For frames for which ground truth was used at pseudo-label generation time,

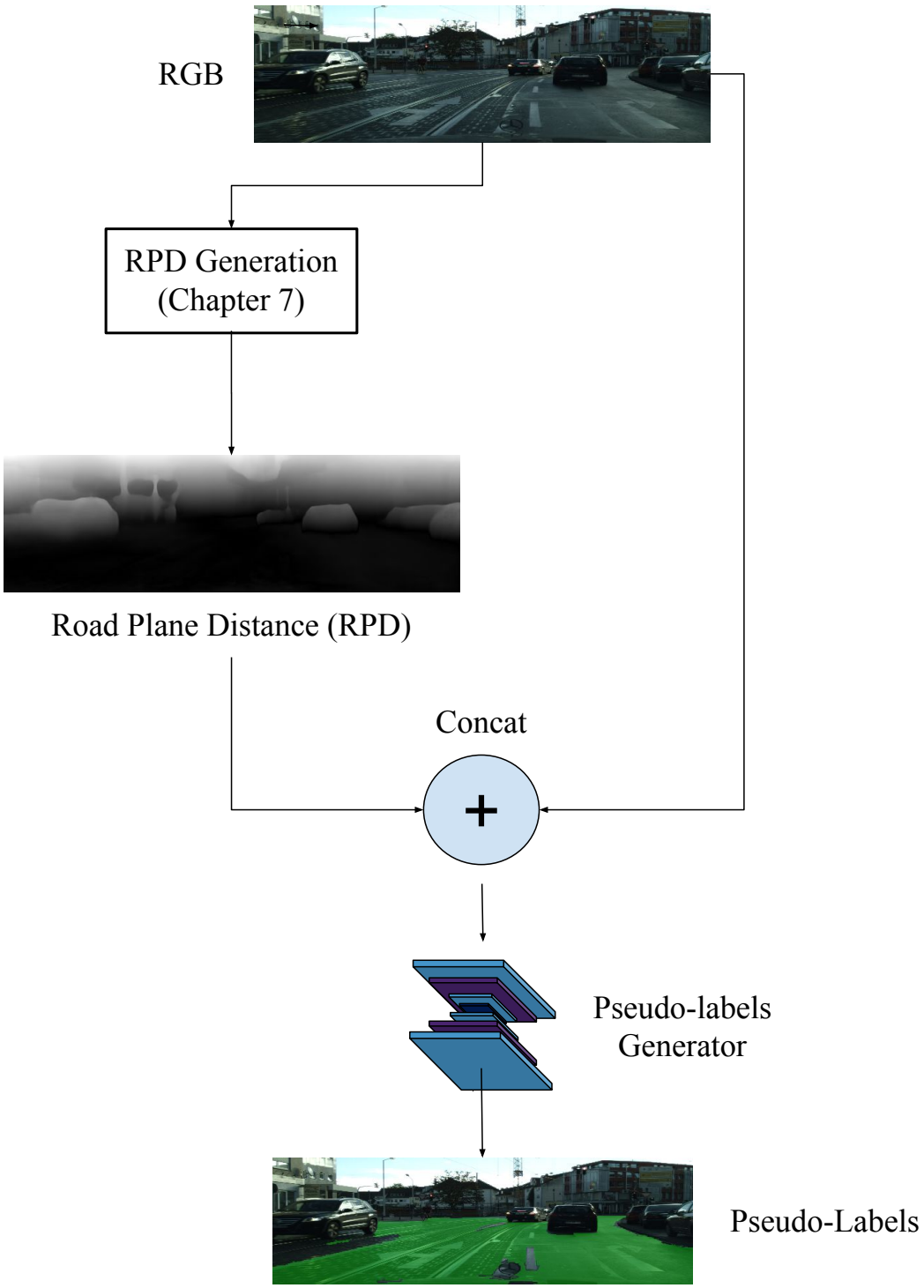


FIGURE 8.1: Semi-supervised pseudo-labels generation procedure. We train a pseudo-label generator to predict pseudo-labels using RGB frames and RPD maps. The network learns using a small amount of ground-truth annotated frames.

we also use pixel-wise ground truth as our target. Details about the model and training procedure used are available in Section 8.4.2.

8.4 Experimental Setup

This section outlines the set of experiments carried out to benchmark the proposed method. To allow for fair comparison with the results presented in Chapters 4 to 7, we again evaluate our method on the Cityscapes dataset described in Section 4.4.1, using IoU, Precision and Recall. A mathematical description of these metrics, and which pixels are ignored during evaluation is given in Section 4.4.2. This section details our network architecture and training procedure.

8.4.1 Network Architectures

Competing semi-supervised approaches are often focused on generic semantic segmentation rather than road segmentation, or use other datasets than Cityscapes as benchmarks [DHS15; Xie+20; PC15; Dur+17; Cha+20]. The few semi-supervised approaches that publish metrics for the *road* class use a wide variety of network architectures, input resolutions, computational requirements, annotations (all classes or road only) and data splits, making direct comparison difficult. To allow for fair comparison and analysis of the impact of using semi-supervised pseudo-labels instead of training from only limited ground truth, all our experiments use the same architecture.

Recent work has shown that properly tuned standard U-Nets can outperform more advanced variants in many segmentation scenarios [Ise+18], and we therefore opt for a standard U-Net architecture based on a ResNet18 residual network backbone [RFB15; He+16]. The fully-supervised and semi-supervised models trained to predict road masks from RGB frames all have 14.3M parameters.

We also use the same architecture for the semi-supervised PLG from Figure 8.1, but we slightly adapt its first layer to accept the additional channel from RPD in addition to the three RGB channels.

8.4.2 Training Procedure

We train randomly initialized models to minimize a binary cross-entropy loss using a batch size of 4. We set an initial learning rate of 10^{-4} and decay it by half when the training loss plateaus for at least 25 epochs. We train our models on a single NVIDIA V100 for up to 500 epochs, with an early stopping strategy that halts training when the validation loss has not improved by at least 0.0003 for 75 consecutive epochs.

We use the Cutmix data augmentation strategy detailed in Section 6.3.1 for all of our models. For models that use RGB as inputs, we also apply the Color-Crop-Flip (CFC) augmentation strategy, also described in Section 6.3.1. CFC applies a color jitter, takes a random crop of appropriate aspect ratio, and randomly perform an horizontal flip of the input. Each augmentation occurs with 50% probability.

For each experiment, we select the model that minimizes the validation loss. For computational reasons and to match the Monodepth2 input shape, we use a 192×640 input resolution in all experiments. Outputs are however re-scaled using nearest neighbor interpolation in order to evaluation metrics in the original 1024×2048 resolution.

8.5 Results

Cityscapes provides pixel-wise ground truth annotations for our 2380 training and 595 validation frames, which allows us to examine the fully-supervised scenario in the first section of Table 8.1. Our supervised baseline that uses 100% of the ground truth reaches an IoU of 0.9454. Using a higher input resolution, a larger model, and pseudo-labels generated for additional frames not included in our dataset, the Naive-Student Video Sequence model is able to further improve IoU to 0.9882 [Che+20a]. We do not include the latter approach in Table 8.1 since it uses ground truth annotations for all Cityscapes classes on 2975 frames and pseudo-labels generated for an additional 109 thousand frames, making a direct comparison with our results impossible.

The second and third sections of Table 8.1 present models trained on a small fraction of ground truth labels, with and without adding pseudo-labels for the remaining frames. Using 10% annotated samples, our semi-supervised model is able to achieve an IoU of 0.9332. This corresponds to a +3% increase over a supervised baseline trained on the same ground-truth frames without pseudo-labels. When dropping the fraction of labeled samples to 1%, our proposed method improves IoU by +7.25% compared to its supervised counterpart and reaches 0.9063 IoU.

These results motivate a pragmatic approach to data annotation for free-space segmentation tasks: even a minimal labeling effort can greatly improve results and enable rapid prototyping for robotics applications. Indeed, annotating only 29 frames allows to increase Precision by +8.74% and IoU by +5.34% over the best unsupervised method of Table 7.1. Although annotating more data is always beneficial, it is also important to notice that semi-supervised models using 1% and 10% of labeled data respectively achieve 95.9% and 98.7% of the IoU obtained using a comparable model trained on 100% of the ground truth.

To assess the impact of using both RGB frames and RPD maps as inputs to our semi-supervised PLG, we conduct an ablation in Table 8.2. The results shows that using RGB alone decreases IoU by 8.7% when using 1% labeled frames. The impact of RPD maps is less extreme when using 10% ground truth annotations, which is expected since the model can learn to extract RPD information from RGB frames alone if given enough

	GT % (frames)	PLG pseudo-labels	IoU	Precision	Recall
Supervised baseline	100% (2975)	0	**0.9454**	**0.9837**	**0.9427**
Supervised baseline	10%	0	0.9032	0.9614	0.9353
Ours (semi-supervised)	(297)	2678	**0.9332**	**0.9807**	**0.9493**
Supervised baseline	1%	0	0.8338	0.9174	0.9009
Ours (semi-supervised)	(29)	2946	**0.9063**	**0.9701**	**0.9310**

TABLE 8.1: Test set results for fully-supervised and semi-supervised road segmentation. The best results for each level of supervision are reported in bold.

GT %	PLG Inputs RGB	PLG Inputs RPD	IoU	Precision	Recall
1%	✓	–	0.8193	0.9277	0.8743
	–	✓	0.8600	0.9274	0.9218
	✓	✓	**0.9063**	**0.9701**	**0.9310**
10%	✓	–	0.9149	0.9665	0.9435
	–	✓	0.9006	0.9645	0.9303
	✓	✓	**0.9332**	**0.9807**	**0.9493**

TABLE 8.2: Test results of our semi-supervised models using different PLG inputs. The best results for a given level of supervision are reported in bold.

annotations. Even in that scenario, adding them still increases IoU by 1.8% over RGB frames only. Table 8.2 also shows that using RPD maps alone does not yield the best results. This is not surprising, since RPD values are approximate and do not contain information to distinguish between the ground and object parts lying close to it.

8.6 Conclusion

This chapter investigates a semi-supervised pseudo-labeling strategy to train a neural network to segment the road using images captured by a single road-facing camera with little manual annotations.

In order to minimize the labeling efforts required to train models, we rely on a novel semi-supervised pseudo-labeling pipeline. Like in Chapter 7, we compute approximate v-disparity maps using a depth prediction network and use them to derive dense Road Plane Distance (RPD) maps. These RPD maps contain useful information for the road segmentation task.

By combining RPD maps with features extracted from a depth estimation network and using them to train pseudo-label generators with minimal supervision, we are able to greatly improve IoU over the unsupervised case. The semi-supervised results obtained using 1% (resp. 10%) of ground truth labels improve IoU by 5.3% (resp. 8%) over the unsupervised approach. These results correspond to 95.9% (resp. 98.7%) of a comparable fully-supervised model.

By relying on some ground truth annotation, the semi-supervised pipeline is also able to address the main drawback of the unsupervised RPD-based approach presented in Chapter 7: its lack of Precision due to an inability to distinguish between physically and legally traversable space. By labeling only 1% of ground truth samples, we can observe a Precision increase of +8.7%.

Considering that 1% of the Cityscapes annotations correspond to only 29 frames, these results motivate a pragmatic approach to labeling for segmentation tasks: even minimal labeling efforts can greatly improve results.

Finally, although this work emphasizes an application to road segmentation, our approaches are not specific to urban scenes. Indeed, the depth network used for RPD computation and features extraction in this work can be trained without any label on other datasets, enabling future work to explore applications to monocular robots operating in less-contrained indoor and outdoor environments.

Chapter 9

Conclusion

To conclude this dissertation, we will review our findings and contributions through the lens of the research questions formulated in Section 1.5, before sharing further research directions.

9.1 Summary & Contributions

9.1.1 End-to-End Steering

End-to-End Steering consists in learning to predict steering angle from road-facing camera frames. Imitation learning tackles this task by recording synchronized camera frames and steering angles during demonstrations from a human expert. While existing work tackles the task of imitating human reactions given only camera frames at training time, our work explored another possibility in order to answer the following question.

Research Question 1: Can imitation learning systems for end-to-end steering benefit from privileged information available at no additional labeling cost?

We addressed the first research question in Chapter 3 by designing and evaluating an end-to-end control system that relies on privileged information. We showed that pixel-relevance heatmaps obtained using VisualBackProp can be used in conjunction with approximate lane masks to penalize the model for looking at irrelevant areas during learning. This Distraction Loss term is added to the standard L_2-loss as a regularizer. We obtained approximate masks using a LaneNet model that was trained on a different dataset, which shows that perfect lane masks are not needed for this method. The use of VisualBackProp also has the benefit of not adding any additional learnable parameter to the system. Our Distraction Loss benefits offline metrics across the board, with progress observed for mean squared error, mean absolute error and quantized classification error. This shows that imitation learning systems for end-to-end steering can indeed benefit from privileged information such as pixel-relevance heatmaps at training time.

9.1.2 Unsupervised Free Space Estimation

Pseudo-labels	GT% (frames)	Section	Co-Teaching	Recursive Training	Cutmix	IoU	Precision	Recall
–	100% (2975)	4.5.1	–	–	–	0.9412	0.9726	0.9727
		6.5.1	–	–	✓	0.9454	0.9837	0.9427
Superpixel Clustering	0% (0)	4.5.3	–	–	–	0.8152	0.8854	0.9138
		5.3	✓	–	–	0.8261	0.9093	0.9027
		6.3	–	✓	✓	0.8377	0.9193	0.9129
Superpixel RPD	0% (0)	7.3	–	–	✓	0.8529	0.8827	0.9623
Semi-supervised RPD-based PLG	10% (297)	8.3	–	–	✓	0.9332	0.9807	0.9493
	1% (29)	8.3	–	–	✓	0.9063	0.9701	0.9310

TABLE 9.1: Summary of our free space estimation results. The "Section" column indicates the section number where the method is presented in details.

Free space segmentation has traditionally been approached using supervised segmentation techniques. Although effective, these techniques require vast amounts of pixel-wise annotated frames. Studies have shown that such pixel-level ground truth is significantly more expensive to craft than image-level labels or bounding boxes [Lin+14]. In addition to the large labor costs entailed by labeling each frame [Cor+15], such approaches are held back by the wide variety of environments and lighting conditions that are present at runtime and need to be captured in training data. This need for ever-larger annotated datasets makes supervised learning unsuitable for solving this problem, which is why research questions 2 to 4 were aimed at exploring unsupervised free space estimation. The main results of our free space estimation experiments are summarized in Table 9.1.

Research Question 2: Can the noise present in existing free space pseudo-labels be explicitly taken into account during training in order to improve generalization?

This second theme was explored in Chapter 5, where we used Co-Teaching to restrict training to likely-correct pixels in free space pseudo-labels. Since Co-Teaching was designed to work on classification problems, we proposed a simple adaptation to segmentation that treats every pixel as a separate sample. The intuition for Co-Teaching is based on memorization properties of deep neural networks. Although they will eventually overfit random noise in their training data, they still tend to learn patterns from clean data first. To exploit this, Stochastic Co-Teaching probabilistically filters out the pixels that incur the highest losses. We compared the two Co-Teaching variants against a traditional training loop, and observed improvements in IoU and Precision in both cases. Stochastic Co-Teaching was the best performer in both the single best student and ensembles scenarios. Although explicitly dealing with label noise during training can certainly help improve performances, we also analyzed the limitations of this approach. While the core assumption of Co-Teaching that clean patterns are learnt first is respected in most of the dataset, clean-high-loss and noisy-low-loss samples are still observed and limit the impact Co-Teaching can have.

Research Question 3: Which data augmentation strategies are most effective for pseudo-supervised free space segmentation?

In Chapter 6, we explored different data augmentation scenarios for free space estimation, and we analyzed how recursive training impacts performances. We studied the effect of three distinct data augmentation strategies:

1. *MixUp* trains models on convex combinations of pairs of samples, resulting in new training samples that lie between the original pair.
2. *Color-Flip-Crop* randomly combines the classical augmentation schemes of color jittering, horizontal flips and crops.
3. *CutMix* also merges existing samples but overlays them rather than taking convex combinations, resulting in more locally coherent frames.

We observed that data augmentation is indeed effective at improving pseudo-supervised training in all cases. In particular, CutMix was particularly effective and improved IoU by almost 2% over not using any augmentation. Our qualitative analysis confirmed that a Cutmix-trained model was able to correct some largely wrong pseudo-labels, and also made errors that were semantically more forgivable. The work presented in Chapter 6 was voted among the best submissions of the BNAIC 2021 conference at which it was presented, and was selected for a book chapter publication. Our subsequent works build on these findings by reusing the Cutmix strategy.

Research Question 4: Is is possible to exploit geometrical cues from approximate depth maps to generate more accurate free space pseudo-labels?

While Chapters 5 and 6 used the existing Superpixel Clustering pseudo-labels, Chapter 7 introduced a novel labeling strategy. We proposed the use of a monocular depth network to estimate pixel-wise Road-Plane Distances (RPD) using the v-disparity algorithm with a novel filtering method. We used RPD maps in a pseudo-labeling pipeline to reach state-of-the-art IoU, with reduced complexity and computations compared to prior work. Although the results were interesting from an IoU standpoint, our RPD-based labels were only able to evaluate whether space is physically traversable, but could not deal with traffic rules. For example, ground-level sidewalks or vegetation were often mislabeled as the road in our unsupervised pipeline, resulting in degraded Precision. To alleviate this issue, we proposed to use a restricted subset of ground truth labels in order to learn to differentiate between physically and legally traversable spaces.

9.1.3 Semi-Supervised Free Space Estimation

After exploring different avenues to improve training and pseudo-labels generation for unsupervised free space estimation, Chapter 8 went on to explore a more pragmatic question:

Research Question 5: Can using a fraction of ground truth free space frames result in substantial performance gains?

We investigated a semi-supervised extension to the superpixel generation method described in Chapter 7 and found that even minimal labeling efforts could greatly improve results. Since the pseudo-labels from Chapter 7 were based purely on RPD maps, they failed to distinguish non-traversable areas that aligned with the road plane. The use of ground truth labels helped alleviate this issue: we combined RPD maps with RGB frames and trained a PLG network using the available ground truth. Using as little as 1% (respectively 10%) of ground truth data, we reached an of IoU of 0.9063 (resp. 0.9332). These results correspond to 95.9% (resp. 98.7%) of the IoU obtained by a comparable fully-supervised model, which motivates a pragmatic approach to labeling. Although we tested our work on road segmentation, this method is generic enough to apply to other less constrained-settings, such as indoor and outdoor ground robot navigation.

9.2 Future Research Directions in FSE

We conclude this document with a presentation of possible future research directions that would build on our results. This section is not meant to be exhaustive and only introduces possible research avenues that we find interesting.

Superpixel-level filtering We presented the Superpixel Clustering method in Section 4.3, and used it to generate pseudo-labels in Chapters 5 and 6. In Chapter 7, we proposed an alternative pseudo-labeling technique based on geometric cues we obtained from approximate depth maps. Because both of these schemes rely on segmenting the RGB space into superpixels before classifying them, mistakes tends to occur in cohesive spatial regions rather than single isolated pixels. In Chapter 5, we attempted to discard individual noisy pixels at training time using Co-Teaching, but future work could instead investigate filtering approaches that ignore entire superpixels. One possibility would be to adapt the Stochastic Co-Teaching that we proposed in Chapter 5 to support this.

Iterative training with refinement Since the output of our models can be seen as stronger pseudo-labels, another promising research direction would be to iteratively train models to refine them. This was already partially explored in Chapter 6, but existing work has shown that refinement steps after each training round can help improve results [Kho+17; DHS15]. For example, the method presented in [Kho+17] progressively refines pseudo-labels extracted from bounding box annotations over 10 training rounds with impressive results. Although most of the ideas are not applicable to the problem of free space segmentation since bounding boxes cannot accurately capture free space, their use of a graphical model refinement step could be combined with our work from Chapter 6. One could also explore the possibility of using bounding boxes for entities that are considered occupied space, and to exploit that knowledge to improve free space segmentation.

Exploiting additional sensor modalities One interesting possibility would be to exploit the precise localization data from GNSS traces present in some datasets, and to combine it with publicly available map data. Existing work investigates the use of OpenStreetMap data to extract the road and project it onto training frames by exploiting the known camera calibration in order to obtain road pseudo-labels [Lad+16], and achieve reasonable results for the KITTI dataset [GLU12]. This approach could be tried on the Cityscapes dataset to obtain results that can be compared with more recent literature, and combined with our approaches in order to improve pseudo-labels.

Leverage temporal sequences The ordered frame sequences typically found in driving datasets contain rich temporal information that is not exploited by our approaches. Some recent methods exploit these sequences exclusively at training time, using structure-from-motion to learn to predict monocular depth maps [Zho+17; God+19] or to segment obstacle footprints [Wat+20]. However, These methods still use single frames at inference time. Another line of work uses similar techniques, but focuses on predicting temporally consistent depth maps in entire videos [Luo+20; Hoy+21]. These concepts could be implemented to predict temporally consistent free space masks.

Appendix A

Image Sources

A.1 Images from Introduction

Figure 1.2 is made of the following publicly available images, from left-to-right and top-to-bottom:

- *Cars Road Trip - Lightning McQueen* by *kurros* is licensed under CC BY-NC-SA 2.0, and available at https://www.flickr.com/photos/49961268@N00/150160407.
- *Car* by *PHOTOPHANATIC1* is licensed under CC BY-NC-SA 2.0, and available at https://www.flickr.com/photos/65089906@N00/35185475274.
- *2000 Fiat Multipla* by *harry_nl* is licensed under CC BY-NC-SA 2.0, and available at https://www.flickr.com/photos/23363966@N02/13452836975.
- *Edinbugh car fog* by *shannonkringen* is licensed under CC BY-SA 2.0, and available at https://www.flickr.com/photos/18161271@N00/7039623059
- *Lower Park Rd car + fog* by *copperhead103* is licensed under CC BY-NC-ND 2.0, and available at https://www.flickr.com/photos/26945939@N05/3153701817
- *Car Snow* by *M_Bitting* is licensed under CC BY-NC-SA 2.0, and available at https://www.flickr.com/photos/141003241@N08/40140747525
- Image extracted from *Assume self-driving cars are a hacker's dream? Think again* by *Alex Hern* published by The Guardian, and available at https://www.theguardian.com/technology/2017/aug/30/self-driving-cars-hackers-security.

References

[Aba+15] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[ACE] ACEA. *Average age of the EU vehicle fleet by country*. https://www.acea.auto/figure/average-age-of-eu-vehicle-fleet-by-country/. Accessed: 2022-05-11.

[Adm13] National Highway Traffic Safety Administration. *National Motor Vehicle Crash Causation Survey: Report to Congress*. 2013. ISBN: 9781492772606.

[AFS17] Takuya Akiba, Keisuke Fukuda, and Shuji Suzuki. "ChainerMN: Scalable Distributed Deep Learning Framework". In: *Proceedings of Workshop on ML Systems in The Thirty-first Annual Conference on Neural Information Processing Systems (NIPS)*. 2017.

[BC18] Devansh Bisla and Anna Choromanska. "VisualBackProp for learning using privileged information with CNNs". In: *CoRR* abs/1805.09474 (2018).

[Bea+16] Amy Bearman et al. "What's the Point: Semantic Segmentation with Point Supervision". English. In: *Computer Vision – ECCV 2016*. Lecture Notes in Computer Science (LNCS). 14th European Conference on Computer Vision 2016, ECCV 2016 ; Conference date: 08-10-2016 Through 16-10-2016. Springer International Publishing, Sept. 2016, pp. 549–565. ISBN: 978-3-319-46477-0. DOI: 10.1007/978-3-319-46478-7_34.

[Ben+09] Yoshua Bengio et al. "Curriculum Learning". In: *Proceedings of the 26th Annual International Conference on Machine Learning*. ICML '09. Montreal, Quebec, Canada: Association for Computing Machinery, 2009, 41–48. ISBN: 9781605585161. DOI: 10.1145/1553374.1553380.

[BFP09] Hernán Badino, Uwe Franke, and David Pfeiffer. "The Stixel World - A Compact Medium Level Representation of the 3D-World". In: *Pattern Recognition*. Ed. by Joachim Denzler, Gunther Notni, and Herbert Süße. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 51–60. ISBN: 978-3-642-03798-6.

[Bin+16] Alexander Binder et al. "Layer-Wise Relevance Propagation for Neural Networks with Local Renormalization Layers". In: *Artificial Neural Networks and Machine Learning - ICANN 2016 - 25th International Conference on Artificial Neural Networks, Barcelona, Spain, September 6-9, 2016, Proceedings, Part II*. 2016, pp. 63–71. DOI: 10.1007/978-3-319-44781-0_8.

[BIS07] Martin Buehler, Karl Iagnemma, and Sanjiv Singh. *The 2005 DARPA Grand Challenge: The Great Robot Race*. 1st. Springer Publishing Company, Incorporated, 2007. ISBN: 3540734287.

[BIS09] Martin Buehler, Karl Iagnemma, and Sanjiv Singh. *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*. 1st. Springer Publishing Company, Incorporated, 2009. ISBN: 3642039901.

[BKC17] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. "Segnet: A deep convolutional encoder-decoder architecture for image segmentation". In: *IEEE transactions on pattern analysis and machine intelligence* 39.12 (2017), pp. 2481–2495.

[Boj+16] Mariusz Bojarski et al. "End to End Learning for Self-Driving Cars". In: *CoRR* abs/1604.07316 (2016).

[Boj+17] Mariusz Bojarski et al. "Explaining How a Deep Neural Network Trained with End-to-End Learning Steers a Car". In: *CoRR* abs/1704.07911 (2017).

[Boj+18] Mariusz Bojarski et al. "VisualBackProp: Efficient Visualization of CNNs for Autonomous Driving". In: *2018 IEEE International Conference on Robotics and Automation, ICRA 2018, Brisbane, Australia, May 21-25, 2018*. 2018, pp. 1–8. DOI: 10.1109/ICRA.2018.8461053.

[BTVG06] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. "SURF: Speeded Up Robust Features". In: *Computer Vision – ECCV 2006*. Ed. by Aleš Leonardis, Horst Bischof, and Axel Pinz. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 404–417. ISBN: 978-3-540-33833-8.

[Cha+20] Yu-Ting Chang et al. "Mixup-CAM: Weakly-supervised Semantic Segmentation via Uncertainty Regularization". In: *31st British Machine Vision Conference 2020, BMVC 2020, Virtual Event, UK, September 7-10, 2020*. BMVA Press, 2020.

[Che+19a] Pengfei Chen et al. "Understanding and Utilizing Deep Neural Networks Trained with Noisy Labels". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, June 2019, pp. 1062–1070.

[Che+19b] Yuhua Chen et al. "Learning Semantic Segmentation From Synthetic Data: A Geometrically Guided Input-Output Adaptation Approach". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 1841–1850. DOI: 10.1109/CVPR.2019.00194.

[Che+20a] Liang-Chieh Chen et al. "Naive-student: Leveraging semi-supervised learning in video sequences for urban scene segmentation". In: *European Conference on Computer Vision*. 2020, pp. 695–714.

[Che+20b] Ting Chen et al. "A Simple Framework for Contrastive Learning of Visual Representations". In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, July 2020, pp. 1597–1607.

[Chi+19] F Chiaroni et al. "Hallucinating a Cleanly Labeled Augmented Dataset from a Noisy Labeled Dataset Using GANs". In: *26th IEEE International Conference on Image Processing (ICIP)*. Ed. by IEEE. Taipei, Taiwan, Sept. 2019.

[CN20] Simon Chadwick and Paul Newman. "Radar as a Teacher: Weakly Supervised Vehicle Detection using Radar Labels". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 2020, pp. 222–228. DOI: 10.1109/ICRA40945.2020.9196855.

[Cod+18] Felipe Codevilla et al. "On Offline Evaluation of Vision-Based Driving Models". In: *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XV*. 2018, pp. 246–262. DOI: 10.1007/978-3-030-01267-0_15.

[Cor+15] Marius Cordts et al. "The cityscapes dataset". In: *CVPR Workshop on the Future of Datasets in Vision*. 2015.

[Cor+17] Marius Cordts et al. "The Stixel World: A Medium-Level Representation of Traffic Scenes". In: *Image and Vision Computing* 68 (Feb. 2017). DOI: 10.1016/j.imavis.2017.01.009.

[Csu17] Gabriela Csurka. "A Comprehensive Survey on Domain Adaptation for Visual Applications". In: *Domain Adaptation in Computer Vision Applications*. Ed. by Gabriela Csurka. Cham: Springer International Publishing, 2017, pp. 1–35. ISBN: 978-3-319-58347-1. DOI: 10.1007/978-3-319-58347-1_1.

[Dav+07] Andrew J. Davison et al. "MonoSLAM: Real-Time Single Camera SLAM". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.6 (2007), pp. 1052–1067. DOI: 10.1109/TPAMI.2007.1049.

[Den+09] J. Deng et al. "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[DHS15] Jifeng Dai, Kaiming He, and Jian Sun. "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.

[Dos+17] Alexey Dosovitskiy et al. "CARLA: An Open Urban Driving Simulator". In: *Proceedings of the 1st Annual Conference on Robot Learning*. 2017, pp. 1–16.

[DT05] N. Dalal and B. Triggs. "Histograms of oriented gradients for human detection". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. 2005, 886–893 vol. 1. DOI: 10.1109/CVPR.2005.177.

[Duc79] Claude Duchon. "Lanczos Filtering in One and Two Dimensions". In: *Journal of Applied Meteorology* 18 (Aug. 1979), pp. 1016–1022.

[Dur+17] T. Durand et al. "WILDCAT: Weakly Supervised Learning of Deep ConvNets for Image Classification, Pointwise Localization and Segmentation". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 5957–5966. DOI: 10.1109/CVPR.2017.631.

[DV16] Vincent Dumoulin and Francesco Visin. "A guide to convolution arithmetic for deep learning". In: *ArXiv e-prints* (Mar. 2016).

[Fac] National Safety Council Injury Facts. *Advanced Driver Assistance Systems*. https://www.asirt.org/safe-travel/road-safety-facts/. Accessed: 2022-05-11.

[FB81] Martin A Fischler and Robert C Bolles. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography". In: *Communications of the ACM* 24.6 (1981), pp. 381–395.

[FH04] Pedro Felzenszwalb and Daniel Huttenlocher. "Efficient Graph-Based Image Segmentation". In: *International Journal of Computer Vision* 59.2 (2004), pp. 167–181.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[GLU12] Andreas Geiger, Philip Lenz, and Raquel Urtasun. "Are we ready for autonomous driving? the kitti vision benchmark suite". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. 2012, pp. 3354–3361.

[God+19] Clement Godard et al. "Digging Into Self-Supervised Monocular Depth Estimation". In: *Proceedings of the International Conference on Computer Vision (ICCV)*. 2019.

[Guo+18] Sheng Guo et al. "CurriculumNet: Weakly Supervised Learning from Large-Scale Web Images". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Sept. 2018.

[Han+18] Bo Han et al. "Co-teaching: Robust training of deep neural networks with extremely noisy labels". In: *Advances in neural information processing systems*. 2018, pp. 8527–8537.

[HAS16] Ali Harakeh, Daniel Asmar, and Elie Shammas. "Identifying Good Training Data for Self-Supervised Free Space Estimation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016.

[He+16] K. He et al. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[He+16] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[He+17] Kaiming He et al. "Mask R-CNN". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 2980–2988. DOI: 10.1109/ICCV.2017.322.

[Hir08] H. Hirschmuller. "Stereo Processing by Semiglobal Matching and Mutual Information". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.2 (2008), pp. 328–341. DOI: 10.1109/TPAMI.2007.1166.

[Hof+16] Judy Hoffman et al. "FCNs in the Wild: Pixel-level Adversarial and Constraint-based Adaptation". In: *CoRR* abs/1612.02649 (2016).

[Hoy+21] Lukas Hoyer et al. "Three Ways to Improve Semantic Segmentation with Self-Supervised Depth Estimation". In: June 2021, pp. 11125–11135. DOI: 10.1109/CVPR46437.2021.01098.

[HVD15] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. "Distilling the Knowledge in a Neural Network". In: *NIPS Deep Learning and Representation Learning Workshop*. 2015.

[Ise+18] Fabian Isensee et al. "nnU-Net: Self-adapting Framework for U-Net-Based Medical Image Segmentation". In: *CoRR* abs/1809.10486 (2018).

[Jan+20] J. Janai et al. "Computer Vision for Autonomous Vehicles: Problems, Datasets and State-of-the-Art". In: *ArXiv* abs/1704.05519 (2020).

[Jég+17] S. Jégou et al. "The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2017), pp. 1175–1183.

[Jia+18] Lu Jiang et al. "MentorNet: Learning Data-Driven Curriculum for Very Deep Neural Networks on Corrupted Labels". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, July 2018, pp. 2304–2313.

[Kar+20] Davood Karimi et al. "Deep learning with noisy labels: Exploring techniques and remedies in medical image analysis". In: *Medical Image Analysis* 65 (2020), p. 101759. ISSN: 1361-8415. DOI: https://doi.org/10.1016/j.media.2020.101759.

[Kar20a] Andrej Karpathy. *AI for Full-Self Driving at Tesla, 5th Annual Scaled Machine Learning Conference*. https://www.youtube.com/watch?v=hx7BXih7zx8. 2020.

[Kar20b] Andrej Karpathy. *CVPR 2020: Scalability in Autonomous Driving Workshop*. https://www.youtube.com/watch?v=X2CpuabzRaY. 2020.

[KB15] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *CoRR* abs/1412.6980 (2015).

[Ker+20] Hoel Kervadec et al. "Bounding boxes for weakly supervised segmentation: Global constraints get close to full supervision". In: *Medical Imaging with Deep Learning*. 2020.

[Kho+17] A. Khoreva et al. "Simple Does It: Weakly Supervised Instance and Semantic Segmentation". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 1665–1674. DOI: 10.1109/CVPR.2017.181.

[Lad+16] A. Laddha et al. "Map-supervised road detection". In: *2016 IEEE Intelligent Vehicles Symposium (IV)*. 2016, pp. 118–123. DOI: 10.1109/IVS.2016.7535374.

[LAT02] Raphael Labayrade, Didier Aubert, and J-P Tarel. "Real time obstacle detection in stereovision on non flat road geometry through" v-disparity" representation". In: *Intelligent Vehicle Symposium, 2002. IEEE*. Vol. 2. IEEE. 2002, pp. 646–651.

[LeC+05] Yann LeCun et al. "Off-road Obstacle Avoidance Through End-to-end Learning". In: *Proceedings of the 18th International Conference on Neural Information Processing Systems*. NIPS'05. Vancouver, British Columbia, Canada: MIT Press, 2005, pp. 739–746.

[Lec+98] Y. Lecun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791.

[Li+17] Yuncheng Li et al. "Learning from Noisy Labels with Distillation". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 1928–1936. DOI: 10.1109/ICCV.2017.211.

[Lin+14] Tsung-Yi Lin et al. "Microsoft coco: Common objects in context". In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[Lin+16] D. Lin et al. "ScribbleSup: Scribble-Supervised Convolutional Networks for Semantic Segmentation". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 3159–3167. DOI: 10.1109/CVPR.2016.344.

[Low99] Dawid G. Lowe. "Object recognition from local scale-invariant features". In: *International Conference on Computer Vision, 20–25 September, 1999, Kerkyra, Corfu, Greece, Proceedings*. Vol. 2. 1999, pp. 1150–1157. DOI: 10.1109/ICCV.1999.790410.

[LSD15] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440.

[LSO20] Mingchen Li, Mahdi Soltanolkotabi, and Samet Oymak. "Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 4313–4324.

[Lu+17] Zhiwu Lu et al. "Learning from Weak and Noisy Labels for Semantic Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39 (Mar. 2017), pp. 486–500. DOI: 10.1109/TPAMI.2016.2552172.

[Luo+20] Xuan Luo et al. "Consistent Video Depth Estimation". In: 39.4 (2020).

[MES08] J. Mairal, M. Elad, and G. Sapiro. "Sparse Representation for Color Image Restoration". In: *Trans. Img. Proc.* 17.1 (Jan. 2008), 53–69. ISSN: 1057-7149. DOI: 10.1109/TIP.2007.911828.

[Miy+17] Takeru Miyato et al. "Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PP (Apr. 2017). DOI: 10.1109/TPAMI.2018.2858821.

[MMB16] Nassim Motamedidehkordi, Martin Margreiter, and Thomas Benz. "Shockwave Suppression by Vehicle-to-Vehicle Communication". In: *Transportation Research Procedia* 15 (Dec. 2016), pp. 471–482. DOI: 10.1016/j.trpro.2016.06.040.

[MNA16] F. Milletari, N. Navab, and S. Ahmadi. "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation". In: *2016 Fourth International Conference on 3D Vision (3DV)*. 2016, pp. 565–571. DOI: 10.1109/3DV.2016.79.

[MSS17] Eran Malach and Shai Shalev-Shwartz. "Decoupling "when to update" from "how to update"". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017.

[MUT18] Jakob Mayr, Christian Unger, and Federico Tombari. "Self-supervised learning of the drivable area for autonomous vehicles". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 362–369.

[Nel21] Mills Nelson. *Will Self-Driving Cars Disrupt The Insurance Industry?* https://www.forbes.com/sites/columbiabusinessschool/2021/03/25/will-self-driving-cars-disrupt-the-insurance-industry/. Mar. 2021.

[Nev+18] Davy Neven et al. "Towards End-to-End Lane Detection: an Instance Segmentation Approach". In: *2018 IEEE Intelligent Vehicles Symposium, IV 2018, Changshu, Suzhou, China, June 26-30, 2018*. 2018, pp. 286–291. DOI: 10.1109/IVS.2018.8500547.

[NS13] Tan Nguyen and Scott Sanner. "Algorithms for Direct 0-1 Loss Optimization in Binary Classification". In: *ICML*. 2013.

[OBB16] G. L. Oliveira, W. Burgard, and T. Brox. "Efficient deep models for monocular road segmentation". In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2016, pp. 4885–4891. DOI: 10.1109/IROS.2016.7759717.

[Okt+18] Ozan Oktay et al. "Attention U-Net: Learning Where to Look for the Pancreas". In: (Apr. 2018).

[Pap66] Seymour A. Papert. *The Summer Vision Project*. https://dspace.mit.edu/handle/1721.1/6125?show=full. 1966.

[Pas+19] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035.

[PC15] P. O. Pinheiro and R. Collobert. "From image-level to pixel-level labeling with Convolutional Networks". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015. DOI: 10.1109/CVPR.2015.7298780.

[Pom89] Dean A. Pomerleau. "Advances in Neural Information Processing Systems 1". In: ed. by David S. Touretzky. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1989. Chap. ALVINN: An Autonomous Land Vehicle in a Neural Network, pp. 305–313. ISBN: 1-558-60015-9.

[RF21] François Robinet and Raphaël Frank. "Refining Weakly-Supervised Free Space Estimation Through Data Augmentation and Recursive Training". In: *Proceedings of BNAIC/BeneLearn 2021*. 2021.

[RF22] François Robinet and Raphaël Frank. "Refining Weakly-Supervised Free Space Estimation Through Data Augmentation and Recursive Training". In: *Artificial Intelligence and Machine Learning*. Springer International Publishing, 2022, pp. 30–45. ISBN: 978-3-030-93842-0. DOI: 10.1007/978-3-030-93842-0_2.

[RFB15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. 2015, pp. 234–241.

[Rob] *RoboBus Dataset*. https://github.com/raphaelfrank/robobus. Accessed: 2020-12-01.

[Rob+20] François Robinet et al. "Leveraging Privileged Information to Limit Distraction in End-to-End Lane Following". In: *2020 IEEE 17th Annual Consumer Communications Networking Conference (CCNC)*. 2020, pp. 1–6. DOI: 10.1109/CCNC46108.2020.9045110.

[Rob+22a] François Robinet et al. "Weakly-Supervised Free Space Estimation Through Stochastic Co-Teaching". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*. Jan. 2022, pp. 618–627.

[Rob+22b] François Robinet et al. "Striving for Less: Minimally-Supervised Pseudo-Label Generation for Monocular Road Segmentation". In: *IEEE Robotics and Automation Letters* (2022), pp. 1–7. DOI: 10.1109/LRA.2022.3193463.

[Sch14] Jürgen Schmidhuber. *Who Invented Backpropagation?* https://people.idsia.ch/~juergen/who-invented-backpropagation-2014.html. Accessed: 2022-05-11. 2014.

[Sei+21] Daniel Seichter et al. "Efficient rgb-d semantic segmentation for indoor scene analysis". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 13525–13531.

[SIRT] Association for Safe International Road Travel. *Road Safety Facts*. https://www.asirt.org/safe-travel/road-safety-facts/. Accessed: 2022-05-11.

[SNS11] Roland Siegwart, Illah R. Nourbakhsh, and Davide Scaramuzza. *Introduction to Autonomous Mobile Robots*. 2nd. The MIT Press, 2011. ISBN: 0262015358.

[Soh+20] Kihyuk Sohn et al. "FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence". 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 Hugo Larochelle et al. 2020.

[Suk+15] Sainbayar Sukhbaatar et al. "Training convolutional networks with noisy labels". English (US). In: 3rd International Conference on Learning Representations, ICLR 2015 ; Conference date: 07-05-2015 Through 09-05-2015. Jan. 2015.

[Tei+16] Marvin Teichmann et al. "MultiNet: Real-time Joint Semantic Reasoning for Autonomous Driving". In: *CoRR* abs/1612.07695 (2016).

[Tor] *Torchvision: Datasets, Transforms and Models specific to Computer Vision*. https://github.com/pytorch/vision. 2021.

[Tre+18] Jonathan Tremblay et al. "Training deep networks with synthetic data: Bridging the reality gap by domain randomization". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2018, pp. 969–977.

[TSK17] S. Tsutsui, S. Saito, and T. Kerola. "Distantly Supervised Road Segmentation". In: *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*. 2017, pp. 174–181. DOI: 10.1109/ICCVW.2017.29.

[Tsu+18] Satoshi Tsutsui et al. "Minimizing Supervision for Free-space Segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2018, pp. 988–997.

[Tus] *Tusimple benchmark*. http://benchmark.tusimple.ai. 2017.

[TV17] Antti Tarvainen and Harri Valpola. "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017.

[Var+19] Georgios Varisteas et al. "Evaluation of End-To-End Learning for Autonomous Driving: The Good, the Bad and the Ugly". In: *2nd International Conference on Intelligent Autonomous Systems, Singapore, Feb. 28 to Mar. 2, 2019*. 2019.

[VFR21] Georgios Varisteas, Raphaël Frank, and François Robinet. "RoboBus: A Diverse and Cross-Border Public Transport Dataset". In: *2021 IEEE International Conference on Pervasive Computing and Communications Workshops*. 2021, pp. 269–274. DOI: 10.1109/PerComWorkshops51409.2021.9431129.

[VI15] Vladimir Vapnik and Rauf Izmailov. "Learning Using Privileged Information: Similarity Control and Knowledge Transfer". In: *J. Mach. Learn. Res.* 16.1 (Jan. 2015), pp. 2023–2049. ISSN: 1532-4435.

[Wat+20] Jamie Watson et al. "Footprints and Free Space from a Single Color Image". In: *Computer Vision and Pattern Recognition (CVPR)*. 2020.

[WSL19] Hengli Wang, Yuxiang Sun, and Ming Liu. "Self-supervised drivable area and road anomaly segmentation using rgb-d data for robotic wheelchairs". In: *IEEE Robotics and Automation Letters* 4.4 (2019), pp. 4386–4393.

[Xia+15] Liang Xiao et al. "CRF based road detection with multi-sensor fusion". In: *2015 IEEE Intelligent Vehicles Symposium (IV)*. 2015, pp. 192–198. DOI: 10.1109/IVS.2015.7225685.

[Xie+19] Qizhe Xie et al. "Unsupervised Data Augmentation for Consistency Training". In: *arXiv preprint arXiv:1904.12848* (2019).

[Xie+20] Wenbin Xie et al. "Learning Effectively from Noisy Supervision for Weakly Supervised Semantic Segmentation". In: *BMVC*. 2020.

[Yak19] Pavel Yakubovskiy. *Segmentation Models*. https://github.com/qubvel/segmentation_models. 2019.

[Yao+15] Jian Yao et al. "Estimating Drivable Collision-Free Space from Monocular Video". In: *2015 IEEE Winter Conference on Applications of Computer Vision*. 2015, pp. 420–427. DOI: 10.1109/WACV.2015.62.

[Yao+20] Quanming Yao et al. "Searching to Exploit Memorization Effect in Learning with Noisy Labels". In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, July 2020, pp. 10789–10798.

[Yun+19] Sangdoo Yun et al. "CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019), pp. 6022–6031.

[Zbo+21] Jure Zbontar et al. "Barlow Twins: Self-Supervised Learning via Redundancy Reduction". In: *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 12310–12320.

[Zha+16] Chiyuan Zhang et al. "Understanding deep learning requires rethinking generalization". In: *Communications of the ACM* 64 (Nov. 2016). DOI: 10.1145/3446776.

[Zha+17] Hengshuang Zhao et al. "Pyramid Scene Parsing Network". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 6230–6239. DOI: 10.1109/CVPR.2017.660.

[Zha+18] Hongyi Zhang et al. "mixup: Beyond Empirical Risk Minimization". In: *International Conference on Learning Representations* (2018).

[Zho+16] Bolei Zhou et al. "Learning Deep Features for Discriminative Localization". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 2921–2929. DOI: 10.1109/CVPR.2016.319.

[Zho+17] Tinghui Zhou et al. "Unsupervised Learning of Depth and Ego-Motion from Video". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017. DOI: 10.1109/CVPR.2017.700.