Where can I go? Deep multi-modal scene understanding for outdoor navigation

feat. colors, cats and imaginary robots

Master Thesis

Galadrielle Humblot-Renaux

Aalborg University
Electronics and IT

This document is typeset in LaTeX based on Jesper Kjær Nielsen's AAU report template.
All diagrams were created with Inkscape.

Electronics and IT
Aalborg University
http://www.aau.dk

AALBORG UNIVERSITY
STUDENT REPORT

Title:
Where can I go?
Deep multi-modal scene understanding for outdoor navigation

Theme:
Computer Vision for Robotics

Project Period:
Spring Semester 2021

Project Group:
1061a

Participant(s):
Galadrielle Humblot-Renaux

Supervisor(s):
Rikke Gade
Letizia Marchegiani

Copies: 1

Page Numbers: 141

Date of Completion:
June 3, 2021

Abstract:

This project delves into deep learning-based computer vision for scene understanding in the context of autonomous outdoor navigation. Rather than relying on specific scene-dependent semantic categories, we take an affordance-based approach, proposing to parse egocentric images in terms of how a vehicle or robot can drive in them. We use a SegNet-based image segmentation network as our building block for classifying pixels into 3 driveability levels, and explore soft labelling, pixel-wise loss weighting, and deep adaptive fusion schemes to penalize severe mistakes during learning, improve segmentation in regions of interest, and incorporate infrared and depth data into the prediction. The proposed training schemes and multi-modal architecture are evaluated on 9 public datasets, showing promising results across unstructured forested environments, urban driving scenes, and multi-view hand-held captures.

Summary

Where we started An autonomous vehicle or robot left to roam in the wild needs to know where it can go, and what to avoid. This is especially challenging when navigating outdoors and exclusively relying on images captured by on-board sensors to interpret the robot's surroundings. We specifically tackle two challenging aspects of vision-based scene understanding for outdoor navigation:

First, the robot may cover vast areas, and encounter unfamiliar territory with diverse terrain or previously unseen obstacles. With this in mind, for the purpose of navigation, we argue that learning to detect specific scene elements based on what the robot is expected to encounter - such as cars, pedestrians, or roads - is overly restrictive and does not scale well outside of controlled, predictable environments. Thus, we propose to parse scenes in functional rather than descriptive terms, directly learning a general notion of driveability which can be broadly applied to any type of scene or application.

Secondly, the robot may operate at different times of day or year, thus facing unfavourable lighting or weather conditions which degrade its view. In this case, relying on a single type of sensor is not sufficient: visible spectrum cameras are easily blinded and depth or infrared images, while more reliable, may lack useful appearance-based cues for scene understanding. Thus, we explore fusion approaches for leveraging complementary features from these different modalities.

Where we wandered This work takes a fully supervised image segmentation approach, leveraging existing multi-modal datasets for training and evaluation. We train a SegNet-based architecture to classify pixels into three driveability levels, distinguishing between areas which are *preferable*, *possible*, and *impossible* to drive on, and employ a soft labelling scheme which incorporates inter-class distances during training. Additionally, rather than giving each pixel equal contribution during learning, we present a loss weighting scheme which assigns less importance to pixels located along object boundaries or in the distance, as they bear less relevance for driving decisions.

After training SegNet to predict driveability in visible spectrum images, we explore the potential of infrared and depth modalities as complementary predictors by integrating the network into a deep multi-branch fusion architecture, each branch specializing in a particular modality. Modality-specific features are fused across branches either in the middle of the network where spatial resolution is the coarsest (*mid* fusion), after the decoding stage where a full-resolution prediction is

recovered (*late*), or both (*dual*). Fusion is performed by a small convolutional sub-network which weighs each input modality at the feature-level before producing a combined feature map: the network learns to selectively emphasize the most informative features from each modality. We compare deep fusion architecture variants to an early fusion baseline which simply concatenates modalities at the input.

These methods are first developed and evaluated in isolation on two public datasets: Freiburg Forest, which features RGB, pseudo depth, and near-infared (NIR) images of unstructured forested environments, and Cityscapes, an urban driving dataset with RGB and real stereo depth data. We then jointly validate our soft labelling and loss weighting scheme by learning pixel-wise driveability across a diverse combination of visible spectrum images from 8 datasets. Lastly, we show that these learning strategies can be successfully applied to the proposed multi-modal architecture: we use Freiburg Thermal, a large-scale RGB-thermal dataset, for training, and assess generalization to out-of-dataset RGB-thermal captures.

Where we landed Our experimental results can be summarized as follows:

- we find that pre-training SegNet on descriptive classes, and adapting it to learn driveability via transfer learning achieves higher recall for out-of-domain obstacles than learning driveability from scratch, or mapping object labels to driveability as a post-prediction step. We show that a consistent representation of driveability can be learned across a wide combination of datasets, but that conflicting label definitions and semantic ambiguity causes frequent confusion between areas which are *preferable* vs. just *possible* to drive on.
- when training SegNet under a soft ordinal labelling scheme, the network learns to make less severe mistakes than when using a standard one-hot labelling approach. However, results hinge on the ranking definition used to generate the labels, revealing a trade-off between mistake severity and accuracy. With safety in mind, we opt for an asymmetrical inter-class distance metric based on squared log difference, which assigns the highest penalty when mis-classifying obstacles as *preferable* to drive on.
- compared to a standard uniformly-weighted loss, our loss weighting scheme shows mixed quantitative results depending on the dataset, but has a noticeable qualitative impact, producing a more a smooth and cohesive segmentation, with loss of detail considered an acceptable trade-off for navigation.
- in Freiburg Forest's unstructured forested environments, the addition of NIR and pseudo depth improves segmentation, both in an early and deep fusion configuration, with the most substantial gains brought by NIR. In Cityscapes' urban scenes, the addition of real stereo depth only improves segmentation in a deep fusion configuration, and we find depth completion to be an unnecessary pre-processing step in this case. For deep fusion, we find it beneficial to fuse modality-specific features both at a middle and late stage in the network; the RGB-thermal fusion results in our final experiment confirms the viability of the approach, as well as the valuable properties of thermal imaging for scene understanding.

Contents

Preface **viii**

1 Introduction **1**
1.1 Research directions 1
1.2 Project scope 3
1.3 Outline 3

2 Background **4**
2.1 Beyond monocular RGB imaging 4
2.1.1 Active vs. passive sensing 4
2.1.2 Infrared imaging 4
2.1.3 Depth vision 5
2.2 Deep learning for pixel-wise classification 8
2.2.1 Semantic segmentation as pixel-wise classification 8
2.2.2 Architectures 11
2.2.3 Training neural networks 12
2.2.4 Evaluation metrics 14

3 Related works **15**
3.1 Functional scene understanding 15
3.1.1 Mediated perception: describing what you see and nothing more 16
3.1.2 Scene geometry: just because it's free doesn't mean it's a good idea 17
3.1.3 Behaviour reflex: direct mapping from image to action 17
3.1.4 Affordances for navigation 17
3.2 Multi-modal data fusion for scene understanding 18

4 General approach **20**
4.1 What are we trying to predict? 20
4.2 Battle plan 21
4.3 Implementation details 25

5 Data is everything **26**
5.1 Relevant datasets . . . 26
5.1.1 RGB-D-IR . . . 26
5.1.2 RGB-IR . . . 28
5.1.3 RGB-D . . . 29
5.1.4 Honorable mentions . . . 30
5.2 Chosen datasets . . . 30
5.2.1 Dataset splits . . . 31
5.3 Data preparation . . . 32
5.3.1 Depth maps . . . 32
5.3.2 Image resolution . . . 35
5.3.3 Augmentation . . . 35

6 Navigation-oriented scene understanding **37**
6.1 Segmentation architecture . . . 37
6.2 Object classes to driveability . . . 38
6.3 Soft labels for ordinal segmentation . . . 47
6.3.1 Generating soft labels . . . 47
6.3.2 Ranking definition . . . 48
6.3.3 Training procedure . . . 49
6.3.4 Evaluation . . . 50
6.4 Loss weighting . . . 54
6.4.1 Generating weight maps . . . 55
6.4.2 Weighted loss computation . . . 56
6.4.3 Evaluation . . . 57

7 Segmenting depth and infrared images **60**
7.1 Data augmentation . . . 60
7.2 Depth completion . . . 63

8 Multi-modal fusion **65**
8.1 Channel stacking for early fusion . . . 65
8.2 Cooler fusion . . . 68
8.2.1 Multi-modal segmentation architecture with indexed unpooling . . . 68
8.2.2 Fusion units . . . 70
8.2.3 Initialization and training procedure . . . 73
8.2.4 Evaluation . . . 74
8.3 The slow elephant in the room . . . 84

9 Bringing it all together **85**
9.1 Selected models . . . 87
9.2 No dataset left behind . . . 88
9.2.1 Data . . . 88
9.2.2 Training procedure . . . 90
9.2.3 Evaluation . . . 90
9.2.4 Bonus content: driveability in the wild . . . 99
9.3 No modality left behind . . . 100
9.3.1 Training procedure . . . 101
9.3.2 Evaluation . . . 102

10 Discussion **108**
10.1 Research questions . . . 108
10.2 Challenges and limitations . . . 111
10.3 Going further down the rabbit hole . . . 112
10.4 Towards navigation: next steps . . . 114
10.5 Future directions: dream big or go home . . . 115

11 Conclusion **117**

Bibliography **119**

A Miscellaneous mess **131**
A.1 Mapping from object classes to driveability . . . 131
A.2 Dataset overview . . . 132
A.3 Depth completion . . . 135
A.4 Effect of batch normalization when training SSMA fusion units . . . 137
A.5 Benchmarking - implementation details . . . 138
A.6 Examples of loss weight maps . . . 138
A.7 Demo videos . . . 140
A.7.1 Visible spectrum models . . . 140
A.7.2 Visible+Thermal fusion models . . . 141

Preface

This thesis was written to conclude my study in the Robotics M.Sc. programme at Aalborg University. The experiments were made possible thanks to the AI Cloud computational resources provided by CLAAUDIA's Research Data Services.

merci I would like to genuinely thank my supervisors Rikke Gade and Letizia Marchegiani. Thank you to Rikke for leaving me so much freedom in defining and shaping this project, sharing my enthusiasm from day 1, and indulging my curiosities while keeping me on track with valuable perspectives. Thank you to Letizia for sparking and encouraging my interest in research over the course of this degree, gracefully putting up with all my clueless emails along the way, and hopping aboard this final semester with just as much enthusiasm, bringing a fresh eye to the project. This thesis along with our discussions have only left me more eager to keep learning (and laughing).

Beyond academic walls, special thanks to my bubble for the good company in these deeply weird times - Carolina for the beers and graphic design gems and whacky movies, Eduardo for losing the game so many times, Kia for always knowing how to fish me out of my own head, Pernille for nerding out with me to good music, Pia for the skeleton facts and workaholic walks, and Veve for all the colors and glasses of water and Brazilian coffee.

Thanks to cats for being cats and to robots for staying out of the way :)

Aalborg University, June 3, 2021

Galadrielle Humblot-Renaux
<ghumbl19@student.aau.dk>

Chapter 1

Introduction

Awareness of one's immediate surroundings is crucial for navigation. As humans, we naturally understand what surrounds us and how to interact with different parts of a scene, even when faced with new, unpredictable environments. Achieving human-level scene understanding for autonomous systems is an active field of research, and requires developing robust computer vision methods which can interpret raw sensory data into useful representations for real-world operation. In this work, we tackle egocentric perception for outdoor navigation "in the wild", where a robot is likely to encounter a wide range of different scene geometries, terrains and obstacles, yet should still be able to identify safe and suitable routes, using on-board sensors capturing the scene from its own point of view. We investigate how images of outdoor scenes can be parsed both in a *useful* way, specifically considering the task of autonomous navigation, and a *reliable* way, by incorporating multiple sensing modalities into the prediction.

1.1 Research directions

Outdoor scene understanding for navigation

In the automotive field, scene understanding primarily involves parsing urban landscapes for driving on structured roads and negotiating traffic. In the broader context of mobile robotics, a vehicle may operate in much more diverse, off-road or pedestrian environments - in which case, the question of *where can I drive?* or *what should I avoid?* cannot necessarily be reduced to recognizing specific scene elements such as lane boundaries, cars or people, as the vehicle may encounter unfamiliar terrain, unexpected obstacles, or open areas with no clear path.

With these challenges in mind, we try to take a general, functional approach to scene understanding, with the idea that it may not be necessary to precisely distinguish and categorize every element in the scene in order to determine where a

vehicle can safely drive. In diverse environments, rather than identifying elements by name (e.g. *this is a tree*), it may be more useful to parse the scene in terms of how the vehicle can navigate in it (*this area is more driveable than that one*). Such a representation could then directly be used to generate potential trajectories.

Multi-modal fusion

The choice of sensing modality for computer vision determines the type and quality of features which can be extracted, and the range of conditions in which the system can operate. Monocular RGB imaging is the most widespread modality for scene understanding, providing rich color and texture information at high-resolution. Visible spectrum cameras are widely available, affordable and produce dense, highly structured data which is intuitive to label, being analogous to human vision. However, they are also highly sensitive to ambient illumination, lighting artifacts such as glare or shadows, and adverse weather which may obstruct or degrade the camera view: an outdoor vehicle cannot solely rely on this modality for safely navigating in adverse conditions (e.g. direct sunlight, night-time, fog).

Combining RGB imaging with other modalities is a promising research area with direct application to robotics. In particular, the release of compact low-cost RGB-D cameras like the Microsoft Kinect and Intel RealSense which provide per-pixel depth values via stereo matching has sparked many works which successfully incorporate distance information in a wide range of vision tasks [7]. However, these depth maps suffer from noise and low precision with increasing distance [52]. Depth maps can also be constructed from sparse LIDAR or radar scans. 3D LIDAR provides accurate and detailed scene geometry, and being an active sensor, can operate regardless of ambient illumination, however it requires favorable atmospheric conditions as it cannot penetrate through rain or fog. Conversely, due to its longer wavelength, radar is robust to poor weather and offers longer range but at the expense of resolution, making it poorly suited for characterizing small obstacles [75].

Thermal imaging is also becoming an increasingly attractive modality as an alternative or complement to RGB imaging due to its inherent robustness to illumination changes; it is especially well-suited for detecting people and other heat sources which stand out in their thermal intensity from the rest of the scene [26]. Although shifting away from the visible spectrum involves losing texture and color information, unlike laser and radar scans, thermal images remain visually interpretable, and can be labelled similarly to RGB images.

The complementary properties of these different imaging modalities can be leveraged with deep multi-modal fusion methods, with the goal of improving outdoor scene understanding capabilities compared to RGB-only perception. This also brings interesting challenges in the data collection stage and the choice of neural network architecture.

1.2 Project scope

Research questions

The goal of this work is to investigate how useful visual representations can be learned from multi-modal image data for the purpose of outdoor navigation. To this end, we formulate the following research questions which guide the development and experimentation:

- To what extent does parsing outdoor scenes in terms of driveability rather than specific semantic classes help when faced with unconstrained environments?
- How can multiple imaging modalities be combined for this purpose?
- What are the drawbacks and benefits of a multi-modal architecture for this task, compared to single-modality approaches?

Delimitations

As the first building block in an autonomous vision-based navigation system, this work specifically tackles robotic *perception*, to the exclusion of localisation, mapping, path planning or control methods. To assess the viability of the proposed methods, we rely on existing datasets for offline evaluation: the implementation over-head associated with data collection on a mobile platform, along with the computational constraints associated with real-time embedded operation, are outside of the scope of this work.

1.3 Outline

Chapter 2 sets the stage for the methods that follow by briefly describing relevant computer vision and deep learning concepts. **Chapter 3** presents recent related work in the fields of navigation-oriented scene understanding and multi-modal vision. Building on this existing research, **Chapter 4** provides an overview of our approach and experimental design for tackling the research questions. Since this project is largely data-driven, **Chapter 5** then delves into available datasets and describes the data preparation steps in our experiments. The bulk of our methods is described and evaluated in **Chapter 6**, **Chapter 7** and **Chapter 8**, and jointly validated in **Chapter 9**. Wrapping up, **Chapter 10** discusses the results of this work in relation to the research questions, addresses some of its limitations and challenges, and identifies possible next steps in the broader context of navigation, leading to interesting directions for future work. Lastly, **Chapter 11** concludes this research sandwich by summarizing the main motivations and findings.

Chapter 2

Background

This chapter sets a very brief theoretical foundation for the methods used in this work. We first give an overview of the sensing modalities that we consider for multi-modal perception, and their basic properties. We then introduce relevant machine learning concepts and deep neural architectures for pixel-level prediction.

2.1 Beyond monocular RGB imaging

2.1.1 Active vs. passive sensing

Passive imaging sensors capture naturally-occurring radiation in a scene such as reflected light or emitted heat, thus relying on external energy sources. Outdoors, the sun is the main source of visible light, which can be captured by visible spectrum cameras - however, the level of ambient illumination entirely determines the quality of images. Thermal imaging also falls in this category and is an attractive alternative, since it can capture heat emitted by objects regardless of illumination. In contrast, active sensors provide their own illumination by emitting radiation into the scene and measuring its reflection or scattering. These include laser and radar scanners, for instance.

This section focuses on infrared and depth vision, which may fall under either active or passive sensing, depending on the type of device.

2.1.2 Infrared imaging

All objects with a temperature above 0 K emit infrared radiation. While invisible to the human eye, infrared imaging devices can be used to convert this energy into a human-readable spatial intensity map. They operate within a certain spectral range, depending on the desired usage - the warmer a surface, the shorter its peak radiation wavelength. Capturing the infrared signature of a scene allows us to

"see" beyond the visible, without the distraction of variable illumination. The 3 infrared sub-divisions of interest are outlined in Figure 2.1 and introduced below.

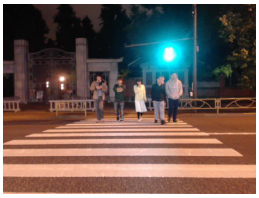
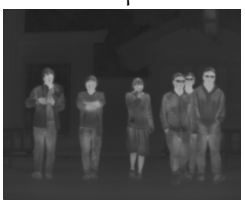
visible	NIR	MWIR	LWIR
0.4–0.7 µm	0.7–1.4 µm	3–5 µm	8–15 µm
			

Table 2.1: Wavelength range of the 3 main infrared imaging sub-divisions compared to the visible spectrum, with an example camera capture of the same scene in each band. Images are from the MIR object detection dataset [104] (cf. Section 5.1).

Near-infrared (NIR) wavelengths are adjacent to the visible spectrum and can be captured passively with a modified digital camera or specialized NIR detector. NIR images provide no color information but retain a high level of detail and contrast; they resemble grayscale visible-light images, while being less sensitive to ambient illumination and atmospheric haze. Similarly to the visible band, since only very hot objects emit significant radiation in the lower end of the infrared spectrum (over 2600 °C for a peak wavelength of 1 µm), NIR radiation in regular scenes is essentially limited to reflected light from the sun or other light sources. Thus, night vision capabilities require an active system. [114] elaborates on the properties and applications of NIR imaging in contrast with longer infrared bands.

Mid-wave infrared (MWIR) and long-wave infrared (LWIR) are referred to as thermal infrared - objects at room temperature emit significant radiation in these bands, and thus can be captured passively by a thermal camera. While details, textures or patterns may be lost in this range (eg. the road markings in Table 2.1), the scene's layout remains clear, and pedestrians or moving vehicles (the most common dynamic obstacles in urban environments) appear particularly salient due to their heat. Most thermal surveillance cameras operate in the LWIR range, since the infrared energy emitted at human body temperature peaks around 10 µm. An in-depth overview of thermal imaging technology and its applications to computer vision is given in [26].

2.1.3 Depth vision

Within the scope of this work, we treat depth data not as an unorganized set of 3D points, but as a structured 2D (or 2.5D) imaging modality, where each pixel encodes an estimate of its distance from the viewer. This allows us to employ standard 2D convolutional neural networks (CNNs) analogously across visible spec-

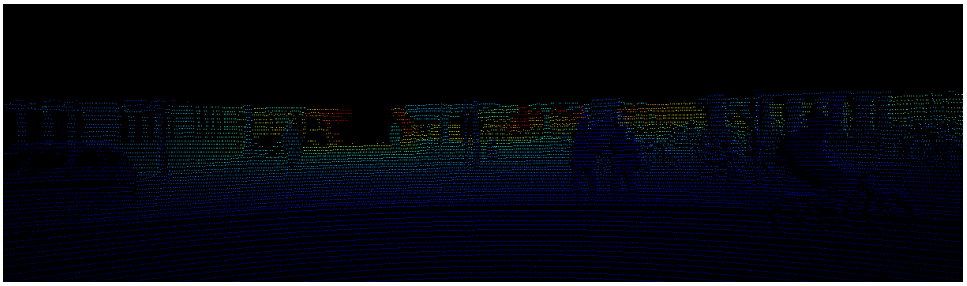
trum, infrared, and depth images, rather than requiring a dedicated architecture for 3D understanding [37].

Given a pair of monocular cameras with known parameters and partially overlapping fields of view (FOV), a disparity map can be constructed via stereo matching by finding pixel-wise correspondences between the two camera views, thus enabling RGB-D vision. 3D LIDAR technology is less accessible than RGB-D vision due to its cost, but its use is widespread in autonomous driving systems [118]. Sparse 2D depth maps can be computed from 3D laser scans by projecting point clouds to the image plane, as done in [30] for instance. For an overview on different 3D $\rightarrow$ 2D scan representation methods commonly used for scene understanding, we refer to [25].

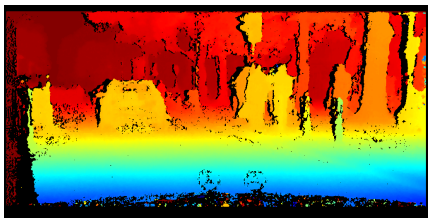
In the interest of time and space, we do not elaborate on depth sensing technologies here, but rather focus on relevant pre-processing steps for depth-based feature extraction.

Depth completion

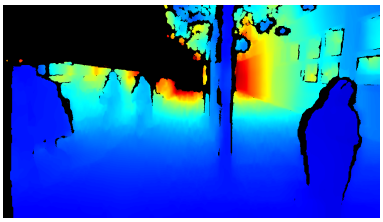
Depth maps generated from raw range-sensing data are sparse by nature. As shown in Figure 2.1, 3D LIDAR scans projected onto the image space appear as structured scan lines with only a small portion of pixels encoding a depth value, while stereo vision produces dense but noisy and discontinuous depth data with specks or patches of missing values.



(a) 3D LIDAR scan (Velodyne HDL-64E) from the Kitti dataset [30]



(b) Stereo disparity (OnSemi AR0331 sensor) from Cityscapes [21]



(c) Stereo depth (ZED camera) from ROB7 project [46]

Figure 2.1: Color visualization of sparse depth maps from existing datasets. Missing values are shown in black.

Input data sparsity is an issue for CNNs, since they are designed following the assumption that every pixel encodes a valid observation. For instance, [110] found that using raw stereo depth maps in a deep multi-modal pipeline gave poor results for outdoor scene segmentation. Although effective methods have been proposed for directly feeding sparse depth maps to CNNs (eg. through masked convolutions which ignore missing values [107] or training strategies which encourage sparsity-invariant learning [51]), these require special considerations in the architecture design. Therefore, the most straightforward approach remains to first infer a dense depth map as a pre-processing step, and using this dense image as input to a standard CNN, as done in [110, 56, 109].

Given a sparse depth image with missing values, the aim of depth completion is to recover an estimate of the true depth value for every pixel. Classical approaches employ hole-filling strategies such as filtering and in-painting [3]. Leveraging visual information in order to guide the depth estimation process has also proven beneficial, however, this places additional constraints on the availability and quality of RGB images for every depth frame. While the current state-of-the-art is dominated by image-guided CNN-based approaches, [59] demonstrates that traditional image processing techniques are able to perform on-par with deep architectures on depth completion benchmarks, at the fraction of the computational cost and without the overhead of training or reliance on additional modalities.

2.2 Deep learning for pixel-wise classification

The question *"what is in this image?"* can be answered at different levels, as illustrated in Figure 2.2. At the coarsest level, **image classification** generates a single description for the whole image, based on the presence of one or more classes. **Object detection** additionally provides the general location of different class instances as bounding box coordinates. **Semantic segmentation** operates at the pixel-level by partitioning the image into class regions: such dense predictions are crucial for vision-based navigation, in order to precisely delimit driveable regions or obstacles. **Instance segmentation** goes a step further by also distinguishing between instances of the same type. While this can be useful for human-robot interaction for instance, where it may be desirable to uniquely identify and segment people in the image rather than grouping them under a common label, we do not consider this step necessary for navigation-oriented scene understanding - therefore, we focus on semantic segmentation.

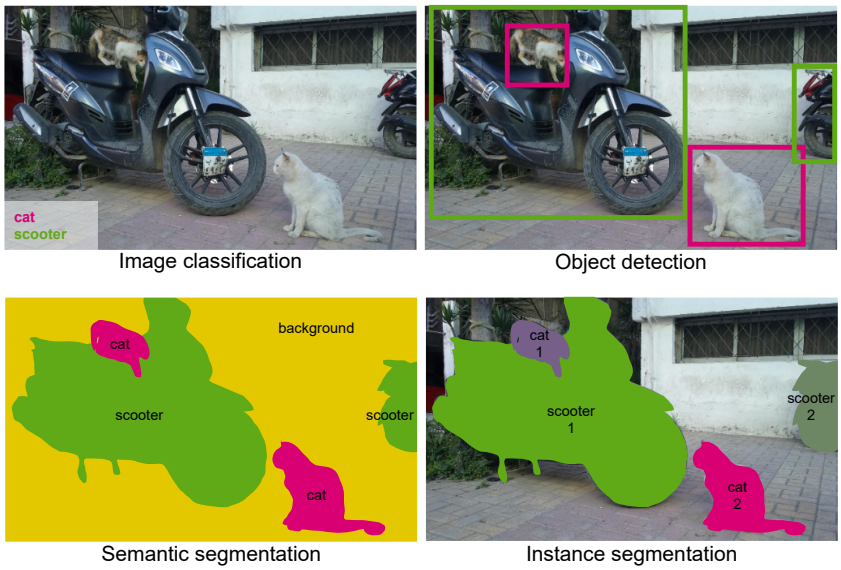
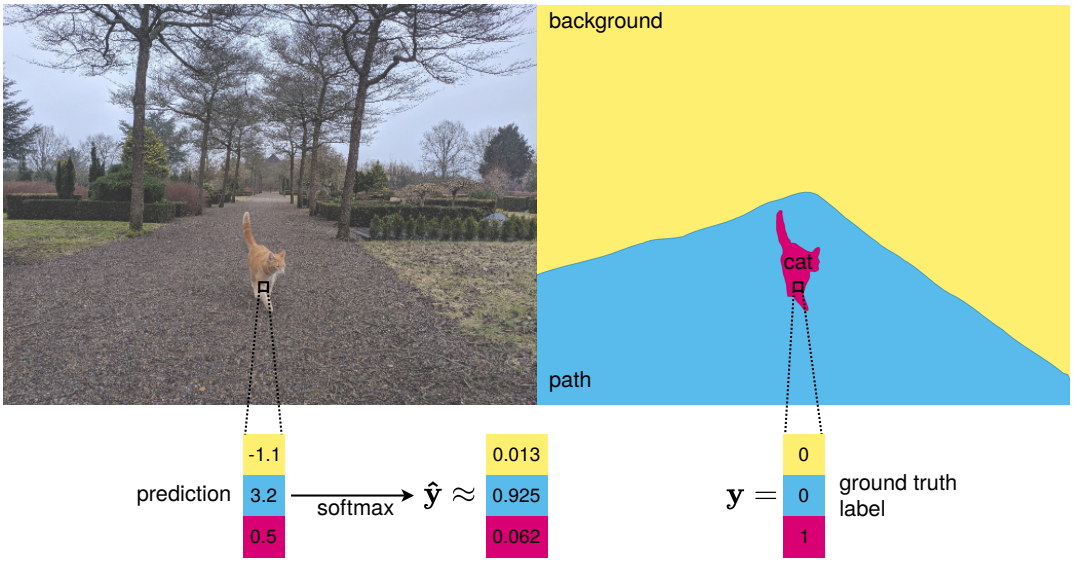


Figure 2.2: Egyptian cat models demonstrating different image description tasks, from coarsest (top left) to finest (bottom right).

2.2.1 Semantic segmentation as pixel-wise classification

When framed as a machine learning problem, semantic segmentation amounts to inferring a label for every pixel based on a set of pre-defined classes. In a fully supervised learning scheme, ground truth labels are provided on a per-pixel basis at training time. Much like any classification task, the goal is then to learn optimal model parameters by minimizing the error between the true and the predicted

labels. This error is captured by a loss function which penalizes incorrect predictions, with the loss being computed per pixel, and then averaged or summed over the whole image. The simplest loss weighting scheme consists of giving each pixel equal weight. However, more advanced loss weighting schemes have been proposed in order to compensate for class imbalance [67, 4], or achieve finer segmentation along boundaries [88], for instance. It is also common practice to define a *void* class for unlabelled pixels which are explicitly ignored in the loss computation [21].



$$L_{CE}(\mathbf{y}, \hat{\mathbf{y}}) \approx -[0 \cdot \log 0.013 + 0 \cdot \log 0.925 + 1 \cdot \log 0.062] \approx 2.78$$

Figure 2.3: Example of a cross-entropy loss calculation for a single pixel in a 3-class semantic segmentation task, where the label is one-hot-encoded. Predicted values are chosen arbitrarily and the pixel size is exaggerated for illustration purposes.

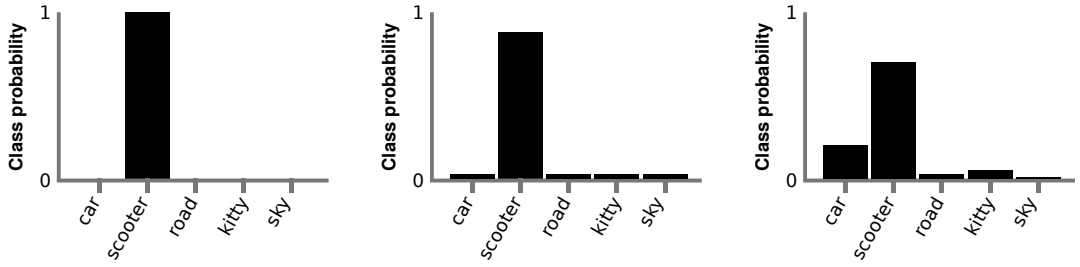
The most widespread loss function for classification is cross entropy; minimizing cross-entropy amounts to maximizing likelihood. This first requires normalizing the model's raw prediction $\hat{\mathbf{y}}'$ to obtain a probability distribution $\hat{\mathbf{y}}$ across the set of classes C (such that $0 < \hat{y}_i < 1 \; \forall i \in C$ and $\sum_i^C \hat{y}_i = 1$) - for a multi-class problem, this is typically done by applying the softmax (normalized exponential) function. The softmax function and cross-entropy loss L_{CE} are given by:

$$\hat{y}_i = \text{softmax}(\hat{\mathbf{y}}')_i = \frac{\exp \hat{y}'_i}{\sum_{j \in C} \exp \hat{y}'_j} \qquad L_{CE}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i \in C} y_i \log \hat{y}_i$$

The label's probability distribution $\mathbf{y}$ can be expressed as a one-hot encoded vector, where the target class has a probability of 1 and the other classes have a probability

of 0. As illustrated in Figure 2.3, when using a "hard" one-hot-encoded ground truth label, only the target class probability contributes to the cross-entropy loss value: the model is encouraged to maximize the log-likelihood of the target class, with the rest of the classes being considered categorically incorrect and infinitely far from the target.

Alternatively, ground truth labels can be encoded as "soft" vectors, where instead of assigning each observation to a single correct class, labels describe class membership probabilities [27]. We show two examples of soft labels in Figure 2.4 compared to a standard hard label. In their most generic form, soft labels can be generated via label smoothing, where the ground truth is taken as a weighted average between the original hard target and a uniform distribution across all classes - [103] presents this (now widely adopted) technique for image classification and shows how it has a regularizing effect, by discouraging high-confidence predictions, and thus aiding generalization. However, uniform label smoothing treats each incorrect class as equally probable. More interestingly, soft labels can be used to express known relationships between classes (eg. similarity [120] or hierarchy [23, 9]) or to capture natural ambiguity in the data (eg. at the borders of segmentation masks [34] or due to inconsistent/subjective labels from multiple annotators [72]).



(a) Hard one-hot-encoded label

(b) Soft label: uniform label smoothing [103]

(c) Soft label: distribution based on semantic similarity

Figure 2.4: Different label distributions illustrated with a dummy example. Here, *scooter* is the target class.

Models can be trained on soft labels with standard classification loss functions such as cross-entropy or Kullback-Leibler divergence (relative entropy) eg. as in [28, 23]. Kullback-Leibler divergence measures the difference between two probability distributions (in this case, the predicted $\hat{\mathbf{y}}$ vs. ground truth $\mathbf{y}$ class probability distributions). When $\mathbf{y}$ is fixed, minimizing Kullback-Leibler divergence L_{KL} is equivalent to minimizing cross-entropy L_{CE}, since they only differ by a constant factor:

$$L_{KL}(\mathbf{y}||\hat{\mathbf{y}}) = \sum_{i}^{C} y_i \log \frac{y_i}{\hat{\mathbf{y}}_i} \qquad L_{CE}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i}^{C} y_i \log \hat{\mathbf{y}}_i = \sum_{i}^{C} y_i \log \frac{1}{\hat{\mathbf{y}}_i}$$

2.2.2 Architectures

Given an input image, deep semantic segmentation networks output dense pixel-level class predictions. This is most commonly tackled as a two-stage process: convolutional feature extraction, followed by expansion to recover the original image resolution. The main challenge lies in capturing high-level context information which plays a key role in scene understanding, while retaining detailed local information which is necessary for precise segmentation [67].

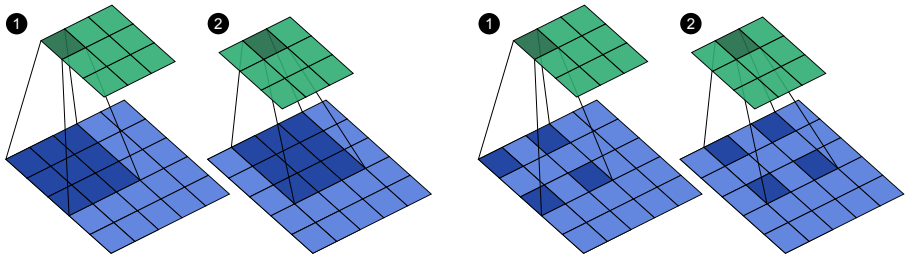
Encoder-decoder models

A highly influential work is the Fully Convolutional Network (FCN) architecture proposed in [67] (with further analysis in [95]), which demonstrates how well-established image classification CNNs such as VGG [99] can be leveraged for deep semantic segmentation by replacing fully connected layers with 1×1 convolution to produce a coarse feature map for each class. This *encoding* stage is followed by up-sampling through deconvolution for pixel-wise prediction. In order to recover the spatial detail which is lost during down-sampling, FCN also demonstrates how intermediate outputs from shallower layers can be incorporated in the up-sampling stage through skip connections to achieve finer segmentation.

Many works derived from FCN employ similar encoder networks and primarily differ in the decoder design. For instance, U-Net [88] proposes multi-stage up-sampling which mirrors the encoder depth; unlike FCN, the filter dimensionality is reduced to the number of classes only at the prediction stage. SegNet [4] follows a similar structure, but proposes a more efficient technique for incorporating boundary information in the up-sampling stage: only the pooling indices are transferred to the decoder, rather than the full encoder feature maps.

Dilated convolution

In the feature extraction stage, down-sampling feature maps (via strided convolution or pooling) effectively widens the receptive field [70] of subsequent filters without needing to increase their kernel size - making it a widely used technique for image classification [58, 99, 42]. However, down-sampling comes at the expense of spatial resolution - this loss of detail is undesirable for dense prediction tasks. While encoder-decoder approaches partially address this by adding deconvolutional layers, an alternative consists of performing *atrous* convolution (coined in [82], later called *dilated* convolution in [117]) for feature extraction, where the filter is up-sampled with zeros, thus covering a larger input area than standard convolution with the same number of weights, while preserving spatial resolution, as illustrated in Figure 2.5. Dilated convolution has been successfully applied to semantic segmentation as an alternative to encoder-decoder approaches [117, 16].



(a) Standard convolution (dilation rate of 1) **(b)** Dilated convolution (dilation rate of 2)

Figure 2.5: Illustration of standard vs. dilated convolution (no stride and no padding), with the input map in blue, and the output map in green - only the first two steps of the convolution are shown. A 2×2 convolutional filter with dilation rate of 2 covers the same input area as a 3×3 undilated filter, with less than half the number of weights. The diagrams are largely inspired by [22].

We have only scratched the surface of deep semantic segmentation here, highlighting two important directions in existing work. For a much more comprehensive overview, we refer to the recent survey in [76] and further explanation of different architecture variants in [31].

2.2.3 Training neural networks

Parameter updates

The model's parameters are first initialized either randomly (typically drawn from a normal or uniform distribution with carefully chosen mean and variance [33, 43]), or with previously learned parameters from a pre-trained model (transfer learning). In gradient descent learning, parameters are updated iteratively in the direction that (locally) minimizes the loss function, based on its gradient with respect to every parameter (computed via backpropagation). In practice, samples are commonly fed to the model in a series of mini-batches, such that parameter updates are based on an estimate of the true gradient computed from a small subset of the training set - the batch size is an important hyper-parameter in this case. Another crucial hyper-parameter is the learning rate, which determines the step size of parameter updates - too big and we'll keep over-shooting to different planets, too small and we'll be 80 by the time we manage to reach the bottom. Plain gradient descent sets a global learning rate for all parameters which remains constant during training unless explicitly adjusted. This motivates the use of adaptive gradient descent algorithms such as Adam [54] which calculate the learning rate on a per-parameter basis. [91] gives a comprehensive overview of gradient descent optimization methods.

Regularization

A common issue in neural networks is over-fitting: the model learns to over-characterize the training set, achieving low training error but failing to generalize to new inputs. To limit over-fitting, regularization techniques can be applied at the data level, during weight optimization or through modifications of the architecture itself [60]. We briefly present a few of these techniques here.

Data augmentation consists of artificially expanding and diversifying the training data by modifying the original training samples, but only to the extent that their essential content is not affected (ie. data labels can be preserved). The goal is to expose the model to a distribution of samples which is closer to that of the real unseen data encountered at run-time. For image semantic segmentation, these modifications can include geometric transformations (cropping, flipping, scaling, skewing, elastic deformation etc), color transformations (shifts in brightness, contrast, hue, etc.), degradation (eg. additive noise, blur, pixel drop-out) or more complex alterations such as adding synthetic rain. See [97] for a survey on data augmentation methods in computer vision.

Dropout [100] was proposed as an effective (and now widespread) technique to prevent deep neural networks from over-relying on the presence of specific features to make predictions - as these features may simply be an artefact of the training set. By randomly de-activating a proportion of neurons at every training iteration, the network is encouraged to explore different parts of the feature space in favour of more robust, generic representations.

Batch normalization [48] is applied in hidden layers to mitigate shifts in the distribution of their inputs as the network's parameters are updated. Each feature map is normalized independently along the batch dimension based on running estimates of batch statistics, and a linear shift-scale operation is then applied based on learnable parameters. This technique has shown to allow for training which is faster and less sensitive to initialization.

2.2.4 Evaluation metrics

Counting correct pixels

Semantic segmentation metrics typically treat each pixel as an independent sample, with the prediction taken as the class with the maximum probability. For multi-class segmentation, the prediction is binarized using a one-vs-rest scheme, such that each pixel can be a true positive (TP), true negative (TN), false positive (FP), or false negative (FN) [29]. Here we define the segmentation metrics used in our evaluation.

Accuracy measures the proportion of correct predictions:

$$A = \frac{TP + TN}{TP + TN + FP + FN}$$

Intersection over Union (IoU) is widely used in semantic segmentation benchmarks. It captures the amount of overlap between two binary masks, penalizing both over- and under-segmentation, and is expressed as:

$$IoU = \frac{TP}{TP + FP + FN}$$

Unlike accuracy which, as a global metric, is inherently skewed towards the performance on the majority class, IoU, precision and recall can be reported on a per-class basis, giving more insight into the performance for under-represented classes.

Quantifying error

Following [23], when tackling segmentation as a pixel-wise ordinal classification task, we also make use of the following regression-based metrics, where n is the total number of samples (pixels), and $e_i = y_i - x_i$ is a pair-wise error for a single pixel i with ground truth y_i and predicted value x_i:

Mean absolute error (MAE) pretty straight-forward

$$MAE = \frac{\sum_{i=1}^{n} |e_i|}{n}$$

Root mean squared error (RMSE) increases the penalty as the pair-wise error grows (an error of 2 is four times worse than an error of 1), and the root brings it back to the original unit

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} {e_i}^2}{n}}$$

Root mean squared log error (RMSLE) scale-invariant, and penalizes over- more than under-estimation, which is quite cool

$$RMSLE = \sqrt{\frac{\sum_{i=1}^{n} [\log(y_i) - \log(x_i)]^2}{n}}$$

Chapter 3

Related works

We explore relevant scene understanding literature at the intersection of the fields of autonomous driving, where the challenges of outdoor operation and dynamic obstacles are especially apparent, and robotics, where the notion of action-oriented or affordance-based perception shows promising applications for control and navigation. Since this work tries to take both a functional and a multi-modal approach to scene understanding, we consider both aspects in this chapter, while limiting our scope to deep learning-based methods which exclusively rely on egocentric images for making predictions at run-time.

3.1 Functional scene understanding

[15] presents an interesting framework for thinking about vision-based autonomous driving, identifying three main approaches which vary in their level of perceptual abstraction:

- at the lowest level, **mediated perception** approaches parse the entire scene into explicit visual representations, which must then be interpreted and condensed into driving decisions by a separate part of system.
- at the highest level of abstraction, **behaviour reflex** approaches bypass visual representations altogether for end-to-end vehicle control.
- [15] argues for an affordance-based **direct perception** approach which falls between these two extremes: sensory input is distilled into useful indicators which are tailored to a driving task, and can be directly used a basis for decision-making.

In the realm of robotic vision, [73] follows a similar line of thought, arguing for *action-oriented* scene understanding which recognizes potential functions, opportunities, or trajectories in the environment, as opposed to traditional *descriptive* scene understanding which reasons at the level of objects or scene attributes.

Figure 3.1 illustrates the different perception paradigms for autonomous navigation which we discuss in the remainder of this section.

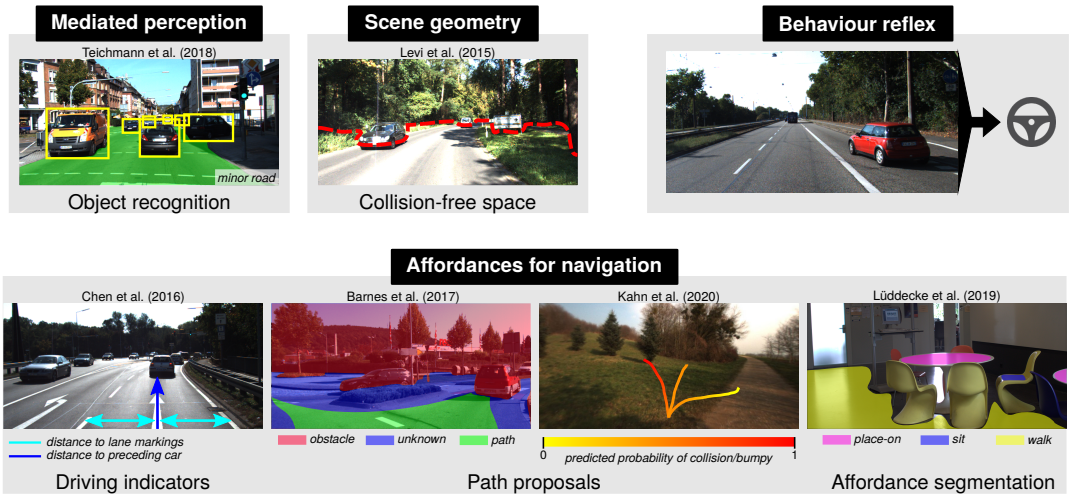


Figure 3.1: Visual perception for navigation: highly non-exhaustive overview. Images were created based on the methods from related works presented in this section [105, 63, 15, 5, 53, 71], using samples from the Kitti [30] (roads), Freiburg Forest [110] (off-road) and RGB-NIR Scene [13] (indoor) datasets (cf. Section 5.1)

3.1.1 Mediated perception: describing what you see and nothing more

Image-based urban scene understanding is most commonly tackled as an object recognition task, either at the pixel-level (where each pixel is labelled as part of a relevant class) or at the instance-level (where bounding box coordinates are regressed for each instance of a class). The release of large-scale annotated driving datasets like Kitti [30] has sparked significant work in this direction - including task-specific methods (eg. pedestrian detection) and multi-task architectures which, for instance, jointly perform road segmentation, car detection, and scene classification [50] (top-left in Figure 3.1).

However, parsing the entire scene in terms of specific semantic categories (eg. pedestrian, cyclist, car, road) requires an over-arching logic to determine where the vehicle should/can safely drive, hinders generalisation to unseen obstacles, and involves fine-grained labelling of objects which may not be relevant for vehicle control (aside from the fact that they should be avoided) [6]. These considerations have sparked works which try to learn more task-oriented representations with varying degrees of abstraction, where the scene is directly parsed in terms of navigability rather than appearance, and whose output can directly be used by a planning algorithm.

3.1.2 Scene geometry: just because it's free doesn't mean it's a good idea

An alternative to object-based semantic labelling consists of parsing the scene geometrically in terms of occupied vs. free space. For instance, [63] labels free space boundaries with LIDAR data and trains a CNN to regress the position of general obstacles in monocular images (top-middle in Figure 3.1). However, identifying collision-free space in the scene is not sufficient for ensuring safe and robust navigation: some areas may be traversable, but nevertheless unsuitable to drive into (e.g. a puddle or a ditch); on the contrary, as confirmed in [53], some elements which are detected by range sensors as obstacles (e.g. tall grass) may in fact be traversable. [5] (bottom-middle in Figure 3.1) extends this approach by learning to segment driveable routes in urban scenes using odometry data for weak supervision, but pre-supposes the presence of an explicit path.

3.1.3 Behaviour reflex: direct mapping from image to action

A fully end-to-end "behaviour reflex" approach (in the top-right of Figure 3.1) consists of directly regressing the vehicle steering angle from a camera capture, rather than learning an intermediate visual representation of the scene [50]. However, as pointed out in [15] and [5], learning a direct mapping from an image to a control action is an over-abstraction which may fail to capture the ambiguity or complexity of driving decisions - for instance, in cases where there are multiple valid directions (e.g. in case of a cross-road or open area). This approach has also shown serious limitations when faced with an increasing number of dynamic obstacles [19]. A recent study on robotic action [124] also suggests that, compared to predicting a direct mapping from raw image to control output, semantic scene segmentation greatly improves task performance in urban and off-road navigation. Thus, for real-world driving, it may not be desirable to bypass visual representations entirely.

3.1.4 Affordances for navigation

Appearance-based or geometric scene understanding are not directly interpretable in terms of potential action, and fail to capture degrees or levels of traversability: while it is *possible* to drive over rough terrain or grass, it may not be practical or preferable if there is a paved area available. An interesting notion which can be used to capture these differences is that of "affordance", which originated in psychology to describe how we perceive our environment in terms of how we can use or act on it to our harm or benefit [32]. This concept has led to interesting developments in robotic perception, especially for service and collaborative robots: reasoning about objects in functional terms (eg. reachable, openable, sittable) rather than semantic labels has been shown to improve generalisation for recognising suitable

actions and grasp poses [2]. However, works which tackle affordance-based perception for robotic navigation remain scarce, compared to the bulk of literature pertaining to object/tool manipulation or human-robot interaction [49]. The very recent survey in [40] highlights the potential of interpreting the environment in terms of visual affordance for intelligent decision making in a wide range of tasks, also noting that research in this field is currently largely limited to indoor scenes.

Here, we focus on methods which predict navigational affordances from images, spreading our attention across the fields of robotics and autonomous driving. A common approach is to rely on exploration-based learning, where information from past interaction with the environment is used to learn useful representations and guide present action. This requires either realistic simulation test-beds, or a real system with physical access to various environments in which exposure to potential hazards through trial and error is acceptable.

For instance, an affordance-based "direct perception" approach for autonomous driving proposed by [15] (bottom-left in Figure 3.1) consists of learning traffic indicators in a driving simulator such as heading angle or distances to preceding cars which can then be directly used by a driving controller, however this method relies on pre-defined visual cues (lane markings, cars) and thus cannot operate outside of a highly constrained urban landscape. In a broader robotic context, [86] uses pixel-wise affordance maps to encode where an agent can safely move in an unpredictable first-person shooter simulator; affordance segmentation is learned through continuous active interaction. Out in the real world, [53] (bottom-middle in Figure 3.1) demonstrates how physical navigation affordances (bumpiness, probability of collision) can be learned in a fully unsupervised manner by an autonomous robot roaming outdoors.

Self-supervised learning through exploration eliminates the need for manual labelling, but is a time and resource intensive process, especially for navigation applications. Indeed, it requires the agent to traverse vast, diverse areas and experience collisions in order to learn affordances, even though they could have easily been human-annotated, since we are naturally skilled at assessing whether an area is navigable by mere visual inspection. Although applied in indoor scenes and not limited to navigation, [90] and more recently [71] (bottom-right in Figure 3.1) demonstrate how dense affordances can be predicted at the pixel-level from RGB images in a fully supervised manner, with ground truth segmentation masks.

3.2 Multi-modal data fusion for scene understanding

Many works have demonstrated the benefit of combining RGB imaging with infrared [38, 110, 102, 46] or depth data [109, 101] for outdoor scene understanding and object recognition. An overview of multi-modal fusion methods for deep scene understanding is given in [121], [25] and [56].

When to fuse? Fusion can be performed at different stages in the network, which we briefly present here, along with their main trade-offs.

Early fusion methods combine imaging modalities at the pixel level, before being fed to the network - for instance, by treating each modality as an additional image channel as in [110]. This approach is perhaps the simplest to implement and the most light-weight, since standard single-input CNN architectures can directly be applied. It also scales well to an increasing number of modalities. However, it also requires that all modalities be present during training and at run-time, and that images are precisely synchronized and aligned at the pixel level across modalities.

Late fusion methods employ a separate network for each modality, and combine the output features in the final stage, at the prediction level. This has the advantage that each branch can be designed and trained independently. However, each additional input brings a significant cost in terms of the network size and computational requirements. This fusion scheme also prevents any information from being shared across modalities during learning.

Middle or *hybrid fusion* methods treat each modality as a separate input, while allowing intermediate representations to be combined or shared at various stages. For instance, [101] and [102] employ a multi-modal encoder-decoder architecture for semantic segmentation, with modality-specific encoders and a common decoder; in the encoder stage, feature maps are fused after each layer. A similar network structure is proposed in [38], but fusion is performed in the decoding stage, where modality-specific feature maps are incorporated via skip connections. Developing a middle fusion schemes requires making many design choices about e.g. where and how should the modalities be fused?

How to fuse? An important consideration in middle and late fusion schemes is how to combine feature maps from modality-specific layers. A common approach is to simply fuse them via concatenation or element-wise addition [38, 110, 102, 101], which gives each modality equal weight regardless of the input quality. In practice, different modalities may provide different levels of usefulness in different conditions, different image regions or for different classes. This has prompted more complex, adaptive fusion methods where the weight of each modality is learned. For instance, the mixture of experts scheme in [108] proposes class-wise gating sub-networks which learn a probability distribution over the modality-specific experts. More recently, the same authors have presented a model adaptation fusion [109] module which learns the correlation between between modality-specific feature maps and weighs them element-wise at different stages in the network.

Chapter 4

General approach

4.1 What are we trying to predict?

Our aim is to parse outdoor scenes at the pixel-level in terms of how a vehicle or robot can navigate in it, solely relying on images captured from its point of view. Loosely inspired by the broad semantic classes $\{path, unknown, obstacle\}$ defined in [5] for urban scene understanding and the robotic action plausibility ratings $\{possible, plausible, impossible\}$ proposed in [72], we define three pixel classes to characterize the driveability/navigability level of a certain area in an image:

- *Preferable*: where we would expect the vehicle to drive
- *Possible*, but not preferable: areas which are technically navigable but more challenging or less suitable, and would not be chosen as a first resort
- *Impossible* or undesirable: any part of the scene which is unreachable (eg. the sky) or should be unconditionally avoided (obstacles, hazardous terrain)

Figure 4.1 shows how an outdoor scene could be segmented according to these three driveability levels.

Figure 4.1: Where can/should I drive? Coarsely & manually annotated outdoor scene based on the proposed driveability level definition: *preferable*, *possible*, *impossible*.

Limiting the number of levels to 3 keeps the labelling process manageable and relatively straight-forward, while still leaving room for a "grey-zone" unlike binary classification which would fail to capture any degree of driveability.

4.2 Battle plan

Learning driveability affordances

Architecture From a technical standpoint, our task aligns with standard semantic scene segmentation: we want to predict a label for every pixel in the image based on pre-defined categories, and thus can employ existing image segmentation architectures. Similarly to [5], we pick the SegNet architecture introduced in Section 2.2.2 as a base network for pixel-wise prediction. While it has been surpassed by other state-of-the-art segmentation architectures eg. [123, 109], it is widely documented and relatively lightweight thanks to its efficient up-sampling technique, making it a solid choice for prototyping and evaluation.

Off the beaten path Our task nevertheless fundamentally deviates from classic descriptive computer vision (as characterized in [73]) in several ways:

- we are not trying to address the question *what is in this image?*, but rather directly asking *how can I operate in this image?*, or more specifically, *where can I go?* Such an action-oriented approach is arguably more *useful* for an autonomous vehicle than learning descriptive representations, but introduces vagueness and ambiguity - while there is usually very little doubt on what an object *is*, it may be more difficult to reach strong consensus or certainty about what an object is *for* or how it can be interacted with.
- segmentation of objects by name (eg. vehicle, tree) is best achieved by learning specific recurring patterns which are unique to those objects; in contrast, recognizing whether an area is driveable or not requires learning a very broad representation of what an obstacle could possibly look like, as many different types of features and scene elements may fall under this category.
- although it is not directly measurable, we are defining the notion of driveability not as a semantic category but as an ordinal quantity, with some areas in the image being more driveable than others.
- typically, every pixel in a prediction is given equal importance, regardless of its position in the image - however, in the context of navigation, areas in the immediate vicinity of the vehicle have a higher significance as guides for potential action.

Lazy shortcuts When it comes to generating ground truth data, our task is facilitated by the reasonable assumption that the notion of driveability can be inferred from the type of object in the image (eg. a car will always be *impossible* to drive on). As discussed in [2], such a mapping from semantic labels to affordance is somewhat reductive as it does not take any contextual information into account, but remains a common approach, since it allows us to leverage existing datasets with pixel-wise semantic labels for fully-supervised learning. In addition, unlike other types of visual affordances which often inherit the challenges of multi-label learning [40] (eg. due to objects having multiple possible functions or uses), our driveability definition can be tackled as a single-label pixel classification problem.

Gray rainbows While it is commonplace to preserve color information in visible spectrum images for scene understanding [4, 5, 105], we choose to convert them to a single-channel grayscale representation for two main reasons. First, we speculate that colour may add unnecessary or distracting information when trying to learn such an abstract concept as driveability, and second, by reducing the number of channels to 1, visible spectrum images can be seamlessly inter-changed with depth or infrared images (which are inherently single-channel) at the input of the network, without needing to adapt the number of weights in the first layer.

Learning order by softening things In order to model the ambiguity and order between the three driveability levels, we opt for a soft labelling approach (introduced in Section 2.2.1). Soft labelling schemes have successfully been applied to semantic segmentation in prior work [28, 34], however these works only consider a binary classification case, and generate pixel labels not based on inter-class relations, but on spatial location, to capture ambiguity along object boundaries. Conversely, works which demonstrate the use of soft labels for ordinal classification [28, 23, 66], apply it to other tasks such as full-image classification (eg. age estimation, aesthetic quality prediction or medical diagnosis) or pixel-wise regression (eg. depth estimation). To the best of our knowledge, the use of soft labels for rank-based scene segmentation has not yet been investigated.

Unequal rights for unequal pixels We investigate how to focus learning towards areas of particular interest for navigation, drawing from the insights in [64, 125], and adapting the method from [88] to develop a pixel-wise loss weighting scheme.

Learning from multi-modal data

The data we want We have introduced some of the interesting properties of infrared and depth images in Section 1.1 and Section 2.1, and briefly reviewed CNN-based fusion approaches for combining them with visible spectrum imaging in

Section 3.2. Thermal imaging and stereo depth seems to be particularly relevant sensing modalities for our task. Humans, animals and moving vehicles are among the most common and safety-critical dynamic obstacles a robot may encounter, and due to their heat, hold a characteristic signature when imaged with a LWIR camera that contrasts with surrounding terrain, buildings, vegetation, and scene clutter. Stereo depth is often readily available on robotic platforms due to the prevalence of integrated RGB-D cameras, providing appearance-agnostic pixel-wise distance information - without this geometric information, it may be difficult to distinguish between flat surfaces and real obstacles. We therefore aim to incorporate both of these modalities into the prediction, as a complement to visible spectrum imaging.

The data we have Our experimentation is limited by the extent of available data - we explore our options in detail in Chapter 5, and fail to find any outdoor tri-modal RGB-D-T datasets with pixel annotations. As detailed in Section 5.2, we resort to using Freiburg Forest [110] (RGB-D-NIR with pseudo depth) and Cityscapes [21] (RGB-D with real stereo depth) for the bulk of our model training and evaluation. Thermal imaging is incorporated in our final experiments, when jointly evaluating selected methods on other bi-modal datasets.

Special treatment for special sensors Before delving into multi-modal fusion, we investigate the effect of pre-processing for depth or infrared images, noting a lack of existing experiments addressing this. For instance, [38, 41] make no mention of data augmentation for RGB-T and RGB-D segmentation respectively, while [108, 102, 109] list augmentations used during training without making a distinction between different modalities. Depth completion seems to be standard procedure [109, 56], but is performed without justification in these works.

Fusion architecture Similarly to [92, 38, 109], we take an early fusion approach as our multi-modal segmentation baseline. For deep fusion, we explore the adaptive fusion approach proposed by [109]: rather than simply concatenating or adding features extracted from different modalities, a convolutional fusion unit learns to combine them in an optimal way. We implement the fusion unit in 3 multi-branch configurations (middle, late, and a combination of both), designing the architecture in a modular way: SegNet's efficient decoding technique is preserved, and the full extent of the network can be initialized with pre-trained SegNet weights and then trained end-to-end in a short period, with only the fusion unit's parameters needing to be learned from scratch. This approach contrasts with the bulk of existing multi-modal semantic segmentation methods for autonomous driving [25], which employ FCN-based decoders and learn their weights from scratch. We also evaluate our deep fusion architecture in a tri-modal configuration, which to the best of our knowledge has not been done in prior work.

TL;DR

Figure 4.2 gives an overview of our experimental approach, following the structure of this report. In **Chapter 6**, we train SegNet to segment visible spectrum images based on our proposed driveability definition, and evaluate its performance on in-domain and out-of domain samples, compared to a naive mapping from object classes to driveability in the post-prediction stage. The soft labelling and loss weighting methods follow in the same chapter. In **Chapter 7**, we train SegNet to segment driveability in NIR, pseudo depth, and stereo depth images, which gives us a measure of how useful each of these modalities can be as a stand-alone predictor before our fusion experiments. **Chapter 8** is where all the fusion magic happens. **Chapter 9** serves as a final wrap-up, where we select the most promising methods from the three previous chapters, and jointly validate them in two experiments on a wider range of datasets.

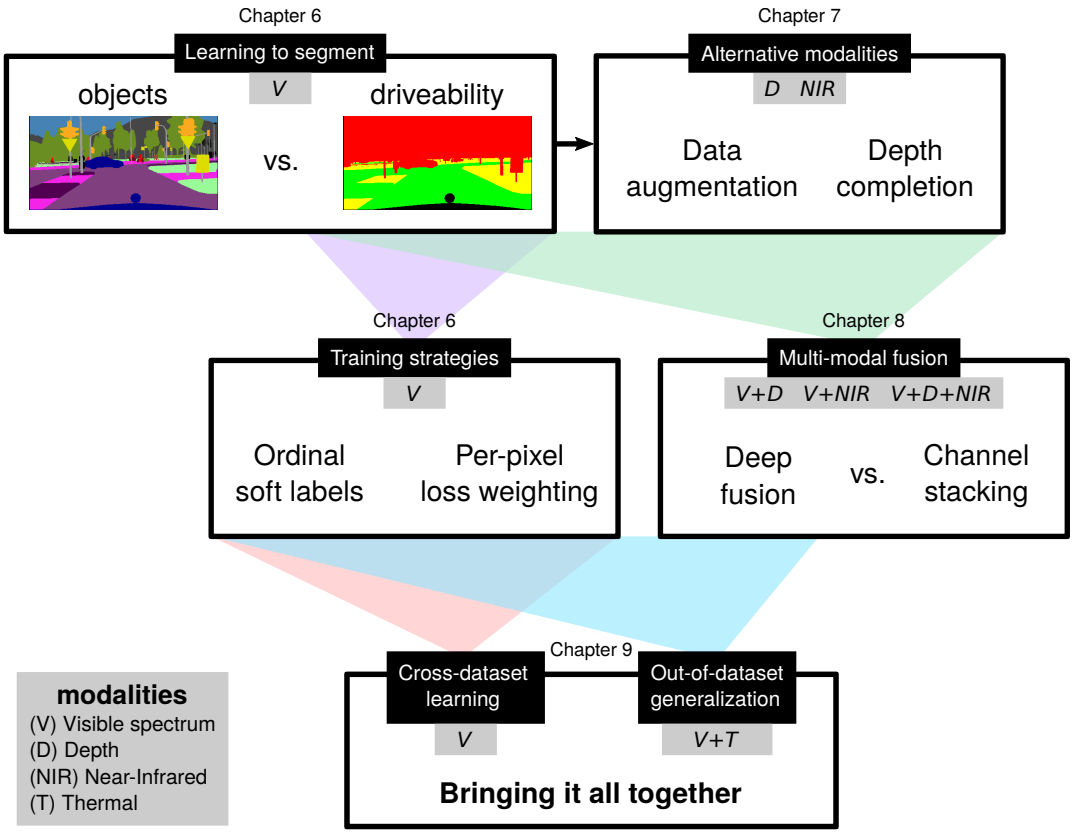


Figure 4.2: Roadmap of the methods we explore from Chapters 6 to 8, followed by 2 final experiments in Chapter 9. For each chapter, we specify the imaging modalities included in our evaluation. The image pair in the top right is a ground truth sample from the Cityscapes dataset (cf. Section 5.1).

4.3 Implementation details

Software This project is developed in Python 3, primarily relying on PyTorch Lightning [83, 24] as a deep learning framework, Weights & Biases [10] for experiment logging, Scikit-learn [84] for computing certain evaluation metrics, OpenCV [12] for simple image processing, Albumentations [14] for data augmentation, and Labelbox [61] for image annotation. We also make limited use of ROS [87] and Matlab for interfacing with certain datasets, and R for visualization.

Hardware Deep learning models are trained on a NVIDIA Tesla V100-SXM3-32GB GPU via a dedicated cloud service[1].

Deliverables The project source code is made available as a public repository:

`https://github.com/glhr/learning-driveability-heatmaps`

with the bulk of the implementation in the following sub-module:

`https://github.com/glhr/lightning-segnet`

Additionally, we showcase some of our main results on video sequences from different datasets - details and links are included in Section A.7 (appendix).

Reproducibility We set a global random seed and enforce deterministic PyTorch operations[2] in order to ensure identical results across repeated runs of the same experiment on a given device (both for training and evaluation), and to ensure that differences in performance between methods are not due to random variation (eg. in the random weight initialization, dropout patterns, data augmentation sequence).

[1] `https://www.claaudia.aau.dk/platforms-tools/compute/gpu-cloud-ai/`

[2] `https://pytorch.org/docs/stable/notes/randomness.html`

Chapter 5

Data is everything

Neural networks can only learn from the examples we show them. Training models to give accurate predictions in diverse situations requires collecting or generating large amounts of diverse data. While large-scale datasets of labelled RGB images are widely available, this is particularly challenging when taking a sensor fusion approach, since it requires data from multiple modalities capturing the same scene, at the same time, and from a similar perspective. In this chapter, we review and compare existing vision datasets with potential applications to multi-modal scene understanding. We then select the most relevant datasets for this task, and outline the pre-processing steps for training our prediction models.

5.1 Relevant datasets

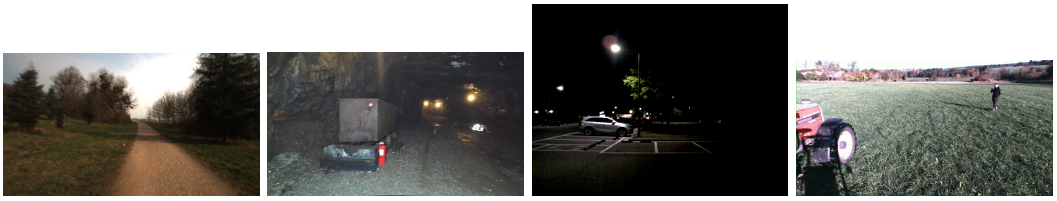
When surveying existing literature for publicly available datasets, we specifically consider those which feature a combination of RGB, depth, and/or IR data, and capture outdoor scenes captured from an egocentric perspective. Aerial or surveillance datasets are excluded, for instance. A summary of these datasets is given in the appendix (Tables A.1 for RGB-D-IR, A.2 for RGB-IR and A.3 for RGB-D), with further comments for each category below.

5.1.1 RGB-D-IR

To the best of our knowledge, **Freiburg Forest** [110] is currently the only dataset capturing outdoor scenes with our three modalities of interest (RGB, depth, and IR), while providing full-image pixel-level annotation. Time-synchronized and calibrated multi-modal image pairs are provided out of the box, allowing for rapid prototyping and experimentation. However the dataset contains less than 300 frames capturing the same forested area, and does not feature any dynamic obstacles. Furthermore, the depth maps in this dataset were generated by a depth

estimation DCNN from monocular images, since the authors found the disparity images from the stereo RGB camera to be too noisy, giving poor segmentation results in unstructured environments. **PST900** [96] is the only other dataset in this category providing pixel-level annotations. It also provides aligned multi-modal pairs with long-range stereo depth maps. However, it is limited to a few specific object classes, and is more suited for search-and-rescue applications than outdoor scene understanding, since it was captured in an underground setting.

Among the other RGB-D-IR datasets in our overview, **ROB7** [46] was collected as part of a previous semester project and contains valuable tri-modal data featuring heavy pedestrian activity and un-marked lanes. However, the thermal captures are un-calibrated and not precisely time-synchronized with RGB-D captures, resulting in significant misalignment when the vehicle is moving at considerable speed. **ViViD** [62] provides full multi-sensor calibration information, however the data is a raw stream recorded at a high frame-rate along a very short trajectory, with an unequal number of readings per sensor. Extracting and annotating multi-modal pairs would require synchronization. Similarly, **FieldSAFE** [57] provides raw data recorded in a single location, albeit a much more unique one. Both datasets suffer from low quality depth maps, making them poorly suited for learning to outline obstacles. **Brno Urban** [65] is an impressive candidate due to its scale, environmental diversity, number of sensors used to capture driving scenes, and particular care paid to time-synchronization, however calibration parameters are approximate, and multi-modal pairs have to be extracted manually from raw streams. We include **CATS** [106] mostly because of its name, but also because it contains interesting hybrid scenes and stereo RGB/thermal pairs - but unfortunately, no annotations to go along with it either.



(a) Freiburg Forest **(b)** PST900 **(c)** ViViD **(d)** FieldSAFE

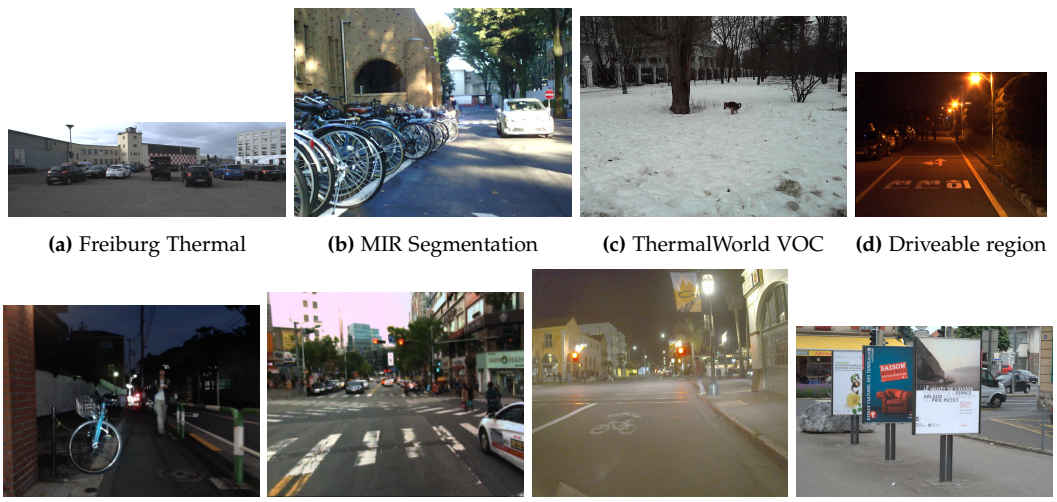
(e) ROB7 **(f)** CATS **(g)** Brno Urban

Figure 5.1: RGB sample for each RGB-D-IR dataset in Table A.1

5.1.2 RGB-IR

Pixel annotated multispectral datasets remain scarce, even when removing the requirement for depth data. To the best of our knowledge, **Freiburg Thermal** [113] (released very recently) is currently the only large-scale RGB-thermal dataset with full-image semantic labels, providing valuable time-synchronized and aligned multispectral data in challenging city scenes. However, its pixel labels were produced by an RGB segmentation teacher network rather than a human annotator, and are therefore often approximate, and affected by lighting artefacts. **MIR Semantic Segmentation** [38] contains images captured in 3 different infrared bands, along with more accurate semantic masks, but only for a small set of dynamic obstacles, thus leaving most of the pixels unlabelled. **ThermalWorld VOC** [55] is more of a general purpose dataset, captured with a hand-held set-up from an unconstrained viewpoint in diverse pedestrian locations, but is also only partially pixel-annotated.

KAIST pedestrian [47] is specifically geared towards all-day pedestrian detection (surprise) in urban scenes, thus only providing bounding box annotations. Similarly, **FLIR ADAS** provides bounding box annotations for 5 common dynamic obstacles, and includes night-time captures; however, this dataset only includes RGB images for reference - they are not spatially aligned with their thermal counterpart, and no calibration information is provided. **Driveable region** [116] has a similar limitation: night-time thermal images are pixel-annotated, but corresponding RGB frames are only provided for a small fraction of the dataset, and are not spatio-temporally aligned. Lastly, **RGB-NIR Scene** [13] contains non-annotated multispectral pairs captured in a wide range of settings from a static viewpoint; however, since the two modalities are captured sequentially, we note significant discrepancies in scene content in some image pairs eg. due to people walking in the background.



(a) Freiburg Thermal **(b)** MIR Segmentation **(c)** ThermalWorld VOC **(d)** Driveable region

(e) MIR Object Detection **(f)** KAIST Pedestrian **(g)** FLIR ADAS **(h)** NIR Scene

Figure 5.2: RGB sample for each RGB-IR dataset in Table A.2

5.1.3 RGB-D

Among the annotated datasets in our overview, **Cityscapes** [21] is perhaps the most common benchmark for outdoor semantic scene understanding, as it provides fine annotations for thousands of images. It also has the advantage of featuring real disparity computed via semiglobal matching [44] from stereo pairs. However, it was captured in ideal outdoor conditions and features highly structured scenes. **Lost and Found** [85] was captured in similar conditions as Cityscapes, along different German streets in sunny weather. However, it was specifically developed to assess small obstacle segmentation on roads: the images feature objects such as toys or boxes along the vehicles path. Only coarse annotations of the road area and the obstacle are provided. **Kitti** [30] is also a well-known urban driving benchmark and features more challenging illumination conditions than Cityscapes, but its semantic segmentation dataset is only limited to 200 samples. Depth information is provided as sparse 2D projections of LIDAR scans. **SYNTHIA video** [89] provides pixel-wise object classes and dense "true" depth at a much larger scale, since it was generated in a simulation environment.

The 3 remaining datasets in this category do not include semantic annotations. **RADIATE** [94] stands out due its particularly challenging weather conditions, including dense fog, snowfall and heavy rain with water coating the camera view; it provides stereo RGB pairs, radar and 3D LIDAR scans but no temporally aligned multi-modal pairs. In contrast to widespread urban driving datasets which are often limited to highly predictable road scenes from European cities, **IDD Multi-modal** [111] captures unstructured traffic in India, providing LIDAR scans in addition to RGB images - but no temporally aligned pairs or calibration information. Lastly, **DIML / CVLAB** [17] is an RGB-D dataset designed for depth estimation in highly diverse scenes - unlike the others in this category, images are taken from a pedestrian perspective, thus extending beyond vehicle-centric street views.

(a) Cityscapes **(b)** Kitti **(c)** Lost and Found **(d)** SYNTHIA video

(e) RADIATE **(f)** IDD Multi-modal **(g)** DIML/CVLAB

Figure 5.3: RGB sample from each RGB-D dataset in Table A.3

5.1.4 Honorable mentions

The following datasets were not included in our overview, either because they do not feature multi-modal data, or because they escaped our attention at the time of writing, but still bear mentioning for future reference:

- **Robot Unstructured Ground Driving (RUGD)** [115] was recorded by a mobile robot in off-road environments, and includes 7k+ RGB images with dense pixel annotations for 24 semantic categories.
- **Dense Indoor and Outdoor DEpth (DIODE)** [112] is a large-scale (20k+) RGB-D dataset captured in diverse indoor and outdoor scenes, with high-quality and long-range depth measurements.
- **RGB and NIR Urban Scene Dataset for Deep Scene Parsing (RANUS)** [18] includes 4k pixel-annotated RGB-NIR pairs of driving scenes for 10 semantic categories.
- **WildDash** [119] (4k+ samples) and **Mapillary Vistas** [80] (25k samples) are two interesting pixel-annotated RGB semantic segmentation datasets for urban driving in challenging conditions, spanning diverse locations, weathers, and sensor characteristics.

5.2 Chosen datasets

Considering the time and effort needed to manually annotate a full image segmentation dataset for training, we specifically select datasets featuring pixel-level annotations of outdoor scenes. In the same spirit of laziness, we also exclude raw datasets which do not provide registered and calibrated multi-modal samples, or pre-computed depth maps. For tri-modal fusion, this leaves us with **Freiburg Forest** as the only candidate. For bi-modal fusion, we note that all the RGB-IR datasets meeting our criteria are either only partially annotated (**ThermalWorld VOC**, **MIR Multispectral Segmentation**), or approximately annotated (**Freiburg Thermal**). Thus, we turn to RGB-D datasets for more options - however, **Lost and Found** is only partially annotated, **SYNTHIA** is a synthetic dataset and thus would not be indicative of real-world performance, and **Kitti**'s segmentation dataset is too small for training.

Therefore, we take **Freiburg Forest** and **Cityscapes** as our primary datasets for development and experimentation - this gives us two completely different types of scenes and a total of 4 modalities to experiment with: visible spectrum, NIR, dense pseudo depth, and real stereo depth, coupled with fine full-image pixel annotations. We use the rest of the aforementioned datasets to complement some of our experiments, and to include thermal imaging in the final stage.

5.2.1 Dataset splits

Table 5.1 details the training/validation/test split used for each of the chosen datasets. The training set is used for model estimation, while the validation set is used for monitoring performance on unseen samples during learning and selecting the best model for evaluation. Evaluation is then performed on the test set. When available, we use the official split. If the dataset is explicitly divided into different sequences or locations, we try to create a challenging split such that separate locations/sequences are used in each set. For datasets which include night-time captures, we only use day-time ones, since we have no fully annotated night-time dataset available for training. We also exclude indoor scenes from ThermalWorld VOC. Note that for small datasets (< 500 samples) like Freiburg Forest and Kitti, we use the same set for validation and testing, to preserve a maximum amount of training data. Instructions for downloading each of these datasets and reproducing our data splits are included in our project repository[1].

Dataset, *sub-set*	**Split**	**Samples**			
		total	train	val	test
Kitti [30]	random 90/10% split of original train set	**200**	180	← 20 →	
Freiburg Forest [110]	official split	**336**	230	← 136 →	
MIR Multispectral [38] *daytime*	official split	**820**	410	205	205
ThermalWorld VOC *outdoor* [55]	random 80/10/10% split of original training set	**1466**	1173	147	146
Lost & Found [85]	original test set + per-street split of original train set	**2239**	949	87	1203
Cityscapes [21]	original train set + per-city split of original val set	**3475**	2975	276	233
SYNTHIA video [89] *3 sequences*	per-sequence split	**7892**	3744	1180	2968
Freiburg Thermal [113] *daytime*	per-sequence split of original training set	**12170**	10067	988	1115

Table 5.1: Dataset splits used in the experiments, shown in order of smallest to largest dataset.

[1] `https://github.com/glhr/learning-driveability-heatmaps/tree/main/datasets`

5.3 Data preparation

5.3.1 Depth maps

The **Freiburg** and **Synthia** datasets already include artificial dense depth maps with valid values for every pixel, and thus can be directly used as input to a deep image segmentation network. However, depth maps from other datasets of interest which provide real range-sensing data such as **Cityscapes** and **Kitti** are noisy and contain missing values. These can be approximated with depth completion as a pre-processing step, as introduced in Section 2.1.3. We investigate the effect of depth completion on segmentation performance in Section 7.2 (single-modality model) and Chapter 8 (multi-modal models).

Depth completion

We employ the method in [59] to perform basic un-guided depth completion on sparse or discontinuous depth maps. While this method was originally demonstrated on LIDAR scan depth maps from the Kitti dataset, [109] successfully applies it to disparity images from the Cityscapes dataset as a pre-processing step for semantic segmentation.

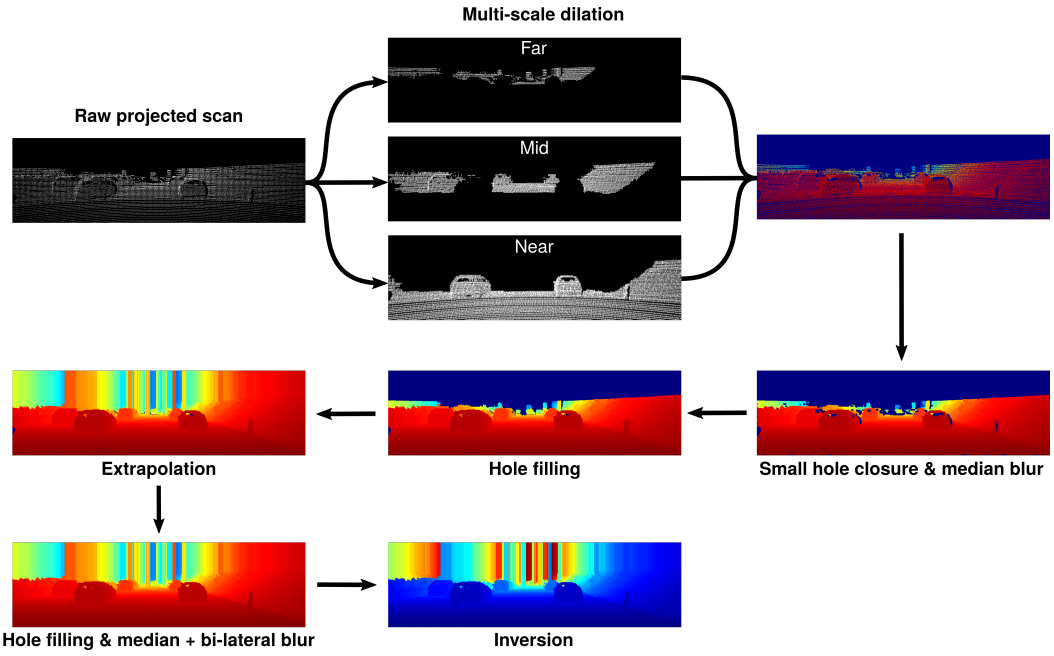


Figure 5.4: Steps in the multi-scale depth completion pipeline proposed in [59], applied to a sparse depth map from the Kitti dataset. The input laser scan is contrast-enhanced and subsequent depth maps are colorized for visualization.

As illustrated in Figure 5.4, this depth completion method applies a series of standard image filtering and morphological operations for filling missing depth values and reducing noise. The official *IP Basic* implementation[2] provides several variants of the method, with different quality vs. speed trade-offs. Since we perform all pre-processing offline with no runtime constraints, we opt for the highest-quality variant, which performs dilation at 3 different scales, distinguishing between close-range, mid-range and far-range pixels, with larger dilation kernels for closer objects. To perform depth completion on stereo depth maps from different datasets, we adapt the original implementation by:

- adjusting the thresholds delimiting the three depth zones (near, mid, and far) for each dataset based on its input type and range, and defining a stereo region of interest (ROI), in which missing values should be estimated. These regions are used to selectively apply filtering to different parts of the depth map using pixel masks, which we illustrate with an example in Figure 5.5.
- performing extrapolation not only towards the top edge of the image, but also towards the other edges if necessary.
- only performing extrapolation for pixels outside of the stereo ROI.
- adding a final hole-filling step with a large dilation kernel size in order to over-write remaining missing values.

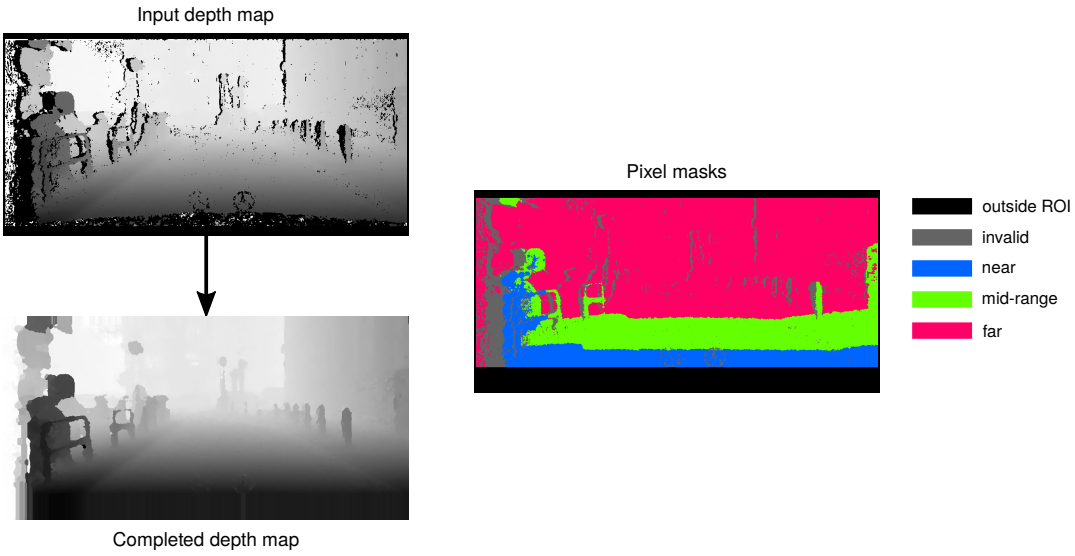
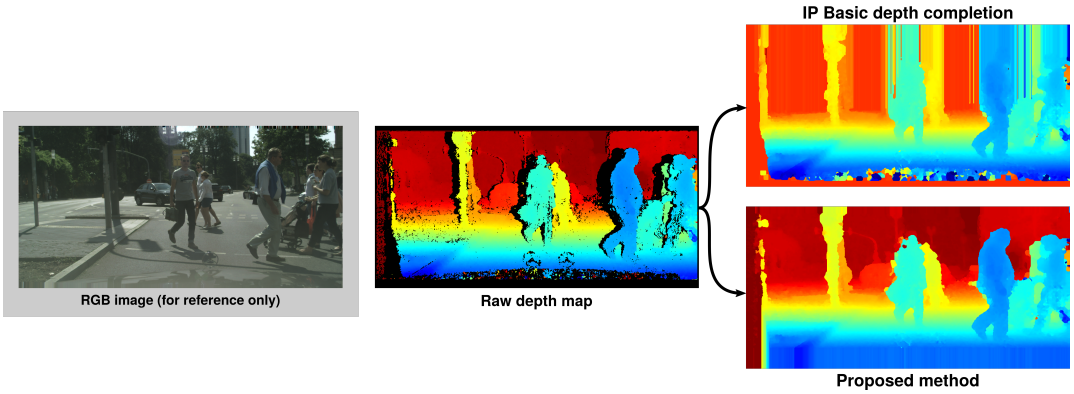


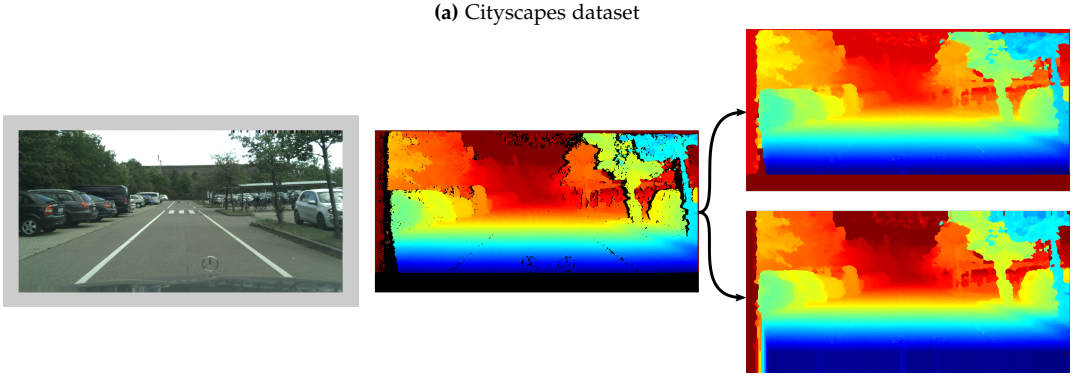
Figure 5.5: Pixel masks used in the depth completion process, illustrated with a sample from the Cityscapes dataset.

[2] `https://github.com/kujason/ip_basic`

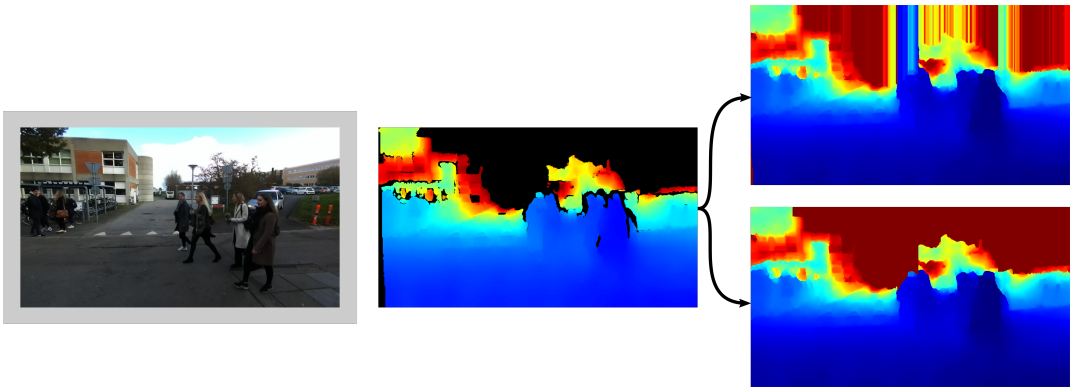
The steps of the full depth completion process are detailed in Section A.3, and Figure 5.6 shows some examples of completed stereo depth maps compared to the original IP Basic implementation.



(a) Cityscapes dataset



(b) Lost and Found dataset



(c) ROB7 dataset

Figure 5.6: Depth completion results on samples from different datasets. Depths maps are colorized for visualization, with missing values in black. We show a comparison between the original IP Basic results vs. our modified implementation.

5.3.2 Image resolution

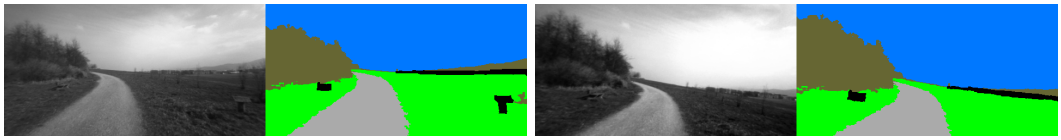
Image resolution and FOV vary widely across datasets. The size of input samples used for training CNNs is an important consideration: preserving high input resolution allows for fine segmentation without loss of detail, however this introduces large memory and computational requirements which may prohibit single-GPU training. Thus, it is common practice to resize input images to fixed dimensions [99]. For instance, the authors in [109] report segmentation performance of their architecture on the Cityscapes dataset for different input resolutions, and found that down-sampling input images from the original size of 1024×2048 to 384×768 brought a drop in accuracy of less than 1% while reducing inference time by over ten-fold. In this work, we do not consider a high level of detail at long distances to be necessary for navigation-oriented scene understanding. Therefore, we opt for a small input resolution of 240×480 - the same width as in [4], but with a wider aspect ratio (to accommodate wide-FOV urban driving datasets like Kitti). Similarly to [41, 109], images are down-sampled with bi-linear interpolation for RGB, and nearest-neighbour interpolation for other modalities and ground truth masks.

5.3.3 Augmentation

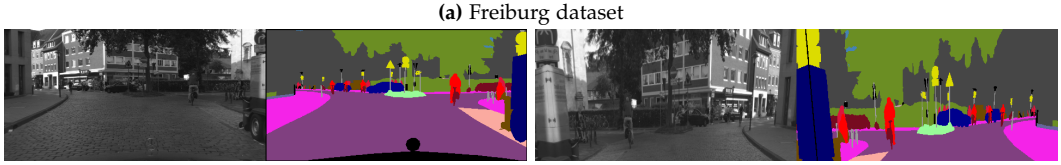
Similarly to existing scene segmentation works [79, 105, 109], we augment the training set with random geometric and photometric transformations which are listed in Table 5.2. In order to preserve label integrity, only geometric transformations are applied to the ground truth mask, with nearest neighbour interpolation. Data augmentation is applied to training samples on-the-fly with each individual transformation having a probability of 0.5. After augmentation, samples are resized to the fixed input size described in Section 5.3.2. Examples of random data augmentation applied to different samples are shown in Figure 5.7.

Geometric transformations	
horizontal flip	
random rotation	-10 to $10°$
random crop	80 to 90 % of the original size
random perspective transformation	4 points, distance of 5 to 15 % from the corners
random grid-based distortion	3×3 grid
Photometric transformations	
random brightness & contrast adjustment	-40 to 40 % of the mean intensity
random tone curve manipulation	scale of 10 %

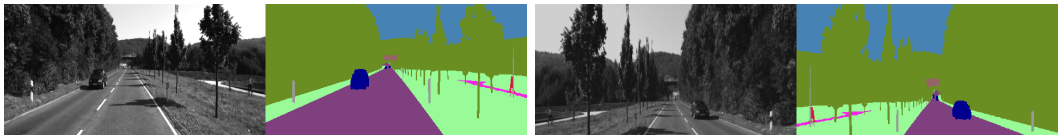
Table 5.2: List of transformations applied on training samples and their parameters. See [14] for a detailed description.



(a) Freiburg dataset



(b) Cityscapes dataset



(c) Kitti dataset

Figure 5.7: Examples of an original sample (left) and randomly augmented sample (right) for different semantic segmentation datasets

Chapter 6

Navigation-oriented scene understanding

In this chapter, we focus on segmenting single-channel visible spectrum images, as a first step towards multi-modal prediction. The goal is not to refine segmentation through architectural improvements, but rather to investigate learning schemes for navigation-oriented scene understanding. We pick SegNet, a well-established segmentation network, describe its architecture in Section 6.1 and use it for all the experiments in this chapter. We start with a standard pixel-wise categorical classification approach in Section 6.2, and investigate how to segment outdoor scenes into the proposed broad driveability classes introduced in Chapter 4, compared to learning specific object classes. In Section 6.3, we then introduce a ranking between these classes in order to learn the inherent hierarchy between driveability levels, thus turning this task into a pixel-wise ordinal classification problem. Lastly, we propose a pixel-wise loss weighting method in Section 6.4, with the aim of focusing learning on areas in the scene which are the most relevant to navigation.

6.1 Segmentation architecture

SegNet follows an encoder-decoder architecture, the encoder being based on VGG-16 [99], and the decoder mirroring the encoder via de-convolution to recover the input resolution. Except for the number of channels at the input and feature maps at the output (determined by the number of image channels and the number of training classes respectively), the network is fully symmetrical. In all blocks, convolution is applied with a kernel of size 3×3, unit stride, and zero-padding to preserve spatial resolution. Max-pooling is applied after each encoder block over a 2×2 window with a stride of 2; the max-pooling indices are transferred to the decoder blocks for up-sampling.

The network variant which we implement is illustrated in Figure 6.1. Compared

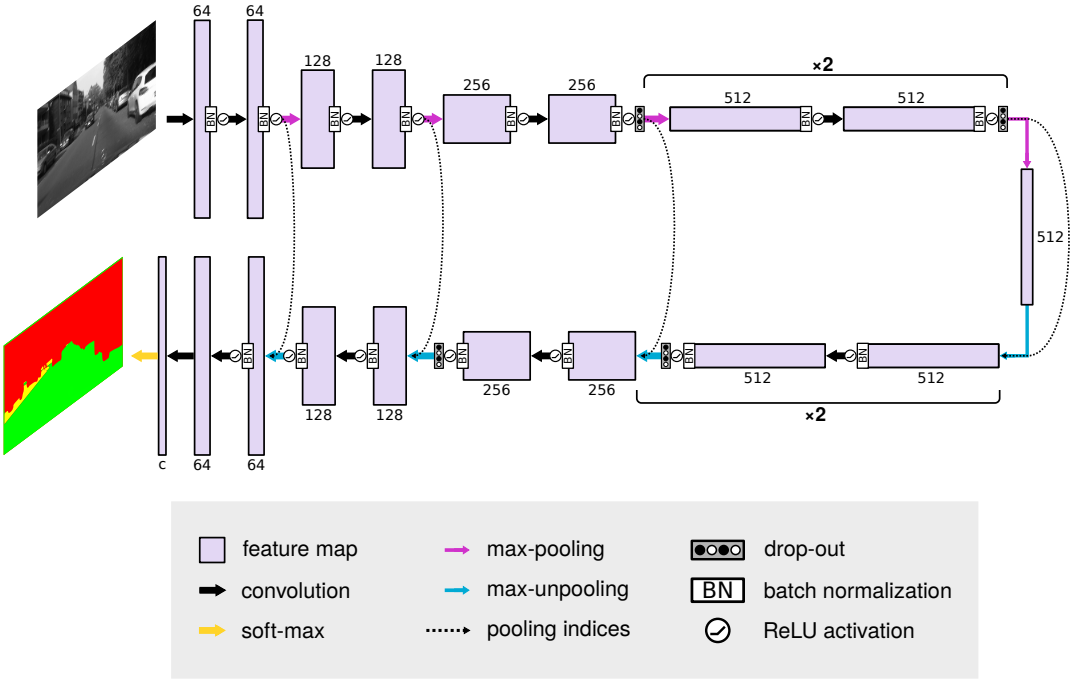


Figure 6.1: Segmentation architecture used for pixel-wise prediction, based on SegNet [4], with the encoder and decoder shown as two parallel branches. We feed the network a single-channel image, and the number of classes c at the output varies per learning scheme/experiment.

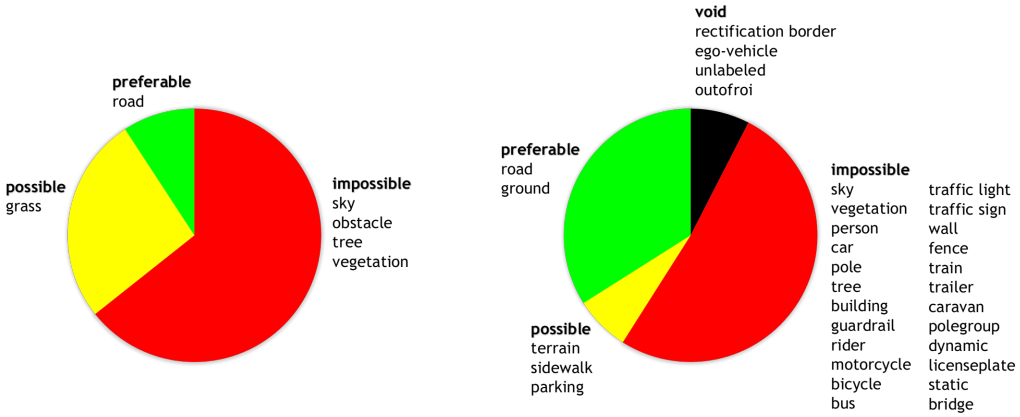
to the original SegNet model, drop-out (rate of 0.5) is applied in the six deepest encoder and decoder blocks for regularization, and the number of convolutional layers in each block is reduced to 2 (as opposed to 3 in the deepest blocks of VGG-16), results in a total of 20 convolutional layers.

6.2 Object classes to driveability

The goal is to compare our proposed class definition (3 driveability levels) to a classical object-based segmentation approach in terms of learnability, accuracy and generalization. To this end, in order to avoid having to label a full dataset from scratch, we leverage existing outdoor semantic segmentation datasets: we make use of the Freiburg Forest, Cityscapes and Kitti datasets, all three of which provide dense pixel-wise annotations across 2 or more imaging modalities, including RGB.

To generate ground truth labels, we perform a blind mapping from the original semantic classes of each dataset to driveability levels, based on the type of scene element. As detailed in Figure 6.2, essentially, any kind of object or barrier in the scene is considered *impossible* to drive on (red), along with the sky. Paved areas or paths are considered fully driveable ie. *preferable* (green), while other terrains (eg. grass) or areas on the side of the path (eg. sidewalk) are assigned to the inter-

mediary *possible* level (yellow). Figure 6.3 shows examples of resulting driveability segmentation masks obtained after applying this ground truth mapping.



(a) Freiburg Forest labels **(b)** Cityscapes labels

Figure 6.2: Pie charts showing how we sort the original semantic classes into driveability levels. The size of the slices indicates the proportion of pixels assigned to each driveability level across the full dataset.

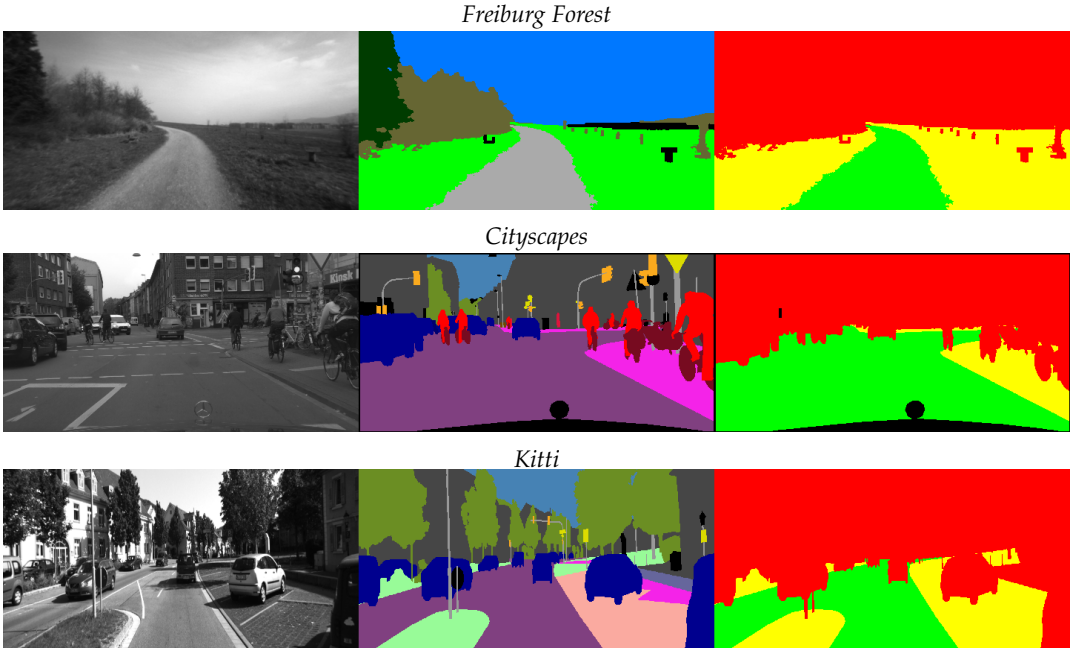
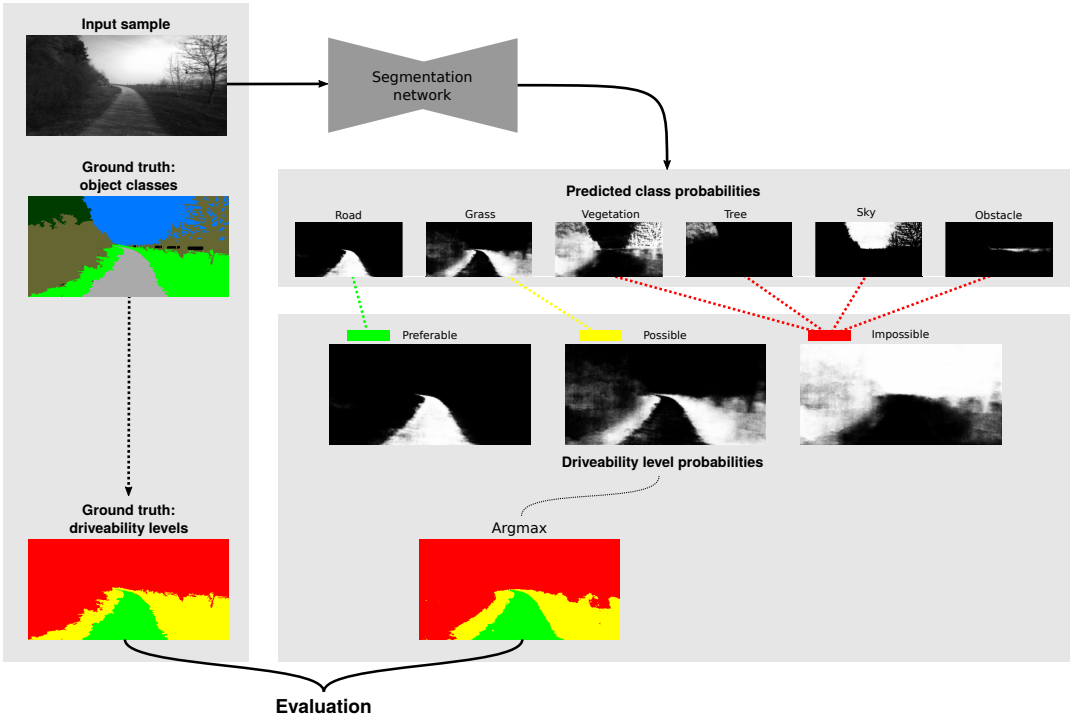
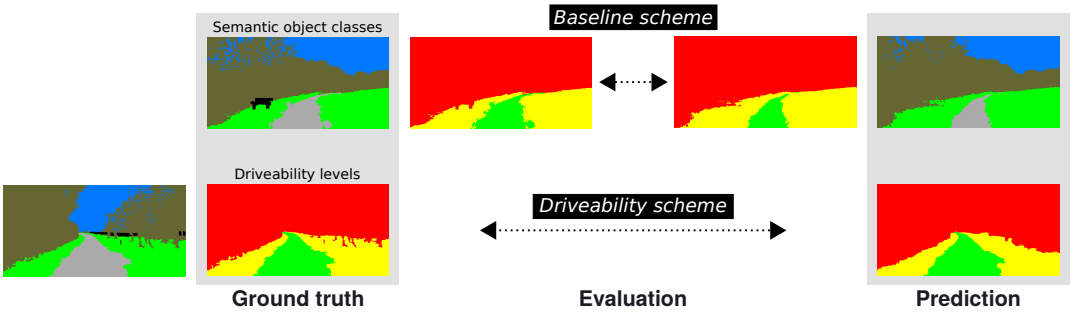


Figure 6.3: Dataset samples: single-channel visible spectrum image (left), original labels (middle), labels mapped from object classes to driveability level (right).

To assess the viability of the proposed driveability class definition compared to object-based semantic classes, we implement and evaluate three learning schemes. In the *baseline* scheme, the network is trained to predict the original object classes in the dataset (eg. tree, sky, car), and these predictions are manually mapped to driveability levels in the evaluation stage. As illustrated in Figure 6.4a, for evaluation of the baseline scheme, the predicted object class probabilities are mapped to driveability level probabilities via simple addition. The predicted driveability level is then taken as the argmax across the class probability vector for each pixel.



(a) Baseline scheme, where ground truth and predicted semantic labels are mapped to driveability levels.



(b) Baseline and driveability learning schemes - the ground truth column shows the target used when training the model, and the prediction column is the model output (after applying argmax).

Figure 6.4: Evaluation procedure, illustrated with samples from the Freiburg Forest dataset.

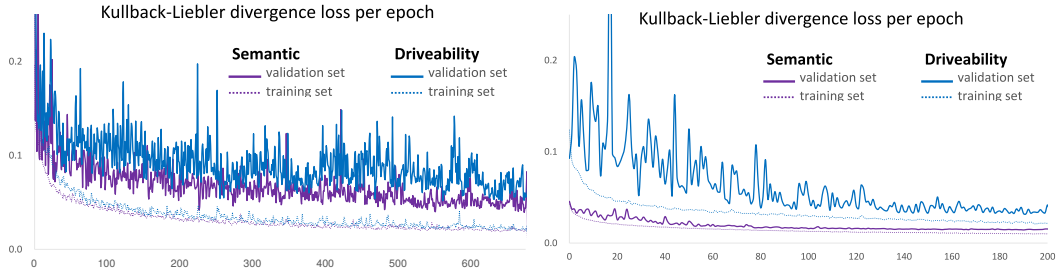
In the *driveability* scheme, the network is trained to directly predict driveability levels, as shown in Figure 6.4b. Lastly, we also implement a *transfer learning* scheme, where the network is first trained on the original object classes (outcome of the baseline scheme), and then adapted to learn driveability levels.

Training procedure

We train SegNet on grayscale visual images which are resized and randomly augmented following the procedure in Section 5.3. The model is trained on each dataset separately, with Adam optimization [54] ($\beta_1 = 0.9$, $\beta_2 = 0.999$, left as default) and a limited batch size of 8 due to memory constraints. For all three schemes, the segmentation model is trained to minimize Kullback-Liebler divergence. The loss per batch is computed as the sum of the loss per non-void pixel, divided by the number of non-void pixels in the batch - such that each labelled pixel contributes equally in the total loss regardless of the proportion of void pixels, image resolution or batch size.

For the *semantic* and *driveability* schemes, the segmentation network is trained from scratch with an initial learning rate of 10^{-3}. The best model for each scheme is selected based on minimal validation loss. In the *transfer learning* scheme, we then adapt the semantic model to predict driveability levels by re-initializing the last convolutional layer with 3 output channels and resuming training with an initial learning rate of 10^{-4} for a small number of epochs until convergence.

Figure 6.5 compares the learning curves of the *semantic* and *driveability* learning schemes on both datasets. We note lower loss, faster convergence and better generalization to the validation set for the *semantic* models - this is especially apparent when learning the Cityscapes dataset, which has a higher number of semantic classes for each driveability level (eg. $24 \rightarrow 1$ for the level vs. $4 \rightarrow 1$ for Freiburg Forest), suggesting that specific semantic descriptions are easier to learn than general driveability levels spanning different object types.



(a) Freiburg dataset **(b)** Cityscapes dataset

Figure 6.5: Learning curves for the semantic and driveability schemes trained on the Freiburg and Cityscapes datasets. The loss is scaled based on the number of training classes for comparison.

Evaluation

For each training dataset and learning scheme, we record the pixel accuracy as well as pixel-wise precision and recall per driveability level on unseen samples, both from the dataset's own test set (intra-dataset evaluation) and from the other segmentation datasets to assess domain generalization (out-of-dataset evaluation). Scores are computed by accumulating a raw confusion matrix of predictions across batches, which is column-normalized (for recall) and row-normalized (for precision) at the end of the testing epoch. Quantitative results are given in Table 6.1 and described below. We also show examples of predictions by all six models in Figure 6.7 (trained on Freiburg Forest) and Figure 6.8 (Cityscapes).

	pixel accuracy	Precision			Recall		
							
Trained on Freiburg Forest, prediction on Freiburg test set							
Semantic	94.32	95.77	89.70	**98.30**	98.27	89.47	79.57
Driveability	**94.78**	**96.40**	90.08	97.62	**98.49**	**90.70**	79.33
Transfer learning	94.30	95.23	**91.02**	97.39	98.48	87.68	**83.27**
Trained on Freiburg Forest, prediction on Kitti dataset							
Semantic	**75.65**	**88.34**	**36.41**	89.28	96.17	62.45	**28.53**
Driveability	70.86	87.51	30.88	88.74	96.26	**63.49**	6.96
Transfer learning	73.56	84.85	34.38	**91.37**	**97.41**	55.32	20.56
Trained on Freiburg Forest, prediction on Cityscapes test set							
Semantic	**68.58**	**90.14**	**15.80**	96.95	97.89	**62.08**	**25.63**
Driveability	60.20	89.10	11.95	96.72	98.38	59.03	2.73
Transfer learning	63.82	88.87	13.25	**97.04**	**98.46**	58.61	12.55
Trained on Cityscapes, prediction on Cityscapes test set							
Semantic	96.85	**99.00**	74.58	98.25	98.94	87.36	95.32
Driveability	**97.30**	98.44	**85.98**	97.39	**99.37**	78.76	**97.43**
Transfer learning	96.91	98.90	75.21	**98.41**	99.10	**87.55**	95.21
Trained on Cityscapes, prediction on Kitti dataset							
Semantic	90.73	**96.14**	83.37	80.55	97.96	53.36	94.35
Driveability	87.15	93.84	**85.96**	72.89	**98.48**	23.83	**95.72**
Transfer learning	**91.49**	95.31	85.04	**84.36**	98.41	**57.22**	94.00
Trained on Cityscapes, prediction on Freiburg Forest dataset							
Semantic	**69.53**	89.97	55.08	27.45	92.53	**4.37**	94.54
Driveability	63.72	**92.59**	**71.49**	21.94	84.76	0.20	**97.90**
Transfer learning	69.46	87.78	58.97	**28.60**	**92.75**	3.73	94.16

Table 6.1: Quantitative results of argmax predictions, comparing the three learning schemes. For readability, we highlight the **best** and worst result for each metric.

We first remark that the results of this experiment are affected by class imbalance: the most infrequent driveability levels (for Freiburg, for Cityscapes) have the lowest recall scores, and although we evaluate the models under the same driveability level definition, note that they were trained under different class definitions and distributions (eg. in Cityscapes, *pole* is a minority class when training the *semantic* model, but falls under the majority class when training the *driveability* model). Thus, in the evaluation, it may be difficult to disentangle the effects of class distribution from differences in learning scheme.

Looking at intra-dataset performance, the models achieve higher pixel accuracy on Cityscapes than Freiburg Forest. We attribute this in part to the dataset's scale ($\approx$ 10 times more data to learn from than Freiburg Forest), and to the fact that its well-lit urban scenes are less ambiguous to segment than the unclear transitions between path, grass, and surrounding vegetation in Freiburg Forest, especially when solely relying on grayscale information. Comparing the three learning schemes, the *driveability* model produces the most accurate segmentation, out-performing the *semantic* and *transfer learning* models by approximately 0.5% on both datasets. This is somewhat expected, since it was trained from scratch to specialize in learning to segment driveability.

The differences in performance between the learning schemes widen significantly for out-of-dataset prediction. Note that this is a particularly challenging task, since the out-of-dataset samples include completely different scenes to those that the models were trained on (urban scenes vs. forested areas): models trained on Cityscapes have seen little to no grass, off-road paths or rocks, while those trained on Freiburg Forest have not seen any paved road, building or dynamic obstacle. We aggregate and visualize the differences in intra-dataset and out-of-dataset performance between the learning schemes in Figure 6.6. Here we pay particular attention to precision scores for the most driveable areas and recall

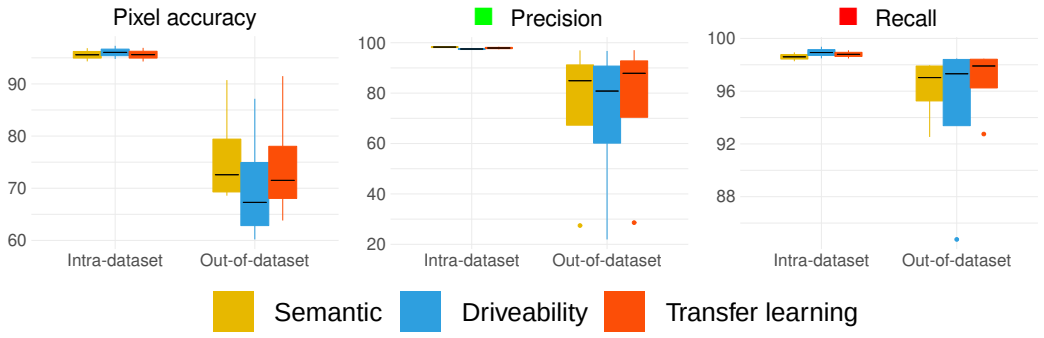


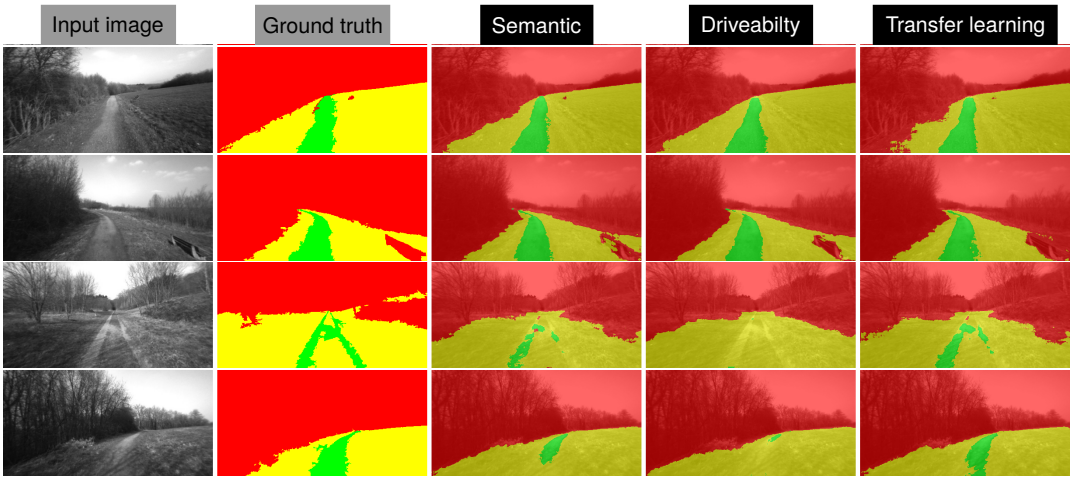
Figure 6.6: Distribution of intra-dataset and out-of-dataset performance metrics across models trained on both datasets (Freiburg Forest and Cityscapes), comparing the three learning schemes. The scores are directly taken from Table 6.1.

scores for undriveable areas, based on the idea that over-segmentation of the driveable path and under-segmentation of obstacles raises safety concerns for navigation: it is better to be conservative about where we should drive, and overly cautious about what to avoid.

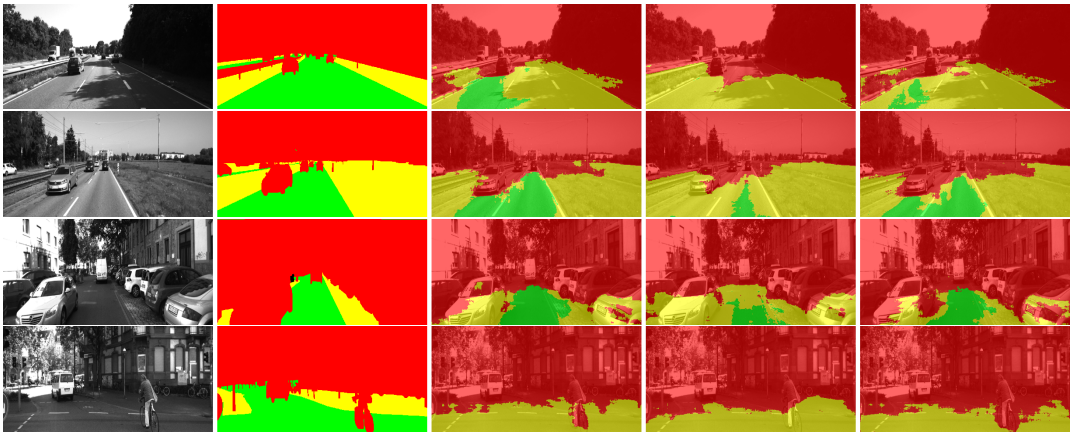
On out-of-dataset samples, models trained under the *transfer learning* scheme jointly achieve higher recall for areas and higher precision for areas than with the two other schemes. The *semantic* models have the highest tendency to under-segment obstacles, while the *driveability* models are the most likely to incorrectly label areas as preferable to drive on. We especially note a large drop in the performance of the *driveability* model when trained on Cityscapes and evaluated on Freiburg Forest.

The pictures Looking at the side-by-side qualitative comparison in Figure 6.7 and Figure 6.8, we observe that the driveability model tends to categorize large areas as the same thing, with less attention to detail, while the *semantic* model is more sensitive to fine-grained patterns and small obstacles. We see this in the first row of Figure 6.7 for instance, where unlike the *semantic* model, the *driveability* model fails to segment the bench in the distance, but produces smoother segmentation outlines. In the third and fourth row, the *driveability* model fails to recognize most of the thin path altogether. The *transfer learning* seems to inherit properties from both of these learning schemes, often outputting what looks like a reasonable middle ground, and more consistent segmentation of obstacles both for intra- and out-of-dataset samples. We also notice that the models tend to mimic the scene geometry and class distribution which they were originally trained on: when trained on Cityscapes for instance, out-of-dataset predictions maintain a low proportion of pixels, and a wide path spanning most of the image, while the opposite holds when trained on Freiburg Forest, where the path constitutes a much smaller fraction of images.

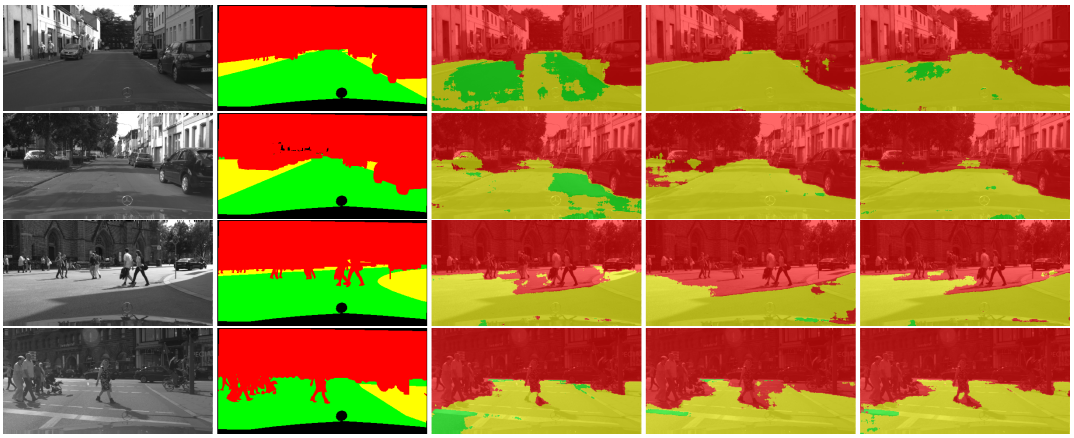
Wrapping up It appears that in being trained on a general class definition from the start, the *driveability* models learn a broad-stroked scene representation which generalizes poorly in completely new scenes and often overlooks relevant details. For generalization to new scenes and obstacles, it seems beneficial to first learn narrow descriptive classes and then adapt the model via transfer learning to learn a more general, functional description. The transfer learning scheme also benefits from fast training compared to learning the proposed driveability definition from scratch, converging in under 40 epochs on Freiburg Forest and in under 20 epochs on Cityscapes. Thus, we opt for this learning scheme in all the following experiments.



(a) Trained on Freiburg Forest, prediction on Freiburg Forest test set

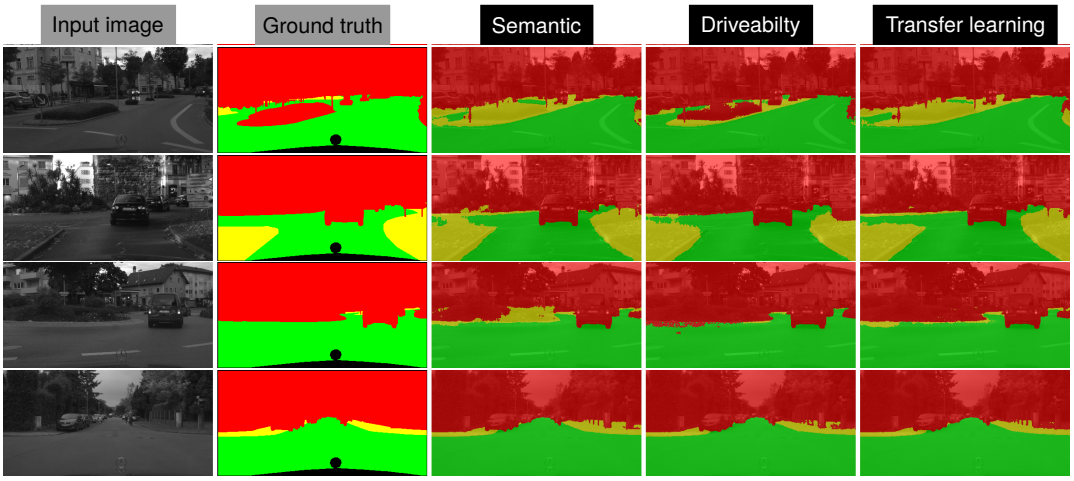


(b) Trained on Freiburg Forest, prediction on Kitti

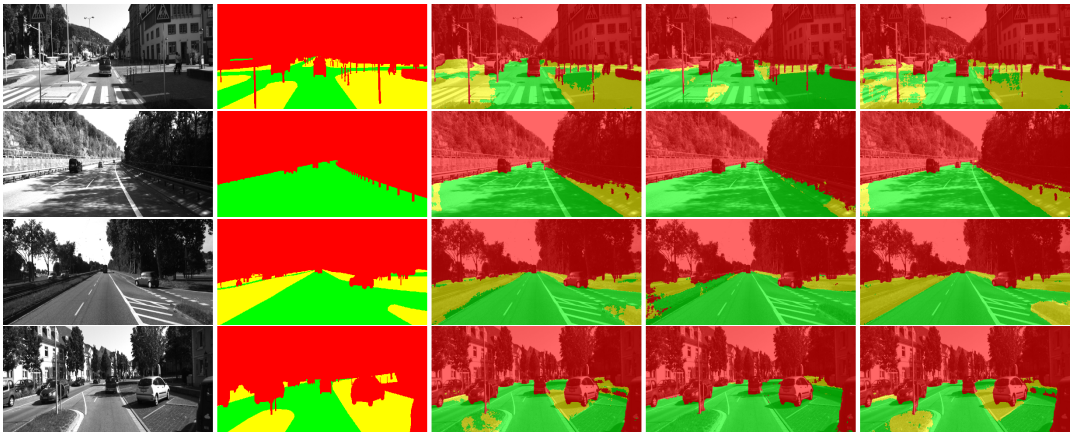


(c) Trained on Freiburg Forest, prediction on Cityscapes

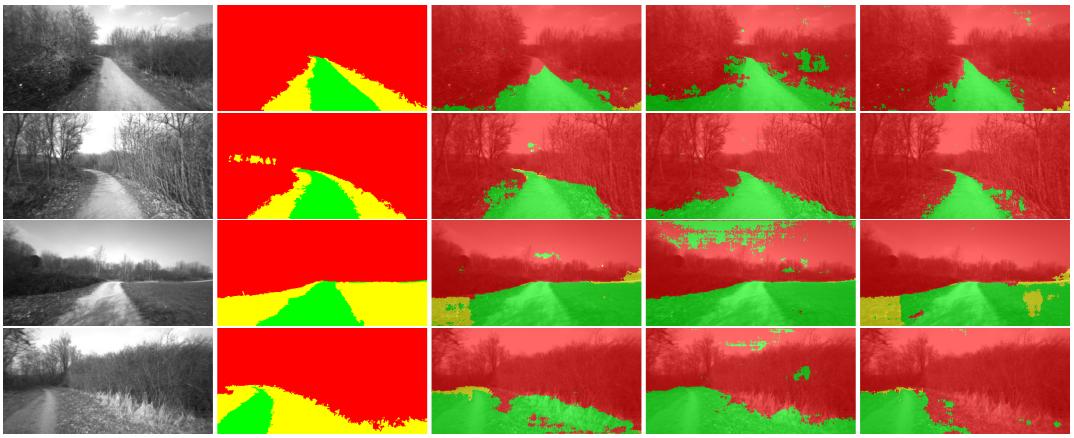
Figure 6.7: Selected samples from the Freiburg Forest, Cityscapes and Kitti datasets, comparing the segmentation output of the Freiburg Forest models in Table 6.1, under the three learning schemes. Predictions are shown overlayed on the input image.



(a) Trained on Cityscapes, prediction on Cityscapes test set



(b) Trained on Cityscapes, prediction on Kitti



(c) Trained on Cityscapes, prediction on Freiburg Forest

Figure 6.8: Selected samples from the Freiburg Forest, Cityscapes and Kitti datasets, comparing the segmentation output of the Cityscapes models in Table 6.1, under the three learning schemes. Predictions are shown overlayed on the input image.

6.3 Soft labels for ordinal segmentation

In Section 6.2, we have treated driveability level prediction as a standard categorical segmentation task, without considering a hierarchy between the three classes, and thus treating all mis-classifications as equally severe. In this section, we incorporate a ranking between the classes during learning such that the further a prediction is from the target class, the higher the loss: for instance, classifying an area which is *preferable* to drive on as *impossible* to drive on should be penalized more heavily than classifying it as *possible*. Thus, rather than only considering how many mistakes the network makes, the goal is to encourage it to make "better mistakes", as expressed in [9]. As demonstrated in [28] and [23] for instance, this can be accomplished by training the network on soft labels which encode inter-class relationships. As opposed to other ordinal classification methods which involve architectural changes or modifications of the loss formulation, a soft labelling approach only requires a modification of the ground truth data, and can be used in a standard classification pipeline, allowing for direct comparison with the default hard labelling scheme.

We first describe how soft ground truth labels are generated, and how we define the inter-class relationships for this task. We then train our segmentation network on these ordinal labels and evaluate its performance in terms of accuracy and mistake severity, compared to the previous experiment.

6.3.1 Generating soft labels

To encode a distance between the driveability levels, we implement the Soft Ordinal vectors (or SORD) labelling scheme proposed in [23], where "hard" one-hot encoded labels are converted to a softmax-normalized probability distribution based on a ranking definition, such that the target class has the highest probability and the other probabilities encode a distance from the target class. Given a set of ranks $R = \{r_{impossible}, r_{possible}, r_{preferable}\}$ (one per driveability level), a SORD ground truth label $\hat{\mathbf{y}}$ can be generated based on a target rank r_t as follows:

$$\hat{\mathbf{y}}_i = \frac{\exp -\phi(r_t, r_i)}{\sum_{k \in R} \exp -\phi(r_t, r_k)} \quad \forall r_i \in R$$

where $\phi(r_t, r_i)$ is a metric loss function which penalizes deviation from the target rank r_t eg. absolute difference $|r_t - r_i|$. As all the distances between the ranks approach infinity, $\hat{\mathbf{y}}$ reduces to a one-hot encoded vector; as the distances approach 0, $\hat{\mathbf{y}}$ approaches a uniform probability distribution.

6.3.2 Ranking definition

In common applications of ordinal classification, the ranking R is already defined based on natural order in the data: for instance, in age estimation, the distance between the classes is simply a difference in years, or in depth estimation, the ranking is based on metric measurements. In our case, the ranking definition is somewhat arbitrary, since we do not have a quantitative measure of driveability: we simply know that $r_{impossible} < r_{possible} < r_{preferable}$.

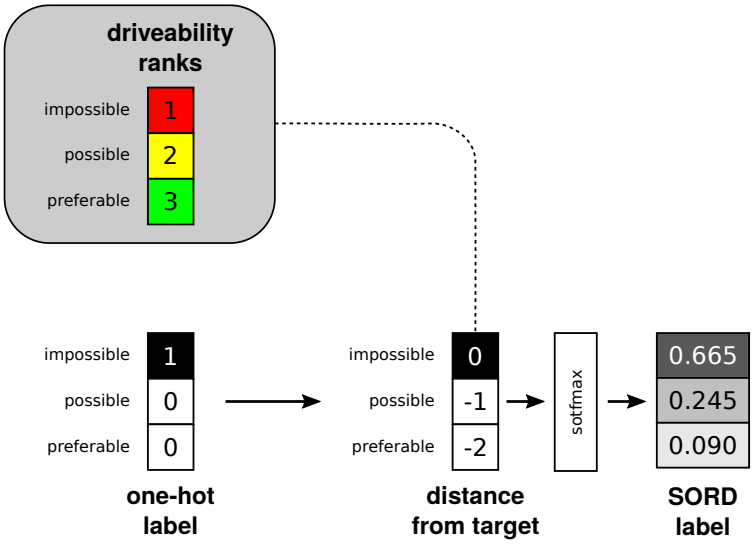


Figure 6.9: Diagram showing how a soft ordinal label is generated from a hard label (for a single pixel) based on a class ranking definition, using absolute difference as metric loss function $\phi(r_t, r_i)$.

We first consider a simple ranking definition, where the classes {impossible, possible, preferable} are mapped to the ranks $R = \{1, 2, 3\}$ (least to most driveable); using absolute difference as our metric loss function ϕ, the inter-class distance in this case is 1. Figure 6.9 shows how a soft label can then be generated based on this definition for a single sample - in this example, the target is *impossible*. A vector encoding the distance of each class from the target class is then computed: the *possible* class has a distance of 2 from the target, for instance. We then simply apply softmax on the distance vector in order to obtain a normalized class probability distribution, to be used as the soft label.

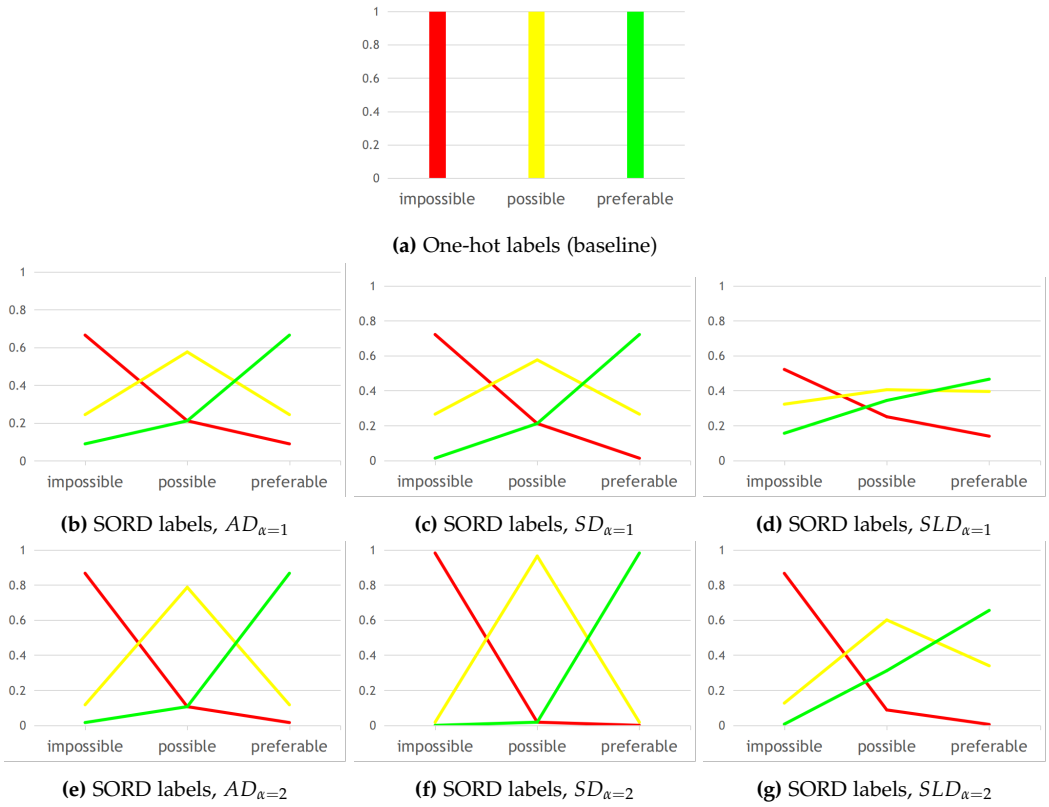
Similarly to [23], in addition to absolute difference (AD), we consider other inter-class distance metrics for generating soft ordinal labels, including square difference (SD), and square log difference (SLD):

$$\phi(r_t, r_i) = \alpha |r_t - r_i| \quad \text{(AD)}$$

$$\phi(r_t, r_i) = (\alpha |r_t - r_i|)^2 \quad \text{(SD)}$$

$$\phi(r_t, r_i) = (\alpha |\log(r_t) - \log(r_i)|)^2 \quad \text{(SLD)}$$

We introduce a factor α which scales the label entropy - as α grows, the SORD label approaches a one-hot label with 0 entropy. For AD and SD, setting α to 2 with the ranks $R = \{1, 2, 3\}$ has the same effect as using $R = \{2, 4, 6\}$ with $\alpha = 1$. Figure 6.10 shows SORD labels generated for the three inter-class distance metrics and two α values which we consider in the experiment. While SD and AD yield symmetric and scale-invariant encodings, SLD yields softer labels with increasing rank.



(a) One-hot labels (baseline)

(b) SORD labels, $AD_{\alpha=1}$ **(c)** SORD labels, $SD_{\alpha=1}$ **(d)** SORD labels, $SLD_{\alpha=1}$

(e) SORD labels, $AD_{\alpha=2}$ **(f)** SORD labels, $SD_{\alpha=2}$ **(g)** SORD labels, $SLD_{\alpha=2}$

Figure 6.10: Label class probabilities generated for different label representations. For the SORD labels, we use the following rank definition: $R = \{1, 2, 3\}$

6.3.3 Training procedure

Loss Aligning with all the experiments in this section and following [28, 23], we compute the loss as the Kullback-Leibler divergence (cf. Section 2.2.1). Figure 6.11 illustrates how the loss value varies for different possible mistakes in each labelling scheme. In the standard one-hot labelling case, all incorrect predictions yield the same loss value; in the SORD labelling scheme, the loss value increases with increasing distance from the target class. Note that when using square log difference (SLD) as the inter-class distance metric, the confusion matrix is not vertically or

horizontally symmetrical: for instance, classifying an obstacle as driveable yields a higher loss than classifying a driveable pixel as undriveable - the network has a higher incentive to assign pixels a low driveability level.

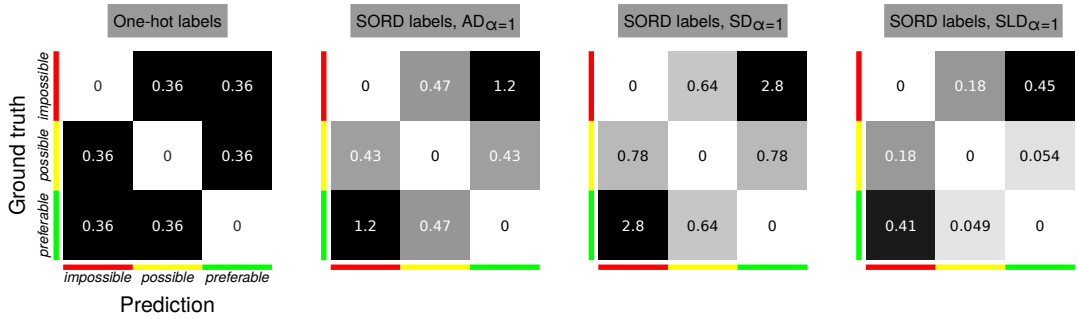


Figure 6.11: Confusion matrix of the Kullback-Leibler divergence between all possible combinations of ground truth vs. predicted class, under different labelling schemes (with $R = \{1, 2, 3\}$). For conciseness, we only show the $\alpha = 1$ variants, since increasing α only hardens the labels.

Learning As a baseline for evaluation, we use the transfer learning model from Section 6.2 (which was trained on standard one-hot encoded labels). For training the network on soft labels to learn driveability, we use the same procedure and hyper-parameters: the network is initialized with the baseline object class model, and then adapted to learn 3 driveability levels by re-initializing the last layer and continuing training with a lower learning rate.

6.3.4 Evaluation

Similarly to [23], we record categorical pixel accuracy as well as regression metrics (cf. Section 2.2.4): mean absolute error (MAE), root mean squared error (RMSE), root mean squared log error (RMSLE), mirroring the three inter-class ϕ metrics (AD, SD, and SLD) which we compare for the SORD vector computation. We also introduce a *mistake severity* metric, which is based on [9], and is computed as the mean absolute error of incorrect predictions. The metric is normalized based on the minimum and maximum possible error for a single prediction (1 and 2 respectively with a ranking of $R = \{1, 2, 3\}$), such that a severity of 0 corresponds to the lowest possible error, and 1 corresponds to the worst possible mistake. Note that while the mean error metrics implicitly depend on the proportion of correct vs. incorrect predictions, the mistake severity metric only considers the error for incorrect predictions, and is thus fully decoupled from accuracy.

Table 6.2 compares the performance of the models trained on SORD labels compared to the one-hot baseline. Looking at the best performing models on each dataset, we first note significantly different results for Freiburg Forest vs. Cityscapes. An important difference between the datasets is the proportion of

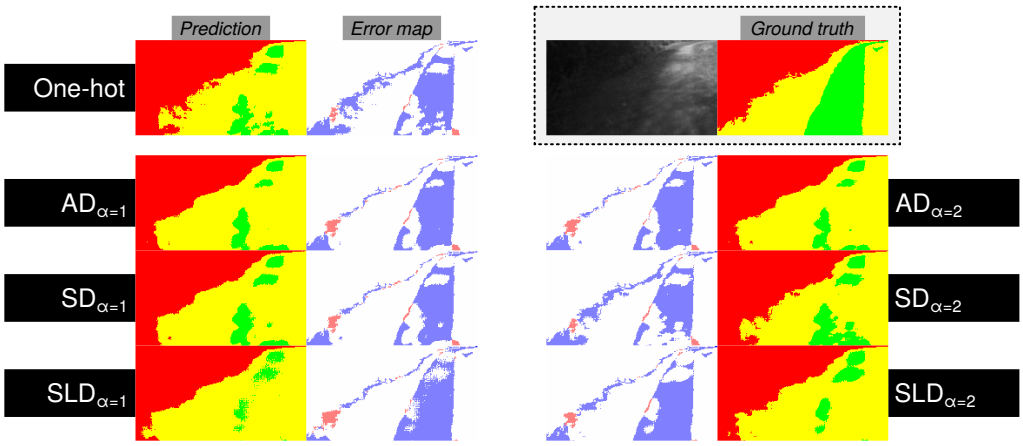
pixels (cf. Figure 6.2) and general location of obstacles: in Freiburg Forest, pixels are much more prevalent due to the abundant grassy areas and obstacles on the driveable path are rare, while in Cityscapes images, the proportion of sidewalk or other terrain is very slim, and obstacles on the road are a common occurence. This transpires through the mistake severity metric: for Cityscapes samples, the error in driveability level is much more likely to be 2 than 1 compared to Freiburg Forest, hence the higher mistake severity. For Freiburg Forest, all SORD label methods yield a significantly lower mistake severity and higher IoU for areas which are impossible to drive on than the standard one-hot labelled baseline. On Cityscapes, the performance of different SORD models is a trade-off between mistake severity and accuracy. For both datasets, $SLD_{\alpha=1}$ makes the least severe mistakes.

	pixel accuracy	IoU			error			mistake severity
					MAE	RMSE	RMSLE	
Trained on Freiburg Forest, prediction on Freiburg test set								
One-hot	94.30	93.85	80.70	***81.45***	0.0580	0.2448	0.1527	0.0173
$AD_{\alpha=1}$	94.39	94.13	81.16	80.27	0.0568	0.2412	0.1491	0.0122
$SD_{\alpha=1}$	**94.53**	***94.21***	**81.54**	81.34	**0.0553**	**0.2374**	**0.1474**	**0.0010**
$SLD_{\alpha=1}$	94.25	**94.26**	80.79	77.67	0.0581	0.2435	0.1487	**0.0010**
$AD_{\alpha=2}$	***94.44***	94.18	***81.36***	80.39	***0.0563***	***0.2401***	***0.1484***	0.0124
$SD_{\alpha=2}$	94.37	93.89	80.87	**82.06**	0.0572	0.2428	0.1518	0.0159
$SLD_{\alpha=2}$	94.08	93.92	80.12	78.46	0.0600	0.2482	0.1529	0.0136
Trained on Cityscapes, prediction on Cityscapes test set								
One-hot	96.91	**98.02**	67.94	93.76	0.0346	0.2051	0.1066	0.1208
$AD_{\alpha=1}$	**97.02**	98.00	**68.26**	**94.11**	**0.0335**	***0.2022***	***0.1059***	0.1236
$SD_{\alpha=1}$	96.63	97.98	65.66	93.10	0.0373	0.2110	0.1088	***0.1078***
$SLD_{\alpha=1}$	96.79	***98.01***	65.63	93.81	0.0345	**0.1979**	**0.1030**	**0.0731**
$AD_{\alpha=2}$	***96.98***	98.00	***68.06***	***93.95***	***0.0340***	0.2041	0.1066	0.1261
$SD_{\alpha=2}$	96.95	***98.01***	68.02	93.87	0.0343	0.2046	0.1067	0.1244
$SLD_{\alpha=2}$	96.92	97.95	67.46	93.88	0.0346	0.2055	0.1075	0.1238

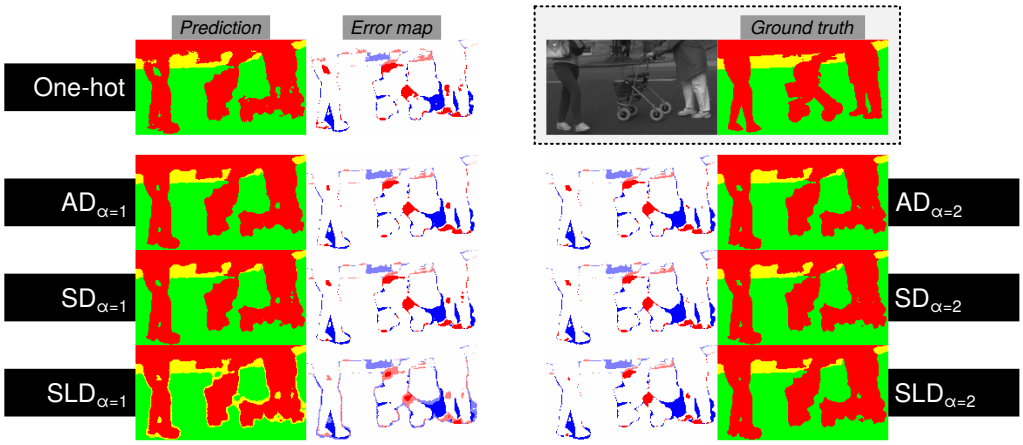
Table 6.2: Quantitative results of argmax predictions on the Freiburg Forest and Cityscapes test sets, comparing models trained on soft ordinal labels vs. the baseline one-hot label representation. For readability, we highlight the **best**, ***second-best***, and the worst result for each metric. We also highlight results which perform worse than the one-hot baseline.

Figure 6.12 presents a visual comparison between predictions by the different labelling schemes on a selected image crop from each dataset, and also includes an error map for each prediction, computed as the pixel-wise difference between the ground truth driveability level and the predicted (argmax) level. While the mistake severity and squared error metrics in the quantitative evaluation only capture the

magnitude of pixel classification error, here we also visualize it's direction: negative errors (in blue) correspond to an under-estimation of driveability (the model is being overly conservative about which areas are driveable), while positive errors (in red) are arguably worse in the context of safe navigation, since the model is over-estimating how driveable an area is. A light shade of blue or red indicates low mistake severity, and white indicates a correctly classified pixel. Looking at the overall aspect of the segmentation, the models trained on the hardest labels (one-hot followed by $SD_{\alpha=2}$) yield the most rough and spotty segmentation out-



(a) Freiburg Forest



(b) Cityscapes

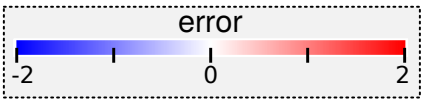


Figure 6.12: Argmax predictions and color-coded error maps of the 6 SORD models compared to the one-hot baseline, on a cropped sample from the Freiburg Forest and Cityscapes test sets.

lines, while the softer labelling schemes produce smoother contours. The two *SLD* models are the most effective at reducing the number of severe positive (red) errors: due to the asymmetrical class probability distributions in their labels, the model is encouraged to under-estimate rather than over-estimate driveability. Figure 6.13 shows a few examples of full-image predictions by these two models. For $SLD_{\alpha=1}$, this comes with an increase in the number of low-severity errors which visually manifests as a layer of ■ pixels around obstacles. For $SLD_{\alpha=2}$, severe errors transfer over to the opposite direction, resulting in obstacles which are slightly more over-segmented than by the rest of the models.

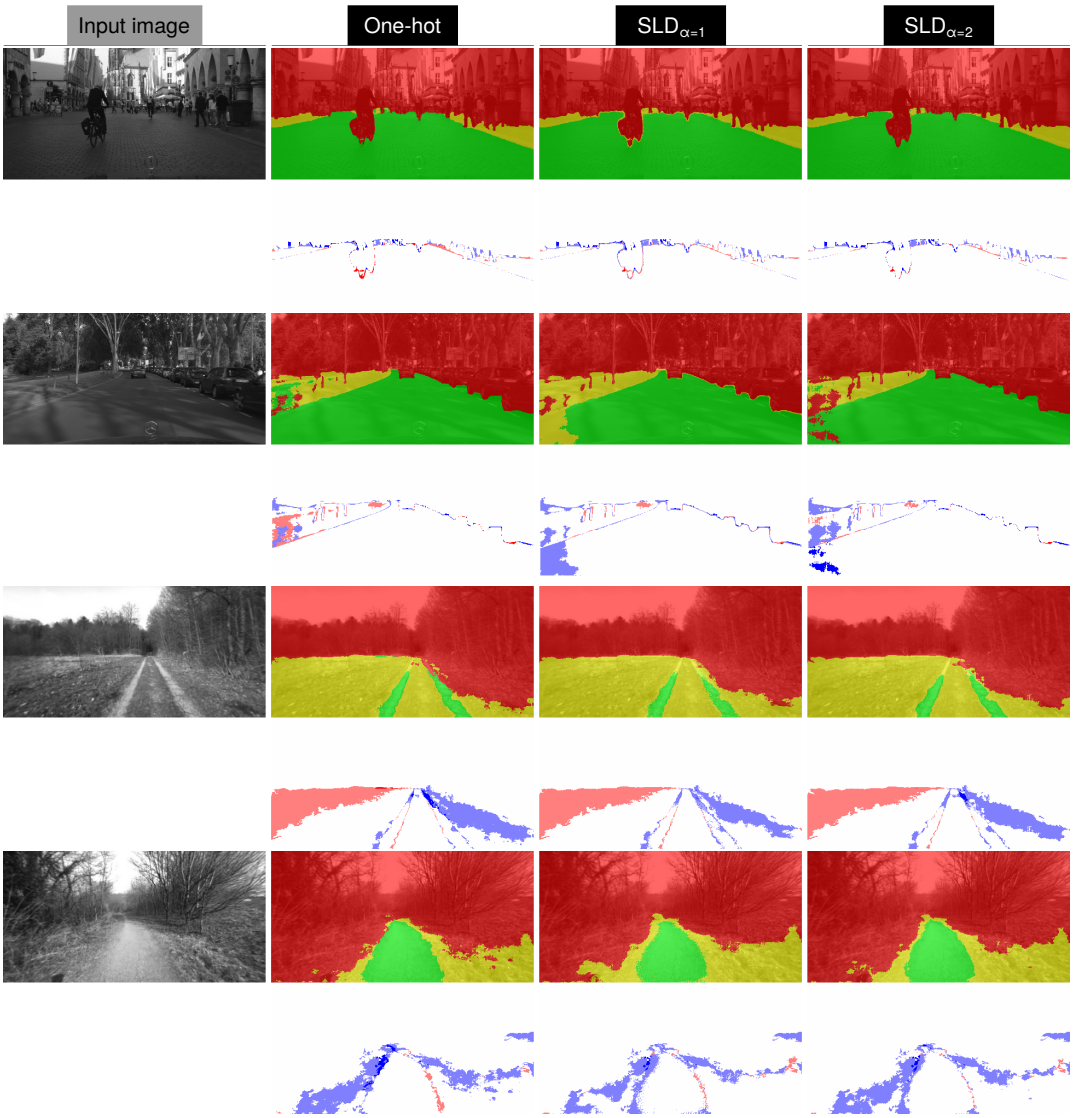


Figure 6.13: Argmax predictions (top) and color-coded error maps (bottom) of the 2 SLD models compared to the one-hot baseline, on samples from the Cityscapes and Freiburg Forest test sets.

6.4 Loss weighting

In the previous experiments, the loss per image is computed as an un-weighted average of pixel-wise loss, thus giving every pixel equal contribution during learning. Upon examination of the loss value per pixel on test images (Figure 6.14), we find that the highest loss occurs along segmentation boundaries - this aligns with [64, 125] which found boundary pixels to be the most difficult to segment.

However, for navigation applications, fine segmentation along the edges of different classes may not be necessary, since a vehicle is expected to stay in the middle of driveable areas. Thus, it may be beneficial to introduce leniency along boundaries during learning and evaluation, while giving more weight to central areas. Furthermore, we posit that it is more important to segment the scene correctly in areas close to the vehicle (ie. low in the image), as these directly influence driving decisions, whereas areas high in the image (eg. the sky) bear less relevance for our task. Based on these intuitions and following the method in [88] (albeit with a completely different goal), we implement a loss weighting scheme which encourages the segmentation network to learn pixels with higher importance.

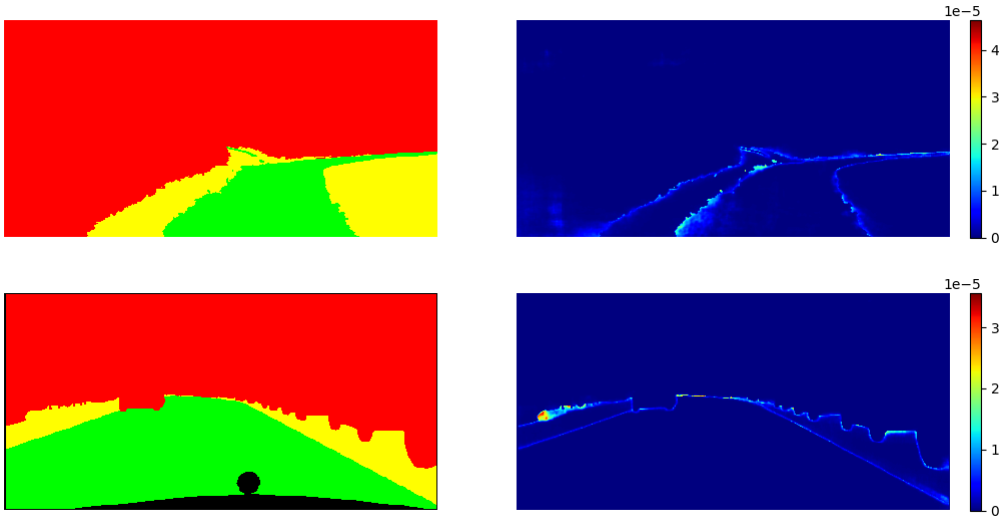


Figure 6.14: Loss on test samples from the Freiburg Forest (top) and Cityscapes (bottom) datasets, based on predictions by the driveability models from Section 6.2. The ground truth segmentation is shown on the left and the Kullback-Liebler divergence per pixel is visualized on the right.

We describe how we generate pixel-wise weight maps from ground truth segmentation masks, and incorporate them into the loss computation. In contrast to [88], where the pixel weight is maximized along borders, and exponentially decayed with increasing distance, we assign the lowest weight to border pixels, and incorporate a notion of depth, with higher edge tolerance for distant objects.

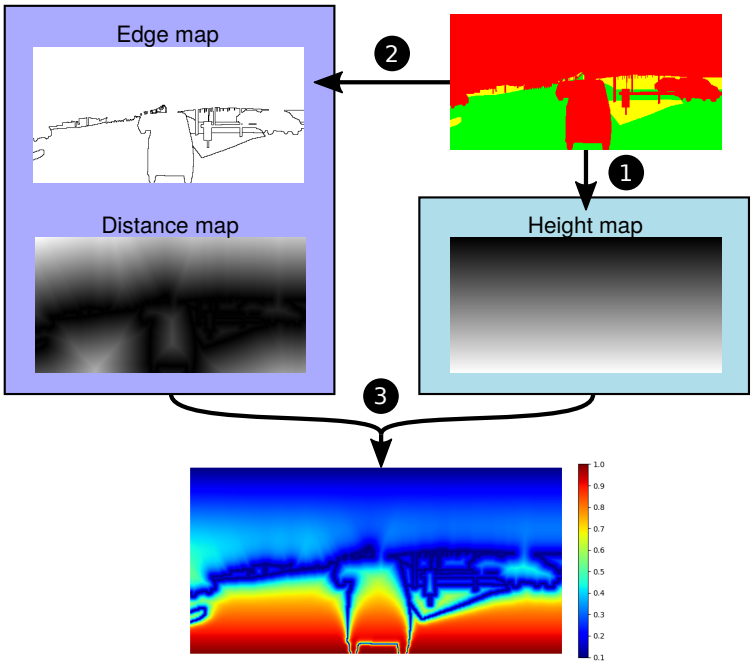


Figure 6.15: Steps in the weight map computation, numbered and illustrated with a ground truth sample from the Kitti dataset.

6.4.1 Generating weight maps

Given a pixel location $\mathbf{p} = [p_x, p_y]^T$ in the ground truth mask, we formulate a weight map which depends on its Euclidian distance $d(\mathbf{p})$ to the closest boundary and on its vertical position (height) in the image $h(\mathbf{p})$:

$$w(\mathbf{p}) = h(\mathbf{p}) \cdot \left[1 - \exp\left(-\frac{d(\mathbf{p})}{1 + \alpha(1 - h(\mathbf{p})^2)^2}\right)\right] \tag{6.1}$$

where α is a constant which we experimentally set to 30. The height map $h(\mathbf{p})$ is used to scale the rate at which the pixel weight increases when moving away from a boundary, and as a pixel-wise multiplication factor which assigns higher weight to lower pixels. We generate a weight map $w(\mathbf{p})$ for every ground truth mask in three steps which are illustrated in Figure 6.15:

1. the height map $h(\mathbf{p})$ is pre-computed for all possible pixel locations based on the image height H as: $h(\mathbf{p}) = p_y/H$ such that pixels in the lowest row of the image have the value 1 and those in the top row have a value of 0.
2. for computing the distance map $d(\mathbf{p})$, we first perform edge detection on the gray-scaled ground truth mask, binarize the edge map, and apply a distance transform [11] with a 5×5 kernel

3. the weight map $w(\mathbf{p})$ is computed following (6.1), and then normalized to lie within a range of 0.1 to 1.

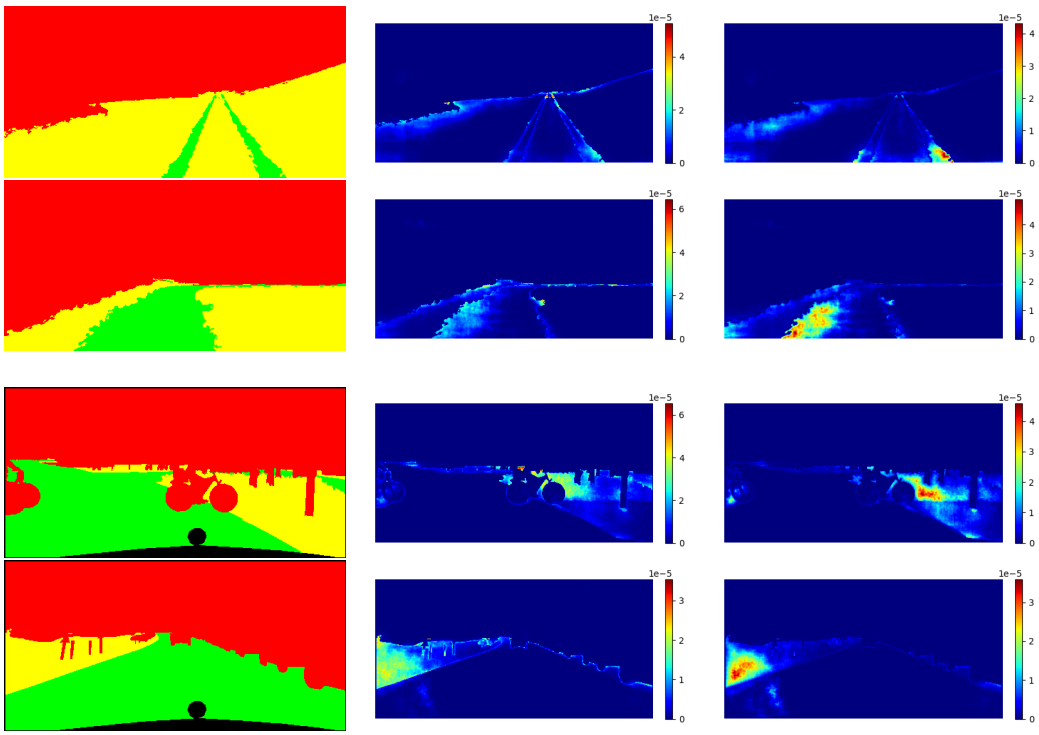
This results in a weight map with the lowest intensity along segmentation boundaries (dark blue in Figure 6.15), and the highest intensity in the lower region of the image (red). See Section A.6 for examples of a weight map generated on a Freiburg Forest and Cityscapes sample.

6.4.2 Weighted loss computation

Similarly to [88] and derivative works such as [35], the weight map is applied to the loss per pixel $\mathbf{p}$ via element-wise multiplication:

$$L_{KL}(\mathbf{y_p}||\hat{\mathbf{y}}_\mathbf{p}) = w(\mathbf{p}) \sum_{i \in C} y_{p,i} \log \frac{y_{p,i}}{\hat{y}_{p,i}}$$

where $\mathbf{y_p}$ and $\hat{\mathbf{y}}_\mathbf{p}$ are the pixel ground truth and predicted class probability vectors respectively, and C is the set of classes. As visualized in Figure 6.16, after applying



(a) Ground truth **(b)** Unweighted pixel-wise loss **(c)** Weighted pixel-wise loss

Figure 6.16: Loss on test samples from the Freiburg Forest (top) and Cityscapes (bottom) datasets, based on predictions by the driveability models from Section 6.2. The ground truth segmentation is shown on the left and the Kullback-Liebler divergence per pixel is visualized before (middle) and after applying the pixel-wise weight map (right).

the weight map, the loss per image is primarily concentrated in the bottom half of the image, and in areas away from boundaries, rather than along object contours.

6.4.3 Evaluation

For comparison with the standard unweighted loss approach, we repeat the same training procedure as in the transfer learning scheme of Section 6.2, where the network is initialized with weights from the object segmentation task, and then trained to predict driveability levels with an initial learning rate of 10^{-4} until convergence. The network is trained on the Freiburg Forest and Cityscapes datasets separately. We record pixel accuracy and also introduce a weighted accuracy measure, which assigns variable importance to pixels based on the weight map w generated from the ground truth mask. These two metrics are given in Table 6.3 and calculated as:

$$A_{unweighted} = \frac{\sum_{\mathbf{p}\in P} c(\mathbf{p})}{\sum_{\mathbf{p}\in P} 1} \qquad A_{weighted} = \frac{\sum_{\mathbf{p}\in P} w(\mathbf{p})c(\mathbf{p})}{\sum_{\mathbf{p}\in P} w(\mathbf{p})}$$

where P is the set of non-void pixels across all samples in the testing epoch and $c(\mathbf{p})$ is a correctness indicator function which returns 1 if the pixel p is correctly predicted (true positive or true negative) and 0 otherwise.

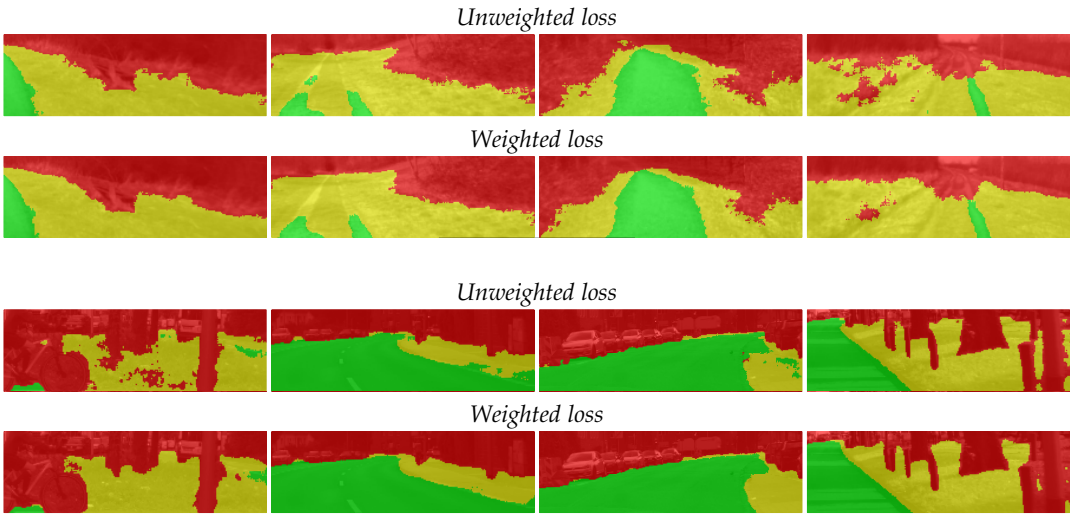
	Pixel accuracy		IoU			mistake severity
	$A_{unweighted}$	$A_{weighted}$				
Trained on Freiburg Forest, prediction on test set						
unweighted loss	**94.30**	93.68	93.85	80.70	**81.45**	**0.0173**
weighted loss	**94.30**	**93.79**	**93.91**	**80.80**	80.85	0.0179
Trained on Cityscapes, prediction on test set						
unweighted loss	**96.91**	**97.24**	**98.02**	**67.94**	**93.76**	**0.1207**
weighted loss	96.79	97.13	97.93	66.86	93.55	0.1209

Table 6.3: Performance of the driveability segmentation model with the proposed loss weighting scheme vs. the standard model from Section 6.2.

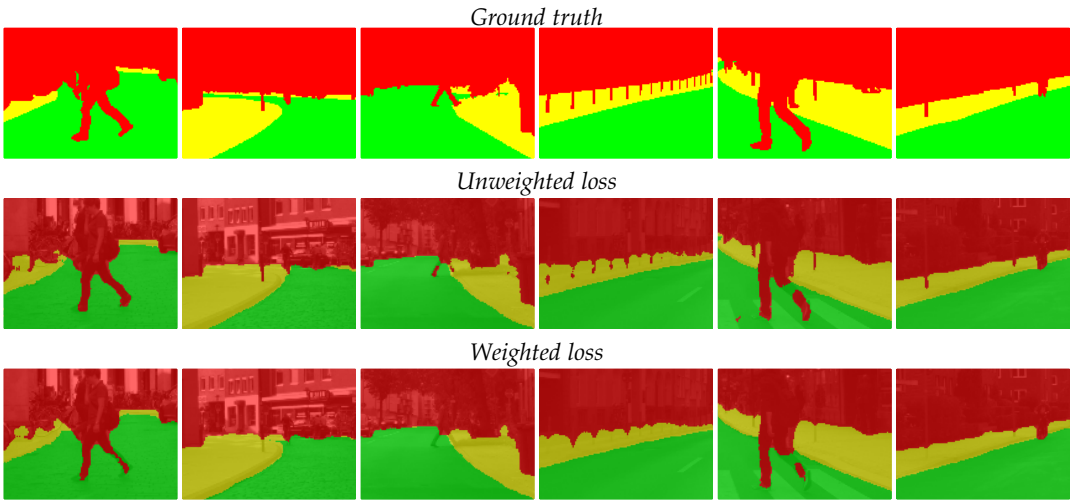
On the Freiburg Forest dataset, training the model with the proposed loss weighting method yields an increase in accuracy for the pixels of higher importance, as indicated by the weighted score. However, this improvement does not extend to the Cityscapes dataset.

Comparing the unweighted vs. weighted accuracy, it is interesting to note that for Cityscapes, the pixels of interest (low in the image and far from edges) are easier to learn, while for Freiburg Forest, the model achieves higher accuracy for distant pixels. This could be attributed to the fact that Cityscapes features much more cluttered scenes with objects which are difficult to segment precisely in the distance but

appear more clearly delineated in the foreground; in contrast, Freiburg Forest images have rather uniform background elements (primarily sky and vegetation), but more ambiguous terrain in the foreground. Although both models yield quantitatively similar results, the model trained under the loss weighting scheme produces less patchy or fuzzy segmentation for both datasets, with smoother boundaries and a noticeable loss of detail, especially in distant (high) areas (eg. legs, thin poles). We highlight this with selected image crops in Figure 6.17, and additionally show a side-by-side comparison of full-image predictions in Figure 6.18.



(a) Segmentation is smoother and less fragmented with loss weighting
Freiburg Forest (top) & Cityscapes (bottom)



(b) Details are lost along the way - Cityscapes

Figure 6.17: Crop of predictions on selected samples from the Freiburg Forest and Cityscapes test sets, showing the effect of loss weighting. Predictions are shown overlayed on the input image.

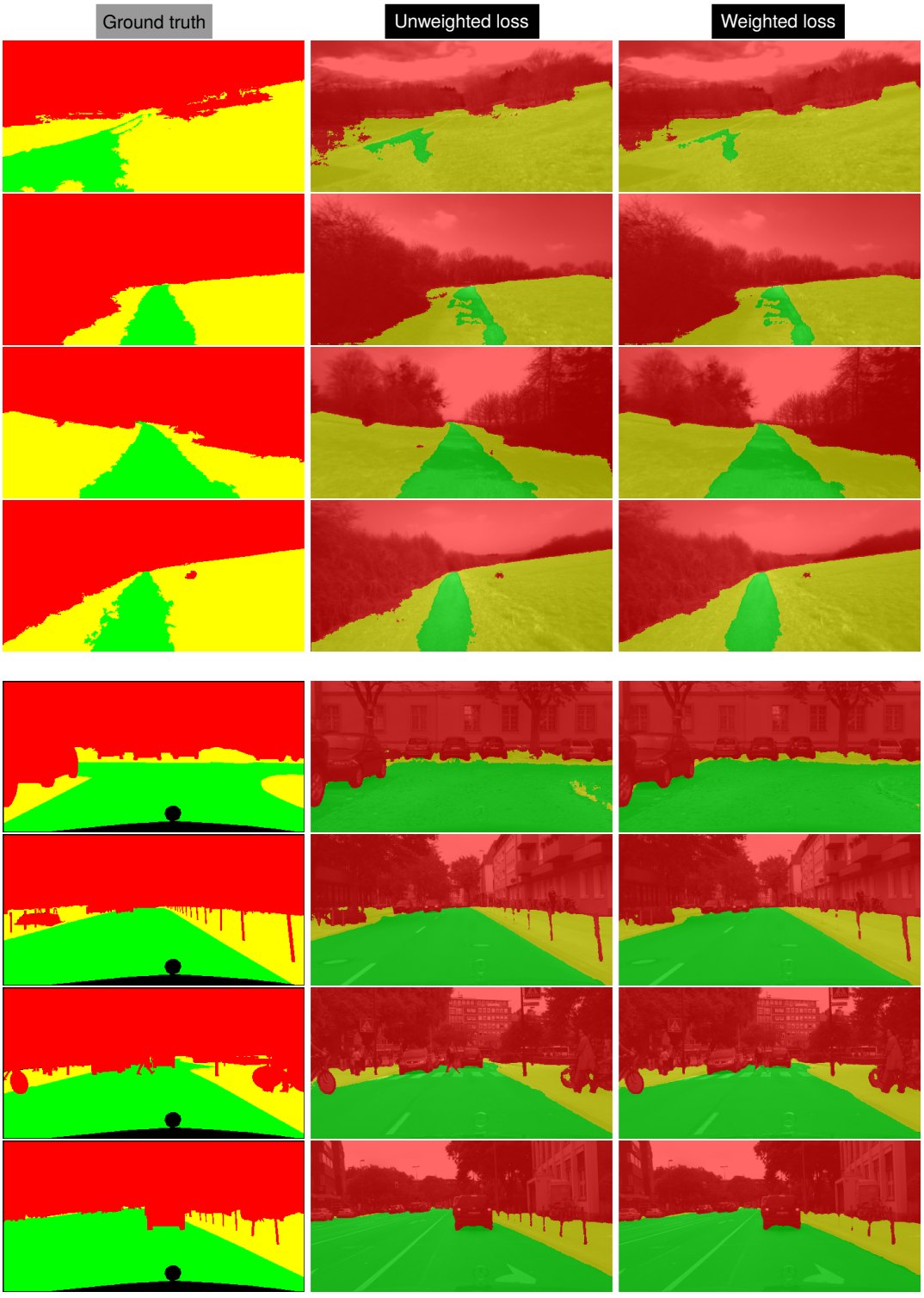


Figure 6.18: Selected samples from the Freiburg Forest (top) and Cityscapes (bottom) test sets for qualitative comparison of the proposed loss weighting scheme vs. the standard model from Section 6.2. Predictions are shown overlayed on the input image.

Chapter 7

Segmenting depth and infrared images

To guide the next experiments, the goal is to first train our segmentation network on alternative modalities from the Freiburg and Cityscapes datasets, to assess their usefulness as individual predictors of driveability, and to obtain pre-trained models which we later integrate into a multi-modal architecture. Since the use of depth and IR imaging also raises questions about the kind of pre-processing which should be applied, we also specifically evaluate the effect of data augmentation and filtering on segmentation performance.

In these experiments, for direct comparison with the model trained on visible spectrum images, we employ the same training procedure and hyper-parameters as in the experiments from Chapter 6, where SegNet is initialized with the baseline model trained on object classes and then adapted to learn 3 driveability levels. Rather than learning to segment each new modality from scratch, we use the pre-trained visible spectrum model weights for initialization of the object baseline, and then learn driveability on the new modality. We use one-hot encoded labels and no loss weighting during training. For evaluation, we report global pixel accuracy, weighted accuracy (cf. Section 6.4), class-wise IoU, as well as mistake severity (introduced in Section 6.3.4).

7.1 Data augmentation

In the experiments of Chapter 6, we trained SegNet on grayscale visible spectrum images with random photometric and geometric transformations (cf. Section 5.3.3) applied on-the-fly during training for data augmentation. However, whether these transformations should analogously be applied to infra-red or depth images for multi-modal segmentation remains an open question.

Therefore, we tackle this question by training the model on depth and IR images and comparing two data augmentation variants: in the *geom* variant, only geometric transforms are applied (thus leaving pixel intensity intact), and in the *geom+photo*, all the augmentations listed in Section 5.3.3 are applied. Similarly to [41], depth and IR images are min-max normalized to $[0, 255]$ - the same range as visible spectrum images. Results on the Freiburg dataset are reported in Table 7.1.

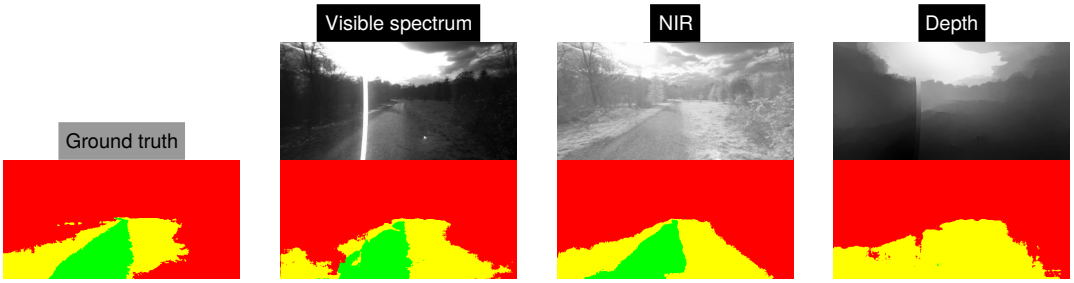
Evaluation Comparing the segmentation performance across modalities on the Freiburg dataset, visible spectrum images achieve the most consistent scores across levels and are the most informative modality for learning driveable areas (and), while NIR images yield superior segmentation for non-driveable areas ( ), but remain ambiguous when distinguishing between the and levels. The largest variation in performance between modalities can be seen in the IoU for the level; relying on depth data alone for segmentation yields poor results, since it especially lacks the visual cues necessary to distinguish between different types of terrain.

We also note that due to the nature of the depth data in this dataset, which was generated artificially from visible spectrum images alone, light artefacts such as glare or shadows wrongly transfer to the depth modality, as can be seen in Figure 7.1a. In this example, the sun ray causes discontinuity in the segmentation of the visible spectrum and depth images, while the NIR modality is unaffected and yields the most consistent segmentation. The two examples in Figure 7.1 also show that the pseudo depth images are rather poor estimates of true distance: clouds appear unrealistically close, the path appears darker/closer than the neighbouring grass, and the bench in Figure 7.1b is almost lost in the background.

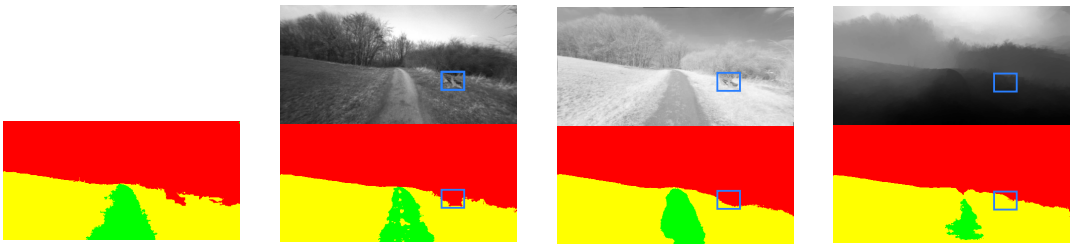
Comparing the two data augmentation variants, our results suggest that augmenting images with photometric transformations is beneficial for NIR, however it hinders performance for depth data. This also aligns with the intuition that intensity-based transformations are more sensible for visible spectrum and NIR images, since they simulate natural variations in illumination - applying them to depth maps is perhaps analogous to warping the scene geometry in an unrealistic manner. Therefore, in the following experiments, we implement the *geom* data augmentation variant for depth data, and $photo + geom$ for infra-red images.

	Pixel accuracy		IoU			mistake severity
	A	$A_{weighted}$				
Trained on Freiburg dataset, prediction on test set						
$V_{photo+geom}$	**94.30**	**93.68**	93.85	**80.70**	**81.45**	**0.0173**
D_{geom}	88.40	85.00	<u>91.20</u>	66.48	33.41	0.0236
$D_{photo+geom}$	<u>87.90</u>	<u>84.08</u>	91.21	<u>65.20</u>	<u>27.30</u>	0.0240
NIR_{geom}	92.77	92.10	93.98	77.31	61.05	<u>0.0359</u>
$NIR_{photo+geom}$	93.26	92.58	**94.55**	78.47	62.84	0.0228

Table 7.1: Segmentation performance on depth (D) and NIR images, compared to the visible spectrum (V) model from Section 6.2. We highlight the **best** and <u>worst</u> results for each metric.



(a) Effect of glare



(b) Segmentation of a mid-range obstacle, which is highlighted in blue

Figure 7.1: Comparison of segmentation results of different single-modality models ($V_{photo+geom}$, D_{geom} and $NIR_{photo+geom}$) on selected samples from the Freiburg test set.

7.2 Depth completion

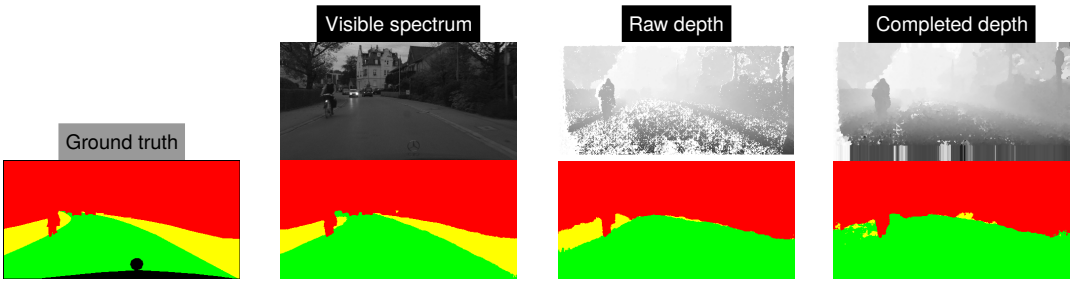
While [109] applies depth completion as a pre-processing step for multi-modal segmentation of Cityscapes scenes, the authors do not report comparative results to show its usefulness compared to training the network on raw disparity maps. Therefore, to investigate the effect of depth completion on this dataset, we compare the segmentation performance when training our network on raw stereo disparity maps vs. completed maps (generated following Section 5.3.1), with the same experimental procedure as in the previous section. Quantitative results are reported in Table 7.2.

	Pixel accuracy		IoU			mistake severity
	A	$A_{weighted}$				
Trained on Cityscapes dataset, prediction on test set						
V	**96.91**	**97.24**	**98.02**	**67.94**	**93.76**	**0.1207**
D_{raw}	96.24	97.15	97.12	60.39	92.80	0.1707
D_{comp}	<u>95.64</u>	<u>96.51</u>	<u>96.63</u>	<u>56.84</u>	<u>91.25</u>	<u>0.2242</u>

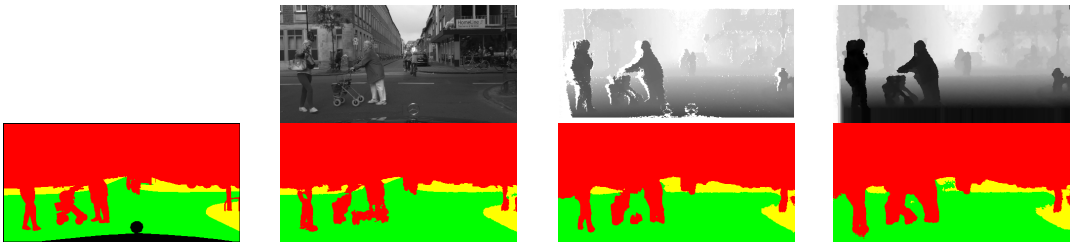
Table 7.2: Segmentation performance on depth (D) images, compared to the visible spectrum (V) model from Section 6.2. We highlight the **best** and <u>worst</u> results for each metric.

Evaluation Comparing the models trained on raw disparity maps (D_{raw} in Table 7.2) vs. depth-completed maps (D_{comp}), applying depth completion as a pre-processing step results in a clear drop in performance across all metrics, especially for segmentation of the driveability level. Figure 7.2a shows an example where the D_{comp} model incorrectly segments the entire ground area as , while the D_{raw} model is able to correctly predict the left sidewalk as . This is most likely due to the loss of detail caused by the filtering operations in the depth completion process: while missing values are filled, objects lose their sharp delineation. Depth completion also makes obstacles appear fuller and results in a more approximate, "blobby" segmentation, as can be seen for the pedestrians and pole in Figure 7.2b. In this example, D_{comp} is the only model which correctly segments the stroller as a single object.

We also note that compared to the Freiburg dataset where pseudo depth alone is a poor indicator of driveability level (cf. Table 7.1), on the Cityscapes dataset, the network is able to accurately segment scenes in terms of driveability from stereo disparity, especially in the areas of interest (cf. Section 6.4), with less than a 1.5% drop in overall pixel accuracy for both D models compared to the model trained on visible spectrum images, and an even smaller drop for weighted pixel accuracy (less than 0.1% for D_{raw}).



(a) Sidewalk boundaries are lost with depth completion



(b) Obstacles are over-segmented with depth completion

Figure 7.2: Comparison of segmentation results of different single-modality models (V, D_{raw} and D_{comp}) on selected samples from the Cityscapes test set.

Chapter 8

Multi-modal fusion

In this chapter, we explore how the proposed driveability levels (presented in Section 4.1) can be learned from multi-modal imaging data, with the aim of improving segmentation compared to relying on visible spectrum images alone. We first present and evaluate a fusion baseline in Section 8.1, where images are combined at the input of the network via simple concatenation - thus requiring minimal architectural changes. In Section 8.2, we adapt the work proposed in [109], and explore middle and late fusion configurations where feature representations from modality-specific branches are fused adaptively, allowing the network to learn cross-modal correlation and complementarity.

8.1 Channel stacking for early fusion

Similarly to [92, 38, 109], we implement channel stacking as a baseline for multi-modal segmentation. In this case, as illustrated in Figure 8.1, fusion is performed at the lowest level by concatenating multi-modal images at the input of the network: each modality is incorporated as an additional image channel, and the network can be trained end-to-end much like in our single-modality experiments.

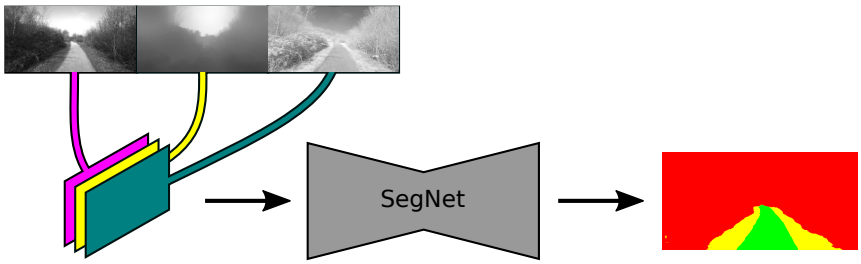


Figure 8.1: Channel stacking for multi-modal segmentation, illustrated with a sample from the Freiburg dataset. Modalities from left to right: visible spectrum, pseudo depth, NIR.

We investigate segmentation performance for different modality combinations in the Freiburg (visible spectrum, NIR, and depth) and Cityscapes (visible spectrum, depth) datasets, following the same training and evaluation procedure as in Chapter 7's single-modality experiments. In the initialization stage, we increase the number of input channels of the first convolutional layer to the number of modalities stacked in the input. Rather than learning these new weights from scratch, we copy the weights from the existing channel trained on visible spectrum images.

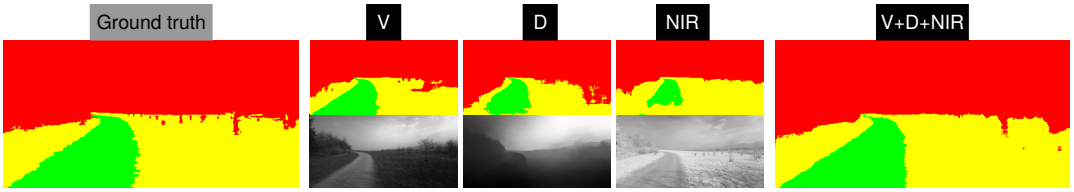
	Pixel accuracy		IoU			mistake severity
	A	$A_{weighted}$	🟥	🟨	🟩	
Trained on Freiburg dataset, prediction on test set						
V	<u>94.30</u>	<u>93.68</u>	<u>93.85</u>	<u>80.70</u>	81.45	0.0173
$V + D$	94.41	93.83	94.07	81.22	<u>80.80</u>	**0.0152**
$V + NIR$	**95.07**	**95.24**	**94.87**	**83.40**	**81.93**	<u>0.0194</u>
$V + D + NIR$	94.83	94.93	94.62	82.47	81.69	0.0181
Trained on Cityscapes dataset, prediction on test set						
V	**96.91**	**97.24**	**98.02**	**67.94**	**93.76**	**0.1207**
$V + D_{raw}$	<u>96.60</u>	<u>97.00</u>	<u>97.61</u>	<u>65.22</u>	<u>93.33</u>	0.1370
$V + D_{comp}$	96.74	97.14	97.67	66.38	93.59	<u>0.1450</u>

Table 8.1: bunch of metrics after mashing modalities together, **best** and <u>worst</u>

Evaluation Table 8.1 compares the performance of single-modality vs. early-fused multi-modal prediction on the Freiburg and Cityscapes dataset. For the Freiburg dataset, results suggest that combining visible spectrum images with depth and/or NIR data in the input is valuable for segmentation. Adding depth data reduces mistake severity while improving IoU for the least driveable areas. Incorporating NIR data yields the highest gain in accuracy and IoU across all levels, however it also results in more severe mistakes - mis-classifications are more likely to occur between the 🟥 and 🟩 levels than between successive levels. Interestingly, unlike the V and V+D inputs where the weighted accuracy remains lower than the overall accuracy, adding NIR data improves segmentation in the areas of interest for navigation. Figure 8.2a shows some examples for which multi-modal prediction via channel stacking is beneficial for segmentation.

In contrast, for Cityscapes, stacking visible spectrum and depth images degrades performance compared to relying on visible spectrum images alone. Furthermore, unlike in the single-modality experiment where more accurate segmentation is achieved using raw depth maps than completed maps, we observe the opposite here, where incorporating raw depth yields the largest drop in accuracy

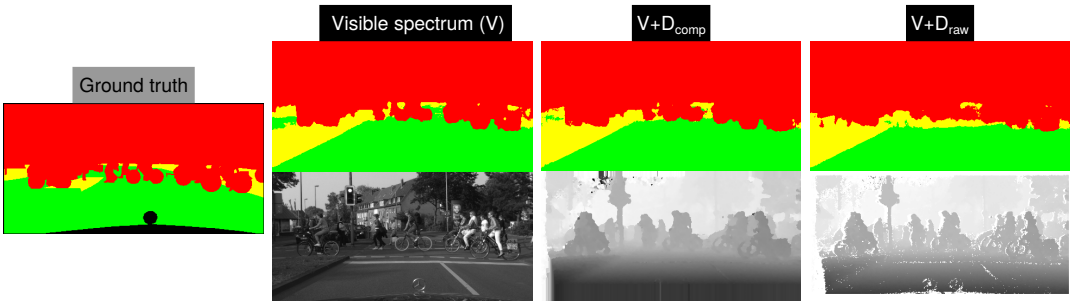
and IoU. We attribute this to the mis-match in data sparsity when combining visible spectrum images with raw depth, causing the input channels to be less correlated than when missing depth values are filled. The drop in segmentation quality caused by channel stacking is shown in Figure 8.2b with two examples.



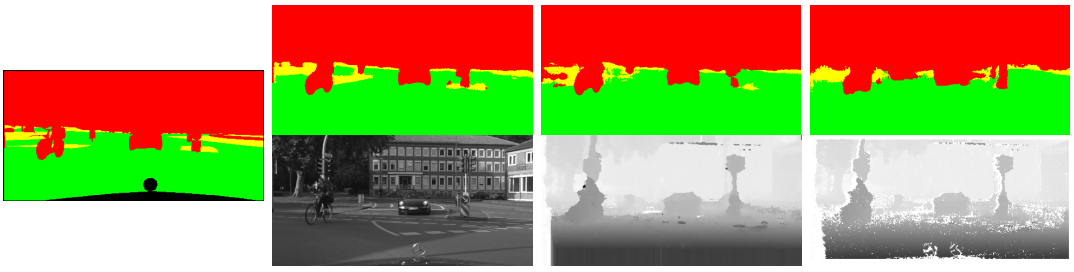
While some background detail from distant trees is lost, the closest ones on the right are more precisely segmented when fusing the three modalities, and the path remains correctly segmented.

Even though the path is incompletely segmented by the single-modality models, it is fully recovered after fusion. The trees on the left are also correctly excluded from driveable areas.

(a) Freiburg Forest: three single-modality models (middle) vs. channel stacking model (right)



Channel stacking with raw depth maps adds noise in the segmentation and results in under-segmented obstacles. Note that the sidewalk area is incorrectly annotated in the ground truth in this example.



Channel stacking degrades the segmentation compared to the visible spectrum model, especially when using raw depth maps. areas in the middle of the road are lost and boundaries appear ill-defined.

(b) Cityscapes: single-modality model vs. two channel stacking models

Figure 8.2: Qualitative results of channel stacking on selected test set samples.

8.2 Cooler fusion

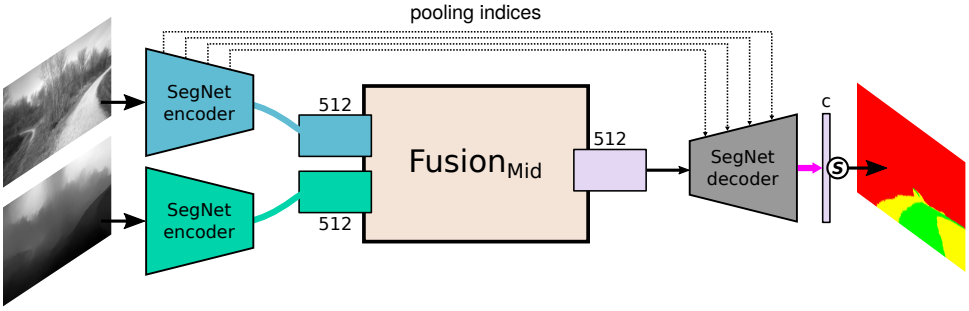
In this section, we investigate how modality-specific features can be extracted and fused at a later stage in the network. While a standard approach consists of fusing feature maps via a fixed operation such as concatenation [38] or element-wise addition [41, 102], this implicitly considers each modality as equally and unconditionally informative for segmentation, regardless of scene context or class. To surpass these limitations, we draw from the insights in [109], which proposes to learn feature-level correlation between different modalities in order to fuse modality-specific feature maps adaptively.

The work in [109] proposes a full segmentation architecture for multi-modal segmentation. Each modality is fed into a separate ResNet-based encoder [42] for feature extraction, followed by a spatial pyramid pooling module with dilated convolutions to capture multi-scale contextual information. These modality-specific streams are then combined in the middle of the network by a fusion unit whose output is up-sampled by a single common decoder. In addition, for refining segmentation in the up-sampling stage, the network employs skip connections: at different stages, feature maps from parallel encoders are fused and then concatenated into the corresponding decoding layer. Although this architecture achieves state-of-the art results, its multi-stage fusion topology cannot be directly applied with a SegNet-based network. Indeed, SegNet fundamentally differs in its up-sampling technique: rather than concatenating entire feature maps into the decoder and performing bi-linear interpolation, SegNet's decoder up-samples via max-unpooling, guided by pooling indices from the encoder's max-pooling layers. Therefore, rather than replicating the full architecture in [109], we only make use of its Self-Supervised Model Adapation (SSMA) fusion unit and integrate it into a SegNet-based architecture, leveraging our pre-trained models from previous experiments. In Section 8.2.1, we first present three fusion architectures which preserve SegNet's encoder and decoder design. Section 8.2.2 then describes the fusion unit itself, and how we adapt it to our configuration.

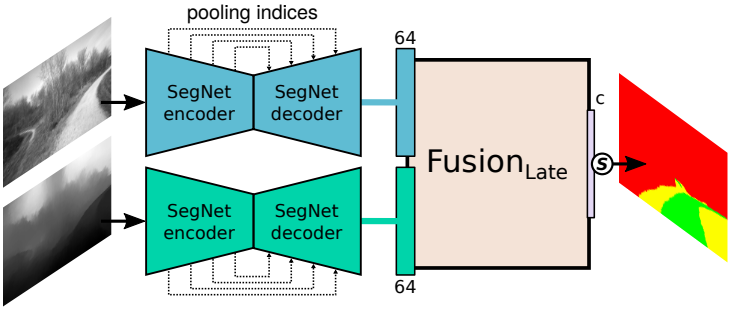
8.2.1 Multi-modal segmentation architecture with indexed unpooling

The three fusion architectures that we employ in our experiments are illustrated in Figure 8.3. For feature extraction, we opt for parallel, independent encoders which each specialize in a particular modality. Each encoder outputs a low-resolution feature map which encodes high-level, modality-specific representations of the scene. Similarly to [109], these feature maps can then be fused into a single cross-modal representation by a fusion unit in the middle of the network. The number of channels at the input and output of the fusion unit is adapted to the feature depth of 512 extracted by SegNet's encoder.

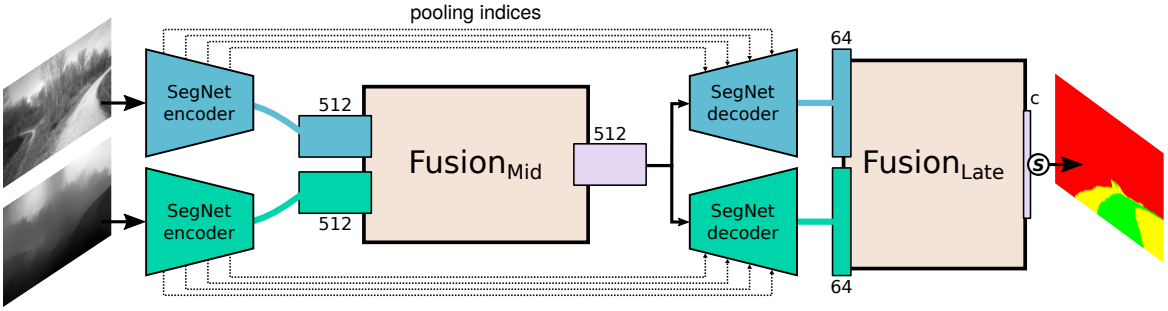
In SegNet, at each down-sampling stage in the encoder, the max-pooling indices are stored to perform indexed max-unpooling at the corresponding up-sampling stage in the decoder. However, in a fusion architecture, the max-pooling indices from the parallel encoders cannot simply be combined like feature maps, since their value encodes spatial locations rather than activations. While it may be beneficial to investigate fusion mechanisms for combining pooling indices from different encoders, we take a simple approach where the pooling indices for each modality are left intact, and compare three different strategies in this direction.



(a) ***Mid*** architecture: middle fusion with modality-specific encoders and a common decoder. For up-sampling in the decoder, only the pooling indices of a single encoder are used.



(b) ***Late*** architecture: late fusion with modality-specific and independent SegNet networks.



(c) ***Dual*** architecture: middle and late fusion with modality-specific encoders and decoders.

Figure 8.3: Segmentation architecture variants used for multi-modal pixel-wise prediction, based on the SegNet [4] network presented in Section 6.1, illustrated with two modalities from the Freiburg dataset. See Figure 8.4 for a legend.

1. The ***mid*** architecture in Figure 8.3a consists of a single decoder which takes fused feature maps as input, but only uses the pooling indices of a single modality-specific encoder for up-sampling. This requires defining a "main" modality which is the most informative for recovering scene detail. In our experiments, based on the single-modality results in Chapter 7, we choose to use the pooling indices of the visible spectrum encoder.

2. An alternative (shown in Figure 8.3b) consists of employing independent segmentation branches up to the prediction stage, and performing ***late*** fusion after feature maps have been up-sampled back to the input image's resolution. In this case, a fusion unit is added at the end of the network, in order to fuse the 64-channel feature maps produced by the decoders into a single c-channel prediction (where c is the number of classes).

3. Thirdly, we present a ***dual*** architecture, where fusion is performed both in the middle and late in the network. As shown in Figure 8.3c, the network keeps a symmetrical topology with modality-specific decoders - the pooling indices of each modality-specific encoder are transferred to the corresponding decoder. However, unlike the late fusion architecture, the modality-specific decoders are fed the same fused feature map for up-sampling, which they then refine independently.

For all three architectures, SegNet's encoder and decoder are left intact, making it seamless to integrate several pre-trained single-modality SegNet networks for fusion.

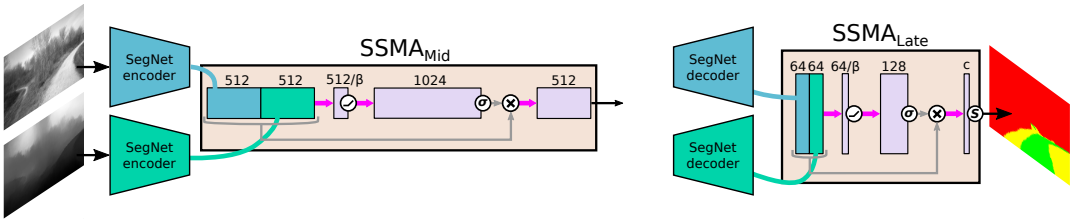
8.2.2 Fusion units

Here we present the SSMA unit from [109], and build on top of this work by proposing a custom unit which is more parameter-efficient and facilitates learning in our configuration. Both units are integrated into the *mid* and *dual* architecture variants presented in Section 8.2.1 for comparison in our experiments.

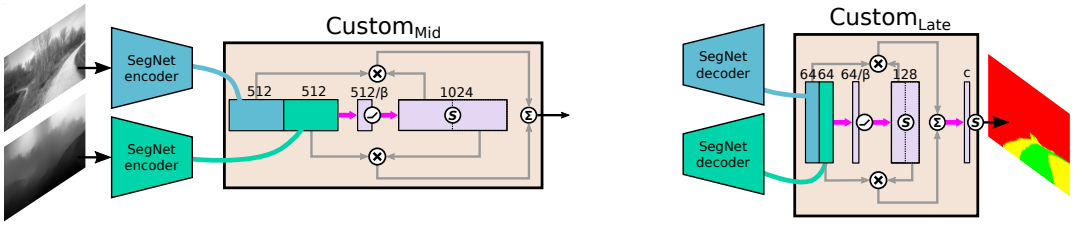
The main idea behind the SSMA unit is to dynamically weigh modality-specific input streams depending on their respective features, in order to optimally combine them into a fused representation. Specifically, the goal is to learn a rich, non-linear mapping between the activations from modality-specific input streams, and element-wise multiplication factors which selectively suppress or emphasize each input stream. This mapping is modelled by a small convolutional sub-network which extracts cross-modal features while preserving spatial structure, and produces a bounded activation for every element in the input streams, which represents how much the input element should contribute to the fused output.

SSMA architecture The SSMA unit takes n modality-specific feature maps of depth D_{in} as input and outputs a single feature map with the same spatial dimensions, whose depth D_{out} matches the number of input channels in the next stage. In its original formulation [109], the SSMA unit is only used for mid-level fusion, where $D_{in} = D_{out}$. In this work, we also adapt it for late fusion to fuse the decoder outputs into a single prediction, in which case $D_{out} = c$. The unit's architecture is detailed below and illustrated in Figure 8.4a with $n = 2$ and $D_{in} = 512 \mid D_{out} = 512$ for middle fusion and $D_{in} = 64 \mid D_{out} = c$ for late fusion (based on the layer dimensions in SegNet).

At the input of the SSMA unit, the modality-specific feature maps (shown in blue and green in Figure 8.4) are first concatenated depth-wise, yielding a feature map of depth nD_{in}. They are then fed through a convolutional bottleneck layer with ReLU activation for dimensionality reduction, resulting in a nD/β feature map, where β is the bottleneck compression rate. A second convolutional layer then expands the feature map back to a depth of nD_{in}, such that it can be multiplied element-wise with the concatenated input feature maps. This layer is activated with a sigmoid function, such that after multiplication, each element in the feature map is scaled by a factor between 0 and 1. Lastly, a third convolutional layer is applied to recover a feature map of depth D_{out}. While [109] additionally



(a) SSMA unit from [109], adapted for the SegNet architecture.



(b) Proposed modification of the SSMA unit.

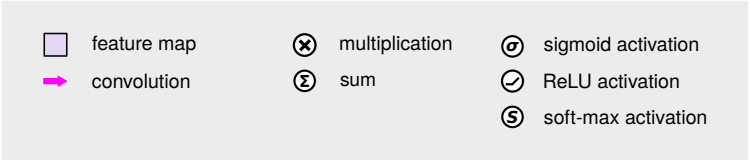


Figure 8.4: Fusion units for middle (left) and late (right) fusion. β denotes the bottleneck compression rate (set to 16 in our experiments), and c corresponds to the number of classes (3 in our experiments).

applies batch-normalization on the fused feature map, we find that it significantly degrades performance in our configuration, and therefore omit this layer (see Section A.1 for a comparison of the model learning curves with vs. without batch normalization).

Custom fustion unit In our configuration, where we aim to leverage pre-trained decoder weights rather than learning them from scratch, the un-bounded output of the SSMA unit (due to the third convolutional layer) is sub-optimal, since it may widely differ from the original feature maps fed to the decoder when training the single-modality model. Therefore, we propose an alternative fusion unit which is based on the same fully convolutional bottleneck architecture as SSMA, but differs in the way that the original modality-specific feature maps are then weighed and combined. As illustrated in Figure 8.4b, rather than element-wise multiplication followed by depth reduction, we apply an element-wise weighted sum between modality-specific feature maps which directly results in a combined feature map of depth D_{in}. Specifically, the feature map of depth nD_{in} produced by the first two convolutional layers is split depth-wise into n modality-specific feature maps, and softmax activation is applied along the modality dimension, such that cross-modality elements in the same channel and spatial location sum to 1. Modality-specific feature map pairs are then multiplied element-wise and summed across modalities to produce a fused map.

This approach greatly facilitates training when using pre-trained decoder or output layers, as further detailed below in Section 8.2.3, since the output of the fusion unit remains in the same range as the input feature maps. It also bypasses the need for a third convolutional layer (in the mid-fusion case), thus reducing the number of parameters.

Parameters For both fusion unit variants, in all layers, convolution is performed with a single-strided 3×3 kernel and zero-padding to preserve spatial resolution. Based on the results in [109], we set the bottleneck compression rate β to 16.

8.2.3 Initialization and training procedure

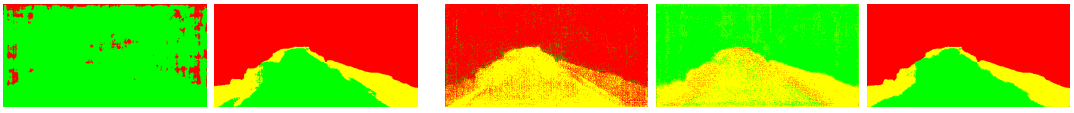
In our experiments, we compare the *mid*, *late* and *dual* architectures presented in Section 8.2.1, and implement each architecture with *SSMA* vs. *Custom* fusion units described in Section 8.2.2 - resulting in six segmentation network variants which are trained and evaluated: mid_{SSMA}, mid_{Custom}, $late_{\text{SSMA}}$, $late_{\text{Custom}}$, $dual_{\text{SSMA}}$, $dual_{\text{Custom}}$.

Where to start Unlike [109] where the weights of the decoder must be learned from scratch, we initialize the encoders and decoders in our fusion network using pre-trained weights from the single-modality SegNet models from Section 6.2 (visible spectrum) and Chapter 7 (depth & IR), which were all trained following the same transfer learning scheme described in Section 6.2.

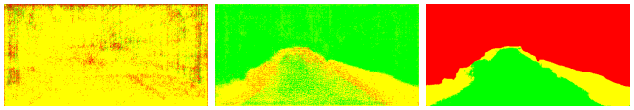
For the *mid* configuration, the common decoder is initialized with the weights from the visible spectrum model. For the *dual* configuration, the decoders are initialized with modality-specific weights, similarly to the encoders. Additionally, since the last convolutional layer in our $Custom_{\text{Late}}$ unit has the same dimensions as the last layer of SegNet, we can initialize it with pre-trained weights from the visible spectrum model as well. The same cannot be done for the $SSMA_{\text{Late}}$ unit, since its last layer has double the number of input channels. Thus, only 2 layers in our custom fusion unit need to be learned from scratch during training, as opposed to 3 for SSMA. In our experiments, to assess the effect of pre-training $Custom_{\text{Late}}$'s last layer, we additionally distinguish between the $dual_{\text{Custom,PTLL}}$ model (pre-trained last layer) and $dual_{\text{Custom}}$, where the last layer is initialized randomly like in the SSMA unit. For random initialization of untrained layers, we apply Kaiming initialization [43] in case of ReLU activation, and Xavier initialization [33] in case of sigmoid or softmax activation.

The network variants with SSMA vs. Custom fusion units produce drastically different outputs upon initialization. We illustrate this in Figure 8.5, which shows the segmentation output on a Cityscapes sample after initializing the five network variants with our pre-trained SegNet model weights. Unlike SSMA, upon random initialization of the custom fusion unit's first two convolutional layers, the features it produces are a randomly weighted yet scale-preserving combination of modality-specific features; thus, the mid_{Custom}, $late_{\text{Custom,PTLL}}$ and $dual_{\text{Custom,PTLL}}$ networks already output viable predictions prior to training. As opposed to mid_{SSMA} and mid_{Custom}, the $dual_{\text{SSMA}}$ and $dual_{\text{Custom}}$ networks have a randomly initialized last layer, resulting in a noisy mapping from feature space to label space - however, the structure of the scene remains clearly visible in $dual_{\text{Custom}}$'s prediction, since the network's activations up until the last layer remain consistent after adding custom fusion units before and after the pre-trained decoders.

(a) Input sample - visible spectrum image (left) and raw disparity map (right)



(b) mid_{SSMA} **(c)** mid_{Custom} **(d)** $late_{\text{SSMA}}$ **(e)** $late_{\text{Custom}}$ **(f)** $late_{\text{Custom,PTLL}}$



(g) $dual_{\text{SSMA}}$ **(h)** $dual_{\text{Custom}}$ **(i)** $dual_{\text{Custom,PTLL}}$

Figure 8.5: Segmentation output on a Cityscapes test sample of the different fusion networks in our evaluation, after initialization and prior to training.

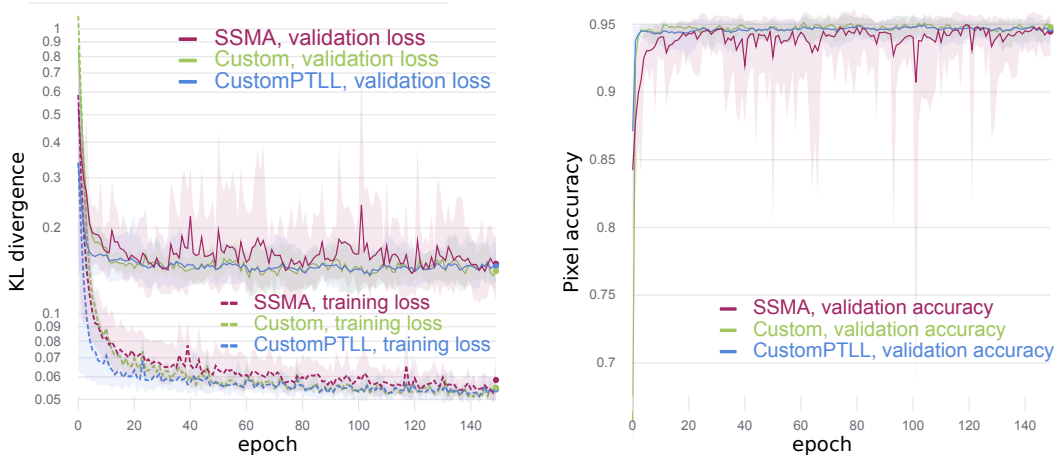
Learning things For training, we use the same hyper-parameters and training procedure as in the previous experiments. Thus, unlike [109] which follows a multi-stage training scheme with different initial learning rates for the encoder vs. decoder layers and polynomial learning rate decay, we apply the same initial learning rate of 10^{-4} across all parameters, and train the network end-to-end in a single phase without learning rate scheduling. While this may not yield optimal performance, it allows for direct comparison with our single-modality and early fusion models.

8.2.4 Evaluation

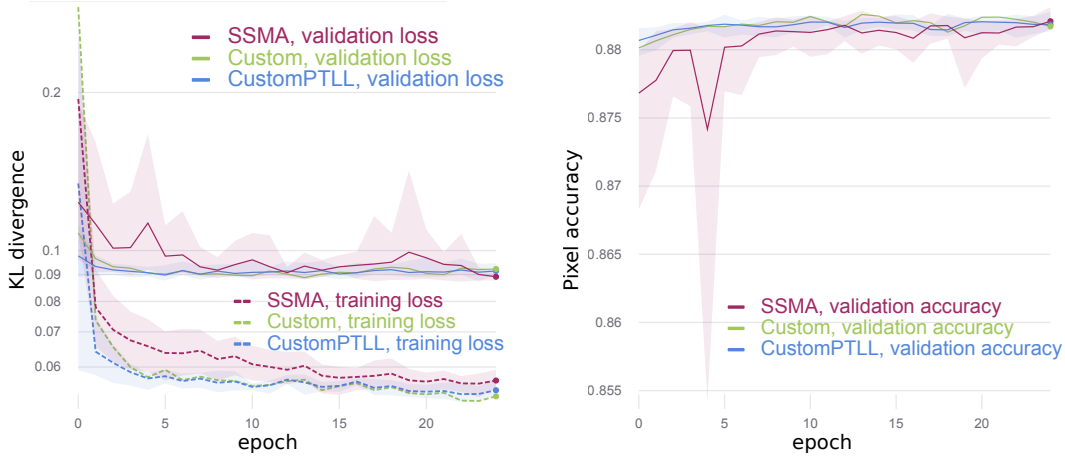
We evaluate the 8 network variants on different modality combinations from the Freiburg and Cityscapes datasets based on the same evaluation metrics as the previous experiments, and compare their performance to the single-modality and early fusion models. Although the fusion architecture has so far only been illustrated for two modalities, we incorporate a third branch for $V + D + NIR$ fusion, resulting in 3 encoders in the feature extraction stage, and 3 decoders in the *late* and *dual* architectures for symmetry. We first point out significant discrepancies in the training process depending on the kind of fusion unit in the network. We then report and discuss the quantitative results in Table 8.2.

SSMA mood swings Figure 8.6 compares the learning curves of models trained with SSMA vs. Custom fusion units. We note significant instability in validation loss and accuracy when training SSMA-based fusion models, while those based on

Custom fusion units quickly converge with minimal fluctuations in performance on the validation set. Additionally, initializing the Custom fusion unit's last layer with pre-trained weights for late fusion ($late_{\text{Custom,PTLL}}$ and $dual_{\text{Custom,PTLL}}$ models) results in high validation accuracy within the first training iteration, and brings a small performance increase compared to the $late_{\text{Custom}}$ and $dual_{\text{Custom}}$ models (where the last layer is randomly initialized) in some cases (cf. Table 8.2).



(a) Freiburg dataset (24 models)



(b) Cityscapes dataset (16 models)

Figure 8.6: Learning curves during training of the 8 architecture variants on different modality combinations (cf. Table 8.2). The models are grouped based on the unit used for fusion (SSMA vs. Custom vs. CustomPTLL), and we show the mean and min-max range for the loss (left) and pixel accuracy (right) per epoch. The loss is plotted on a log-scale for clarity.

modalities	fusion	Pixel accuracy A	Pixel accuracy $A_{weighted}$	IoU	IoU	IoU	mistake severity
V	-	94.30	93.68	93.85	80.70	81.45	0.0173
$V+D$	early	94.41	93.83	94.07	81.22	80.80	0.0152
$V+D$	mid_{SSMA}	***94.87***	**94.47**	94.56	82.58	***82.12***	0.0221
$V+D$	mid_{Custom}	94.56	93.93	94.33	81.66	80.68	**0.0113**
$V+D$	$late_{\text{SSMA}}$	94.65	94.10	94.29	81.83	82.06	0.0131
$V+D$	$late_{\text{Custom}}$	94.42	93.76	94.24	81.35	79.41	0.0140
$V+D$	$late_{\text{Custom,PTLL}}$	94.48	93.86	94.37	81.56	79.10	0.0142
$V+D$	$dual_{\text{SSMA}}$	94.42	93.89	**94.80**	81.96	74.46	0.0139
$V+D$	$dual_{\text{Custom}}$	94.61	94.08	94.58	***82.11***	78.94	0.0134
$V+D$	$dual_{\text{Custom,PTLL}}$	**94.93**	***94.46***	***94.74***	**82.98**	***81.42***	***0.0115***
$V+NIR$	early	95.07	95.24	94.87	83.40	81.93	0.0194
$V+NIR$	mid_{SSMA}	**96.13**	**96.30**	**95.87**	**86.68**	**86.42**	***0.0151***
$V+NIR$	mid_{Custom}	95.36	95.17	95.30	84.25	82.10	**0.0144**
$V+NIR$	$late_{\text{SSMA}}$	95.53	95.48	95.29	84.78	***84.08***	0.0153
$V+NIR$	$late_{\text{Custom}}$	95.36	95.17	95.42	84.55	80.73	**0.0144**
$V+NIR$	$late_{\text{Custom,PTLL}}$	95.38	95.19	95.43	84.57	80.83	0.0154
$V+NIR$	$dual_{\text{SSMA}}$	95.39	***95.58***	95.06	84.45	83.62	0.0353
$V+NIR$	$dual_{\text{Custom}}$	95.41	95.36	95.32	84.51	82.18	0.0176
$V+NIR$	$dual_{\text{Custom,PTLL}}$	***95.58***	***95.58***	***95.64***	***85.14***	81.46	0.0178
$V+D+NIR$	early	94.83	94.93	94.62	82.47	81.69	0.0181
$V+D+NIR$	mid_{SSMA}	95.65	95.47	95.36	84.98	85.13	***0.0146***
$V+D+NIR$	mid_{Custom}	95.30	95.08	95.42	84.23	80.12	**0.0145**
$V+D+NIR$	$late_{\text{SSMA}}$	95.43	95.37	95.34	84.57	82.24	0.0174
$V+D+NIR$	$late_{\text{Custom}}$	95.71	95.69	95.33	85.25	**86.02**	0.0156
$V+D+NIR$	$late_{\text{Custom,PTLL}}$	95.61	95.56	95.21	84.93	85.75	0.0166
$V+D+NIR$	$dual_{\text{SSMA}}$	95.62	95.46	95.52	85.06	83.11	0.0185
$V+D+NIR$	$dual_{\text{Custom}}$	**95.98**	**96.04**	**95.73**	**86.25**	***85.98***	0.0147
$V+D+NIR$	$dual_{\text{Custom,PTLL}}$	***95.81***	***95.80***	***95.54***	***85.61***	85.48	0.0159

(a) Trained on Freiburg dataset, prediction on test set

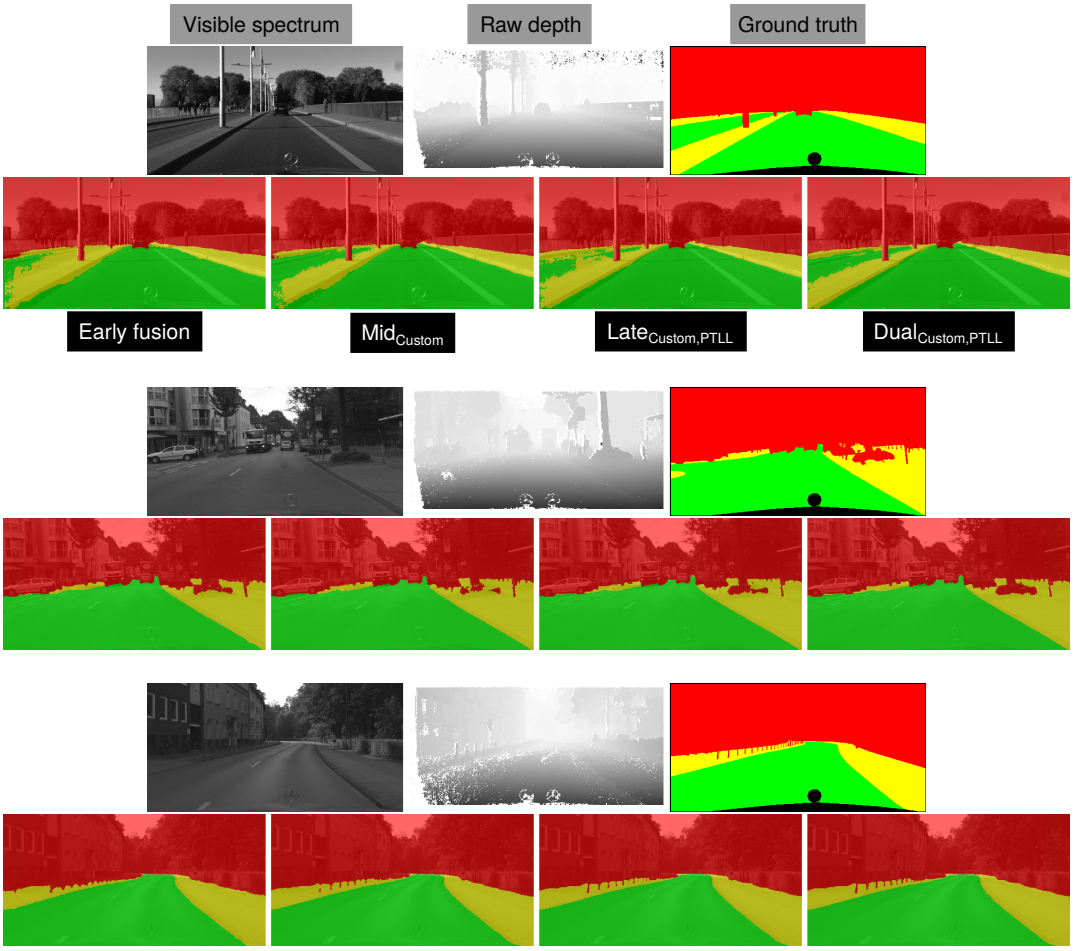
modalities	fusion	Pixel accuracy		IoU			mistake severity
		A	$A_{weighted}$				
V	-	96.91	97.24	98.02	67.94	93.76	0.1207
$V + D_{raw}$	early	96.60	97.00	97.61	65.22	93.33	0.1370
$V + D_{raw}$	mid_{SSMA}	97.00	***97.25***	98.21	68.34	93.80	0.1085
$V + D_{raw}$	mid_{Custom}	**97.22**	**97.70**	98.09	**70.23**	**94.54**	0.1287
$V + D_{raw}$	$late_{\text{SSMA}}$	96.94	97.14	**98.27**	67.98	93.66	**0.0935**
$V + D_{raw}$	$late_{\text{Custom}}$	96.82	96.95	98.14	67.28	93.42	0.1035
$V + D_{raw}$	$late_{\text{Custom,PTLL}}$	96.87	97.00	98.21	67.59	93.51	0.1011
$V + D_{raw}$	$dual_{\text{SSMA}}$	96.69	96.69	98.21	66.02	92.98	0.0983
$V + D_{raw}$	$dual_{\text{Custom}}$	96.95	97.10	***98.24***	67.91	93.70	***0.0971***
$V + D_{raw}$	$dual_{\text{Custom,PTLL}}$	***97.02***	97.22	***98.24***	***68.50***	***93.87***	0.1043
$V + D_{comp}$	early	96.74	97.14	97.67	66.38	93.59	0.1450
$V + D_{comp}$	mid_{SSMA}	96.98	97.30	***98.19***	68.35	93.79	0.1093
$V + D_{comp}$	mid_{Custom}	**97.13**	**97.52**	98.18	**69.30**	**94.21**	0.1140
$V + D_{comp}$	$late_{\text{SSMA}}$	96.88	97.04	98.15	67.88	93.53	0.1121
$V + D_{comp}$	$late_{\text{Custom}}$	96.92	97.17	98.11	68.31	93.63	0.1216
$V + D_{comp}$	$late_{\text{Custom,PTLL}}$	96.83	96.95	98.17	67.48	93.36	0.1110
$V + D_{comp}$	$dual_{\text{SSMA}}$	96.94	***97.35***	98.15	65.97	93.71	0.1233
$V + D_{comp}$	$dual_{\text{Custom}}$	***97.01***	97.29	**98.20**	***68.49***	***93.89***	**0.1059**
$V + D_{comp}$	$dual_{\text{Custom,PTLL}}$	96.97	97.17	***98.19***	68.37	93.75	***0.1107***

(b) Trained on Cityscapes dataset, prediction on test set

Table 8.2: Segmentation performance for different fusion configurations, compared to the visible spectrum (V) model from Section 6.2. We highlight the **best**, ***second-best*** and worst results for each metric/modality combination, and results which perform worse than the V and/or early fusion baseline. Results for the rows in gray are taken from the previous experiments in Section 8.1.

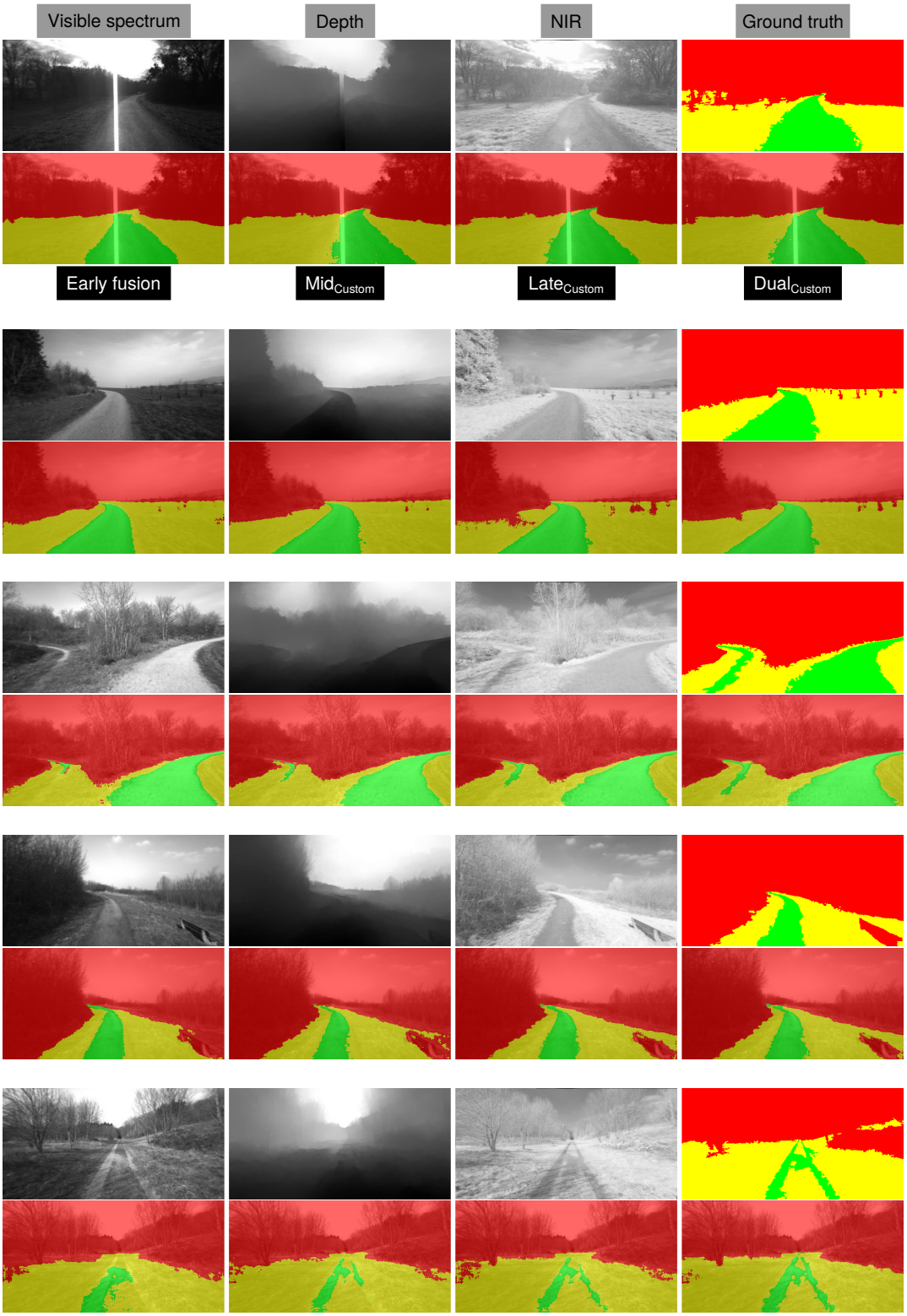
Deep fusion is a good idea On the Freiburg Forest dataset, the deep fusion models consistently out-perform the early fusion and single-modality baselines in terms of accuracy and IoU for the and levels. Compared to early fusion, the best performing deep fusion models increase accuracy and IoU for the level in the order of 1% , over 3% for the level and over 4% for the level. On Cityscapes, all deep fusion models also improve IoU for the level compared to the baselines. For some models however, this comes with a slightly poorer performance when distinguishing and areas, compared to the single-modality baseline - most likely due to stereo depth being a poor indicator of terrain type. Comparing the deep fusion variants, for each modality combination across both datasets, the best quantitative performance is generally achieved with a *mid* or *dual* architecture, indicating that fusing high-level features at the input of the decoder is beneficial

compared to having each branch up-sample modality-specific features independently. Figure 8.7 compares predictions by the 3 architecture variants and early fusion baseline. Upon visual inspection, we see clear improvements in segmentation quality brought by *dual* fusion.



(a) Bi-modal segmentation on Cityscapes

We note that the *mid* architecture achieves competitive results on both datasets, especially Cityscapes, whose images were captured in the most favorable illumination conditions. However, since this architecture only uses the pooling indices from the visible spectrum branch for up-sampling, it inherits some of the limitations of the visible spectrum model when faced with challenging lighting or scene elements which are difficult to delineate in this modality. This transpires in the first example of Figure 8.7b for instance, where the light ray causes poor segmentation of the left side of the path, or the second example, where the trees in the background blend with the grass. We would therefore expect a wider gap in performance between the *mid* and *dual*/*late* architectures in more challenging datasets.



(b) Tri-modal segmentation on Freiburg Forest

Figure 8.7: Qualitative comparison of the deep fusion variants (Custom units) and early fusion baseline on selected multi-modal samples of the Cityscapes and Freiburg Forest test sets.

Tri-modal prediction is worth it Looking at the performance of the $dual_{\text{Custom}}$ architectures for different modality combinations on the Freiburg dataset, the best results in terms of accuracy and IoU are achieved when segmenting the scene using all 3 modalities ($V + D + NIR$), followed by $V + NIR$, and lastly $V + D$. The benefit of incorporating all 3 modalities into the prediction through deep fusion can be seen in Figure 8.8. We note that in this dataset, the addition of NIR imaging brings a stronger improvement than depth, most likely because the distinction between path and surrounding grass is more clear in this modality (due to the high NIR reflectance of vegetation [114]), and as discussed in Section 7.1, the depth data was estimated from visible spectrum images and thus is not representative of true depth. However, as also found in the single-modality and early fusion experiments, results in Table 8.2 indicate that incorporating NIR data comes with a small mistake severity trade-off compared to the $V + D$ combination.

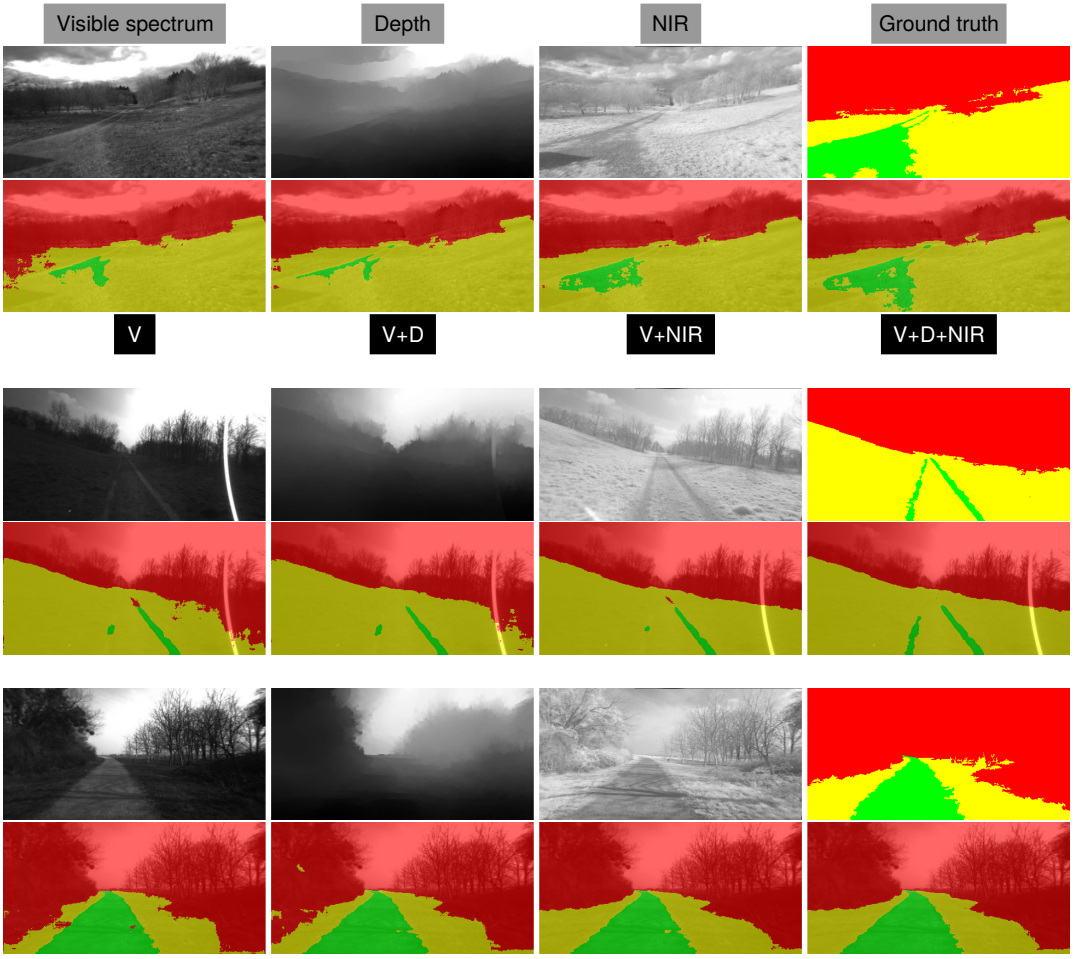


Figure 8.8: Predictions by the $dual_{\text{Custom,PTLL}}$ fusion model for different modality combinations on selected samples of the Freiburg Forest test set, compared to the single-modality baseline.

Depth completion conundrum Comparing the two modality combinations $V + D_{raw}$ and $V + D_{comp}$ for all fusion architectures on the Cityscapes dataset, almost all fusion models reach higher IoU for obstacles ■ with raw depth maps than completed ones, and for all metrics, the best scores (in bold) are higher with $V + D_{raw}$ than $V + D_{comp}$. This aligns with our comparison of single-modality models, where feeding the network raw depth maps resulted in superior segmentation performance, and contrasts with the early fusion model, for which completed depth maps improve segmentation across all metrics. This suggests that depth completion as a pre-processing step is mostly beneficial when the depth maps are being jointly convolved at the input of the network; when feature extraction is performed by a dedicated, modality-specific encoder, this step seems unnecessary at best, and detrimental at worst.

Figure 8.9 shows two examples in which we compare the fusion output for $V + D_{raw}$ vs. $V + D_{comp}$ modality combinations. Although the differences are subtle, for both architectures in the comparison (*dual* and *late*, which have a separate decoder for each modality), we notice a loss of detail with $V + D_{comp}$ fusion, such as the metal poles on the right in the first example, and the curved metal bar in front of the tree in the second example.

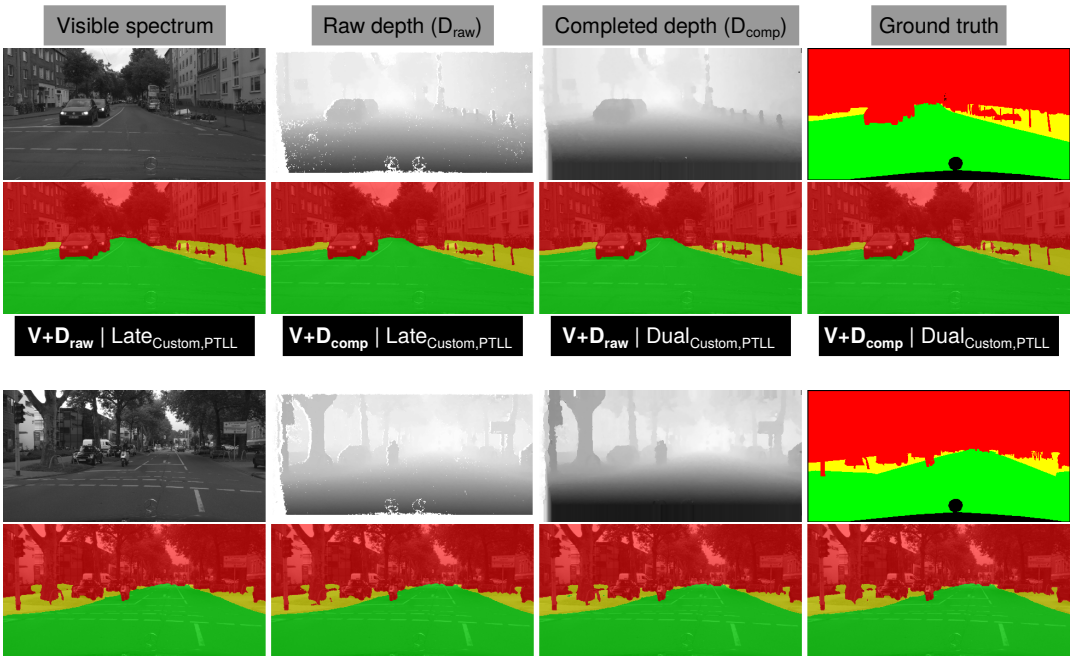
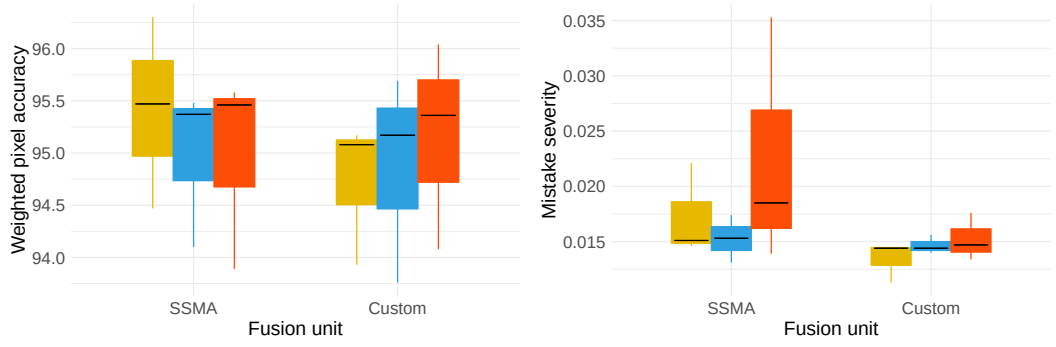
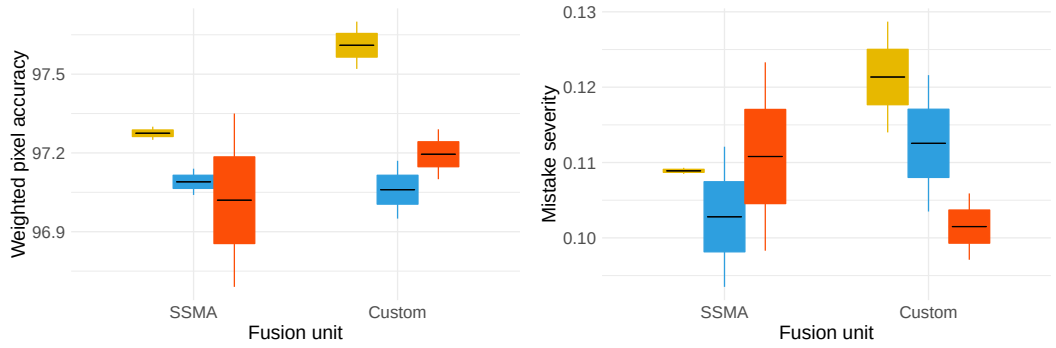


Figure 8.9: Predictions by the $late_{\text{Custom,PTLL}}$ and $dual_{\text{Custom,PTLL}}$ fusion models for the two bi-modal combinations $V + D_{raw}$ and $V + D_{comp}$ on selected samples of the Cityscapes test set.

SSMA vs. Custom fusion units Besides the difference in training speed and stability discussed in the beginning of this section, we note mixed differences in performance when fusing features with SSMA vs. Custom units. Figure 8.10 compares their performance in terms of weighted pixel accuracy and mistake severity on both datasets. Note that for fair comparison, we do not consider the Custom,PTLL models here. We note that Custom fusion models achieve more consistent performance across datasets and modality combinations than SSMA models which exhibit high variability (especially in the *dual* configuration eg. mistake severity on Freiburg Forest, and weighted accuracy on Cityscapes). We also note that *dual* models tend to reach higher pixel accuracy and lower mistake severity with Custom units (on Cityscapes, this difference in performance extends to all metrics); we attribute this to the fact that the *dual* model is the most difficult to train, having 2 fusion units which are jointly learned from scratch. Results suggest that the SSMA unit is best suited for mid-level or late fusion, but not both.



(a) Freiburg Forest - 3 modality combinations



(b) Cityscapes - 2 modality combinations

Figure 8.10: Distribution of weighted pixel accuracy scores (left) and mistake severity scores (right) across different modality combinations, comparing the performance of deep fusion models trained with Custom vs. SSMA fusion units.

The presence of 3 modalities in Freiburg Forest also gives us additional insight into how the fusion units cope with an increasing number of input streams. In Figure 8.11, we examine the models' performance for the different modality combinations in our evaluation. Comparing performance between the $V + NIR$ (bi-modal) and $V + D + NIR$ (tri-modal) combinations, the Custom models benefit from a clear improvement, while the SSMA models fall short in leveraging complementary features of a third modality. We attribute this to the difference in activation function in the SSMA vs. Custom fusion units: in SSMA, the weight for each element in the input is scaled independently across modalities with a sigmoid function, while in the Custom unit, a softmax function is applied across modalities. Thus, in the Custom fusion unit, for a given location in the feature map, low activation for one modality coincides with high activation for the other. Figure 8.12 shows two *dual* fusion predictions with SSMA vs. Custom fusion units.

Comparing the three architecture variants, the *mid* configuration performs best for a bi-modal input, but does not scale well to a third modality. In contrast, the performance of the *late* and *dual* architectures sees a clear improvement with a tri-modal input. This suggests that when incorporating additional modalities, modality-specific intermediate-level features should also be incorporated in the decoding stage.

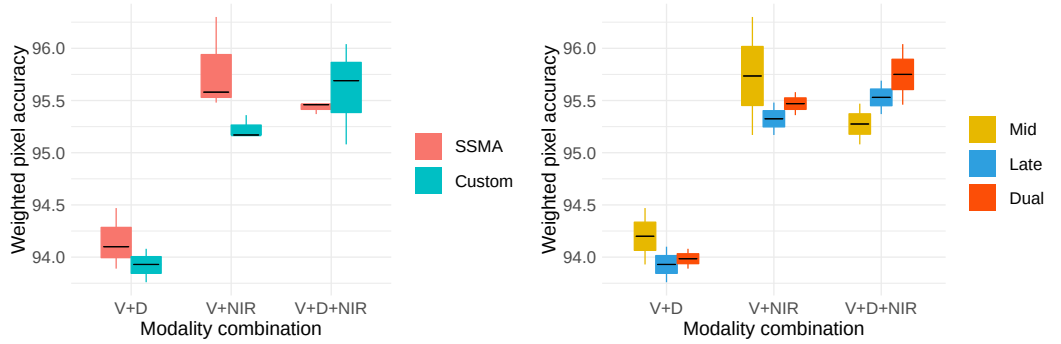


Figure 8.11: Distribution of weighted pixel accuracy scores across different modality combinations, comparing the performance of deep fusion models trained with Custom vs. SSMA fusion units (left), and with different architecture variants (right).

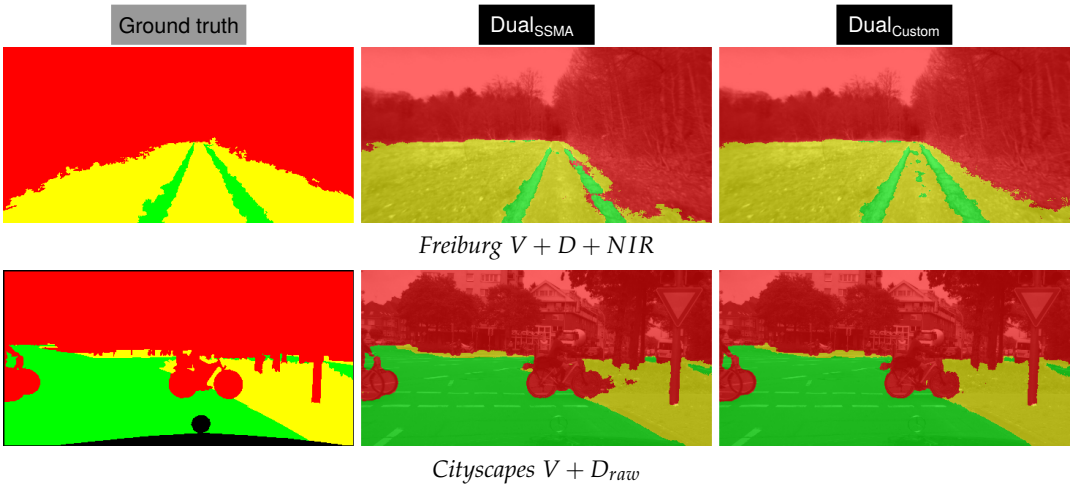
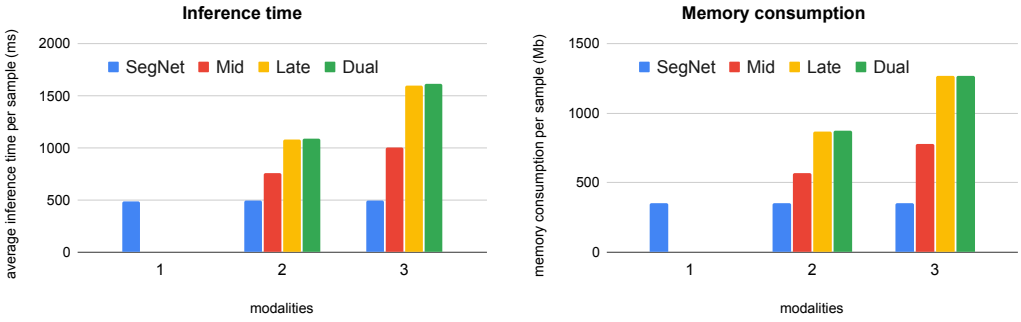


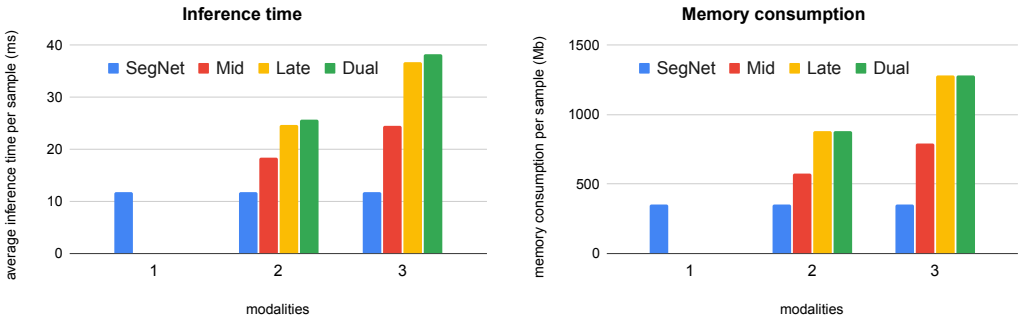
Figure 8.12: Predictions by the Dual fusion architecture trained with SSMA vs. Custom fusion units on selected samples of the Freiburg Forest and Cityscapes test set.

8.3 The slow elephant in the room

While its benefits are clear, the incorporation of additional modalities in the prediction through deep fusion comes with a computational cost, especially in a late or dual scheme, where each additional modality brings an additional encoder-decoder branch. To quantify this cost, we measure the average inference time and memory consumption of a forward pass for the different architectures in our experiment: SegNet (with one or more input channels), and the three deep fusion variants (Mid, Late and Dual). Benchmarking is performed both with multi-threaded CPU-only execution and with GPU-acceleration. Section A.5 provides implementation details, and we report results along with hardware specifications in Figure 8.13. As expected, channel stacking is the most time and memory efficient fusion method: the number of modalities at the input of the network can be increased at negligible computational cost. In contrast, each new modality-specific branch in the deep fusion architectures results in a near-linear increase in inference time and memory consumption.



(a) CPU-only: *Intel Core i7-6820HQ @ 2.70GHz × 8*



(b) GPU: *Tesla V100-SXM3-32GB*

Figure 8.13: Benchmarking results, comparing the computational footprint of the 3 deep fusion models (with Custom fusion units) vs. SegNet.

Chapter 9

Bringing it all together

In Chapter 6, we explore soft labelling and loss weighting training strategies for segmenting driveability levels in visible spectrum images, and in Chapter 8, we evaluate different multi-modal fusion configurations for segmentation.

However, in the previous experiments, we have performed training on specific datasets in isolation: the model learns to specialize in a particular type of scene (eg. backwoods paths in Freiburg Forest, structured roads in Cityscapes) with a consistent view-point, set of sensor characteristics etc. Thus, its performance may not be a good indicator of its ability to cope with diverse environments and captures. Therefore, in this chapter, we investigate whether a robust representation of driveability can be learned across a combination of different, more challenging datasets.

Furthermore, rather than applying our proposed methods separately, in this chapter, we select the most promising models from our previous experiments, and show how these training strategies can be combined and applied to a multi-modal architecture. Figure 9.1 illustrates our approach for combining the proposed methods, which we outline below

- Transfer learning scheme from Section 6.2, where the network first learns specific descriptive object classes from a semantic segmentation dataset, and then is adapted to learn a more functional representation by segmenting the scene in terms of 3 driveability levels.
- Soft ordinal labels from Section 6.3 for modelling a distance-based ranking between the driveability levels - during training, the network learns inter-class distances in order to make less severe mistakes
- Pixel-wise loss weighting from Section 6.4, to focus learning away from fine boundaries, and towards the bottom of the image
- Deep multi-modal fusion from Section 8.2, to leverage the complementary properties of different modalities

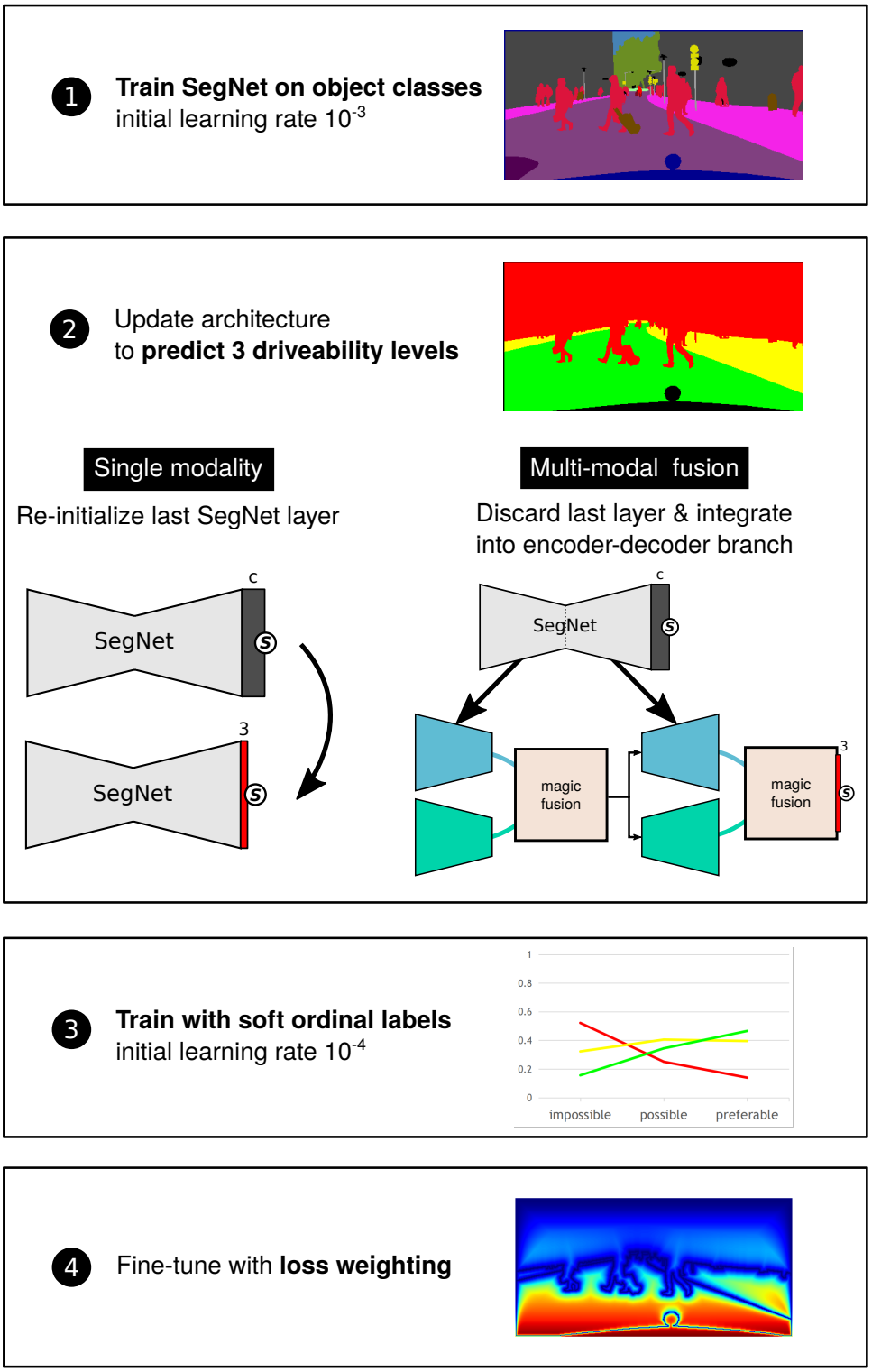


Figure 9.1: Overview of the training procedure combining the proposed methods to learn driveability from one or more modalities.

In Section 9.1, we explain our parameters and configurations of choice for each method, which we then apply in two experiments:

1. **Cross-dataset cocktail**: in Section 9.2, we train SegNet to segment driveability on a combination of visible spectrum images from 8 different datasets.

2. **Thermal fiesta**: in Section 9.3, we train a SegNet-based deep fusion architecture on a large-scale RGB-T dataset, and assess generalization to out-of-dataset samples. Since the thermal modality has not been included in the previous experiments, this is its time to shine.

9.1 Selected models

We base our choice of models on two main criteria: high IoU for areas which are impossible to drive on, and low mistake severity: from a navigation-oriented perspective, not confusing obstacles with the driveable path is crucial.

For generating soft labels, we pick the $SLD_{\alpha=1}$ scheme (as defined in Section 6.3.2), since it jointly achieves amongst the highest IoU for non-driveable areas and the lowest mistake severity (cf. the experimental results in Section 6.3.4). Note that this scheme yields the softest labels among the other inter-class distance definitions in our evaluation, resulting in smooth transitions from to pixels with a "buffer" of pixels around areas - as shown in Figure 9.2. While this deviates from what segmentation ground truth masks look like, we consider this beneficial for navigation, since it essentially adds a safe margin around obstacles.

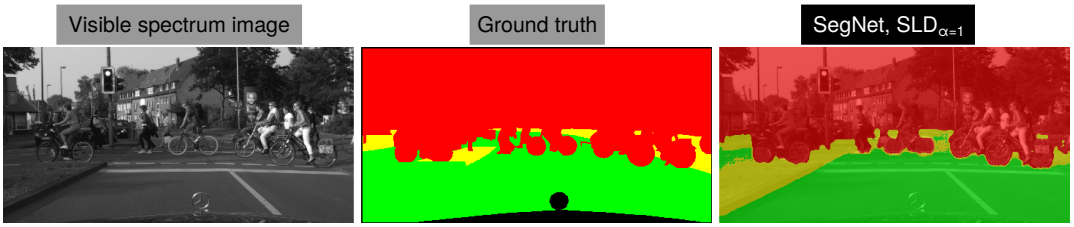


Figure 9.2: Example of a prediction by the $SLD_{\alpha=1}$ model from Section 6.3 on a Cityscapes test sample, compared to the ground truth mask. The prediction is shown overlayed on the input image.

As a deep fusion architecture, we pick the $dual_{\text{Custom}}$ variant from Section 8.2, which uses our Custom fusion unit to combine modality-specific feature representations both in the middle of the network, and after the decoding stage. Compared to the other fusion variants in our evaluation, it consistently achieves a good combination of low mistake severity and high IoU for non-driveable areas across the two datasets and different modality combinations.

9.2 No dataset left behind

In this section, we train a cross-dataset model, as an attempt to learn a more generic notion of driveability than in our previous single-dataset experiments. We show that our proposed driveability definition and training scheme can be used to learn pixel-level navigation affordances from a mix of diverse datasets with different class definitions and partial annotations.

9.2.1 Data

The 8 musketeers We make use of all 8 datasets in our Section 5.2 overview which contain pixel-level annotations for visible spectrum images: Freiburg Forest, Freiburg Thermal, Cityscapes, Kitti, Synthia, ThermalWorld, Lost & Found, MIR Multispectral. Note that while all these datasets except Freiburg Forest were collected in urban areas, they widely differ in their aspect ratio, image quality, annotation style/coarsity level, and each pose a unique set of challenges.

For instance, Synthia's images were generated in a simulation environment, thus its textures are not natural or realistic compared to the other datasets, and its captures are generally very dark due to shadows from high-rising buildings. Freiburg Thermal covers a wide range of driving scenarios and illumination conditions, and its annotations are approximate and blotchy. Lost & Found was specifically created to assess small-obstacle segmentation, and features unusual objects on the vehicle's path. MIR Multispectrual suffers from low image resolution, motion blur, and many of the roads are scattered with puddles and leaves. ThermalWorld VOC is perhaps the most challenging, as it was not captured from a vehicle, but from an unconstrained hand-held viewpoint along walkable areas in different cities and weathers, with many different close-range obstacles.

Combined dataset To gather a combined dataset for training, we randomly select 200 images from each of the 8 datasets (based on the number of the samples in the smallest dataset - Kitti only has 200): 180 from their training set and 20 from their validation set (see Section 5.2.1 for details on the dataset splits). This results in a total of 1440 training samples and 160 validation samples. We then evaluate the cross-dataset model on the datasets' individual test sets.

Ground truth Following the same procedure as in Section 6.2, we generate driveability ground truth data by mapping the original semantic labels of each dataset to driveability levels: ■ (roads and paths), ■ (other terrain, sidewalks), and ■ (sky, any obstacle) - see Section A.1 for the exact mapping. Figure 9.4 illustrates the resulting class distribution for each dataset and for the combined dataset. We also show a driveability sample from each dataset in Figure 9.3. Note that 3 datasets

are only partially annotated: for ThermalWorld VOC and MIR Multispectral, the model can only learn to segment obstacles, and Lost & Found only provides a very coarse outline of the driveable area.

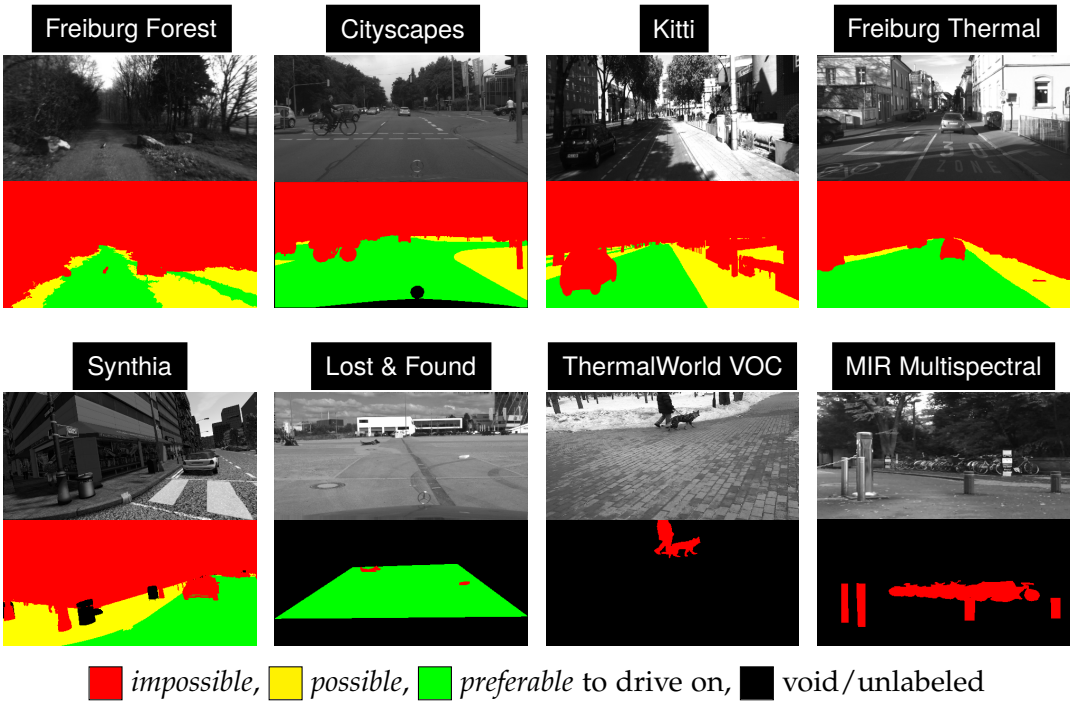


Figure 9.3: Sample from each dataset in the cross-dataset evaluation, with the visible spectrum image at the top, and the ground truth driveability segmentation at the bottom.

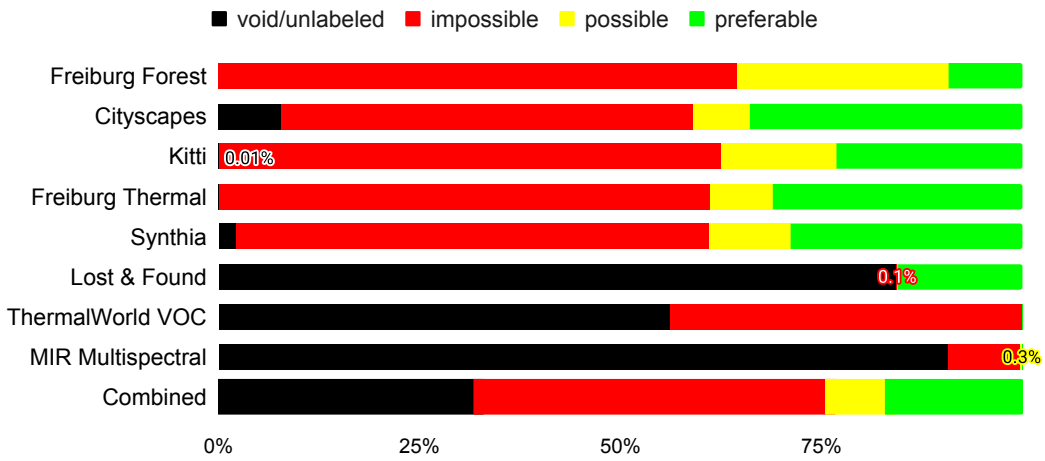


Figure 9.4: Proportion of pixels per driveability level for each (full) dataset in the cross-dataset evaluation, as well as in the combined dataset (200 samples from each of the 8 datasets).

9.2.2 Training procedure

We initialize SegNet with the pre-trained Cityscapes semantic model from Section 6.2 (since this is the largest dataset in our single-dataset experiments), and follow the transfer learning scheme to learn driveability levels from a combination of the 8 datasets. The model is trained on soft labels with $SLD_{\alpha=1}$ until convergence, with the same hyper-parameters as in Section 6.2. Similarly to our previous experiments, during training, samples are randomly augmented on the fly and fed to the model in shuffled batches - in this case, each batch contains samples originating from different datasets.

As illustrated in Figure 9.1, we then apply loss weighting as a final training stage for a few epochs, while maintaining the soft labelling scheme. To examine the effect of loss weighting, we compare the performance of the model before and loss-weighted training stage in our evaluation. Section A.6 shows examples of a loss weight map generated for each dataset in our evaluation.

9.2.3 Evaluation

Table 9.1 records the model's segmentation performance on unseen samples from the test sets of each dataset included during training. The table includes results from the single dataset experiments on Freiburg Forest and Cityscapes for comparison (taken from Table 6.2). We first generally comment on the cross-dataset model's performance on the different datasets, and then specifically compare the cross-dataset vs single dataset model performance for Freiburg and Cityscapes, and elaborate on the effect of loss-weighting. Lastly, we show cases where the model fails to correctly segment driveability from visible spectrum images alone.

Similarly to our earlier experiments in Section 6.2, the cross-dataset model's performance for each driveability level (in terms of IoU) reflects the class distribution: for all datasets where undriveable pixels constitute the majority class, the model achieves an IoU of over 90%. On Lost & Found where obstacles only constitute a small fraction of labelled pixels (cf. Figure 9.4), the model frequently mis-classifies them as driveable. We also note that the cross-dataset model has a tendency to confuse the road/path with other terrain, which manifests as generally low IoU scores for pixels. We attribute this to two main factors:

- the $SLD_{\alpha=1}$ soft labelling scheme introduces significant ambiguity between the and levels: mis-classifying one as the other during training has the lowest penalty compared to all other possible errors (cf. Figure 6.11).
- the distinction between the two levels is ambiguous in the labelling process and class definition itself, with conflicting labels across datasets: for instance, a subset of Cityscapes images feature square-paved roads (which are considered fully driveable), while walkable paths with a similar appearance

are labelled as sidewalk in Freiburg Thermal (considered only possible to drive on).

Note that MIR Multispectral only contains a minute fraction of pixels (cf. Figure 9.4) which all correspond to bumpy strips on the road - the model classifies them as fully driveable, hence the abysmal IoU score for this class.

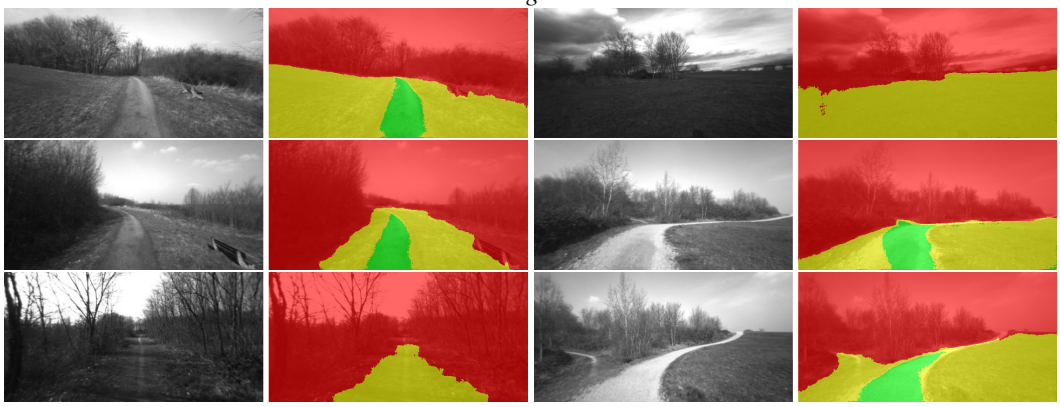
test set	Pixel accuracy		IoU			mistake severity
	A	$A_{weighted}$				
Freiburg Forest	**94.25**	**93.65**	94.26	**80.79**	**77.67**	**0.0010**
	93.55	***92.69***	**94.86**	***78.82***	***66.63***	***0.0049***
	93.26	92.12	***94.49***	77.60	66.35	0.0054
Cityscapes	**96.79**	**97.24**	**98.01**	**65.63**	**93.81**	0.0731
	94.93	94.56	97.51	53.51	89.27	***0.0724***
	95.20	***94.93***	***97.52***	***54.44***	***90.06***	**0.0712**
Kitti	93.25	93.65	**94.27**	**61.40**	85.32	**0.0621**
	93.36	**93.85**	94.21	60.96	**86.11**	0.0654
Synthia	91.61	89.93	**91.09**	54.64	81.70	**0.3720**
	92.01	**90.38**	91.07	**57.65**	**82.34**	0.4272
Freiburg Thermal	92.43	91.94	**95.60**	43.82	81.97	**0.1478**
	92.71	**92.26**	95.59	**44.80**	**82.74**	0.1601
Lost and Found	89.58	91.92	**41.16**	-	89.69	**0.0481**
	91.20	**93.38**	39.19	-	**91.31**	0.0603
MIR Multispectral	97.53	97.08	97.66	9.48	-	**0.0587**
	97.55	**97.33**	**97.69**	**9.49**	-	0.0805
ThermalWorld VOC	99.26	98.67	99.26	-	-	0.1249
	99.62	**99.44**	**99.62**	-	-	**0.0967**

Single dataset Cross-dataset Cross-dataset, LW

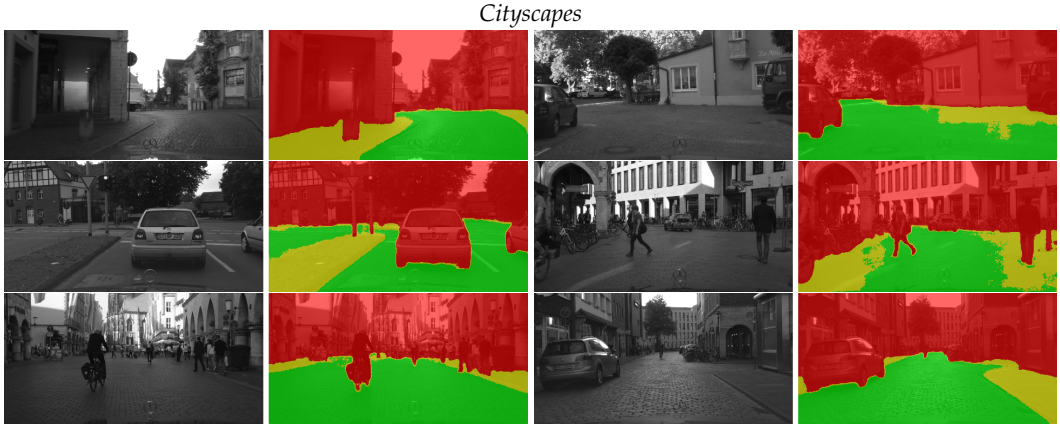
Table 9.1: Quantitative results for the SegNet model trained on visible spectrum images with SORD labels ($SLD_{\alpha=1}$) from a combination of 8 datasets (180 training sample from each). We record performance on the test set of each dataset separately, before and after the loss-weighting (LW) training stage. For reference, we also include the results of the single-dataset models from Section 6.3.4 (Cityscapes and Freiburg) and highlight them in gray. We highlight the **best** and ***second best*** results.

Figure 9.5 shows examples of predictions by the loss-weighted cross-dataset model (green in Table 9.1) on each test set in our evaluation.

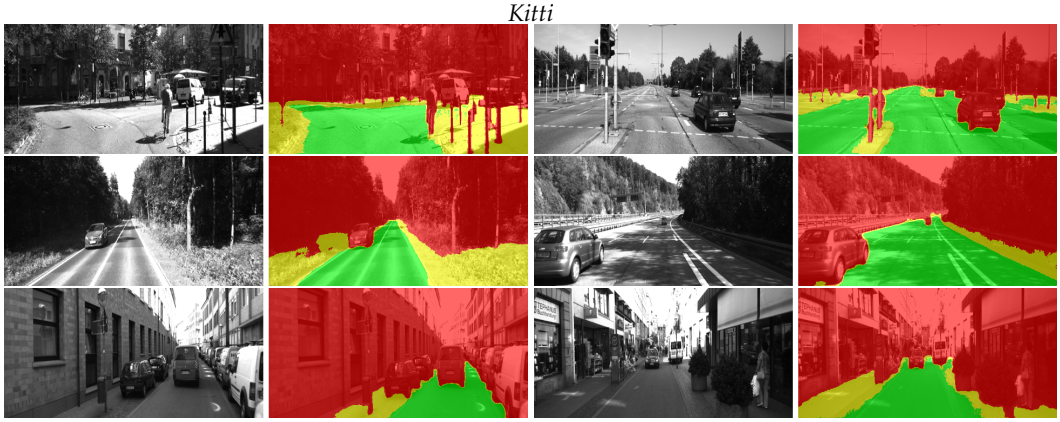
Freiburg Forest



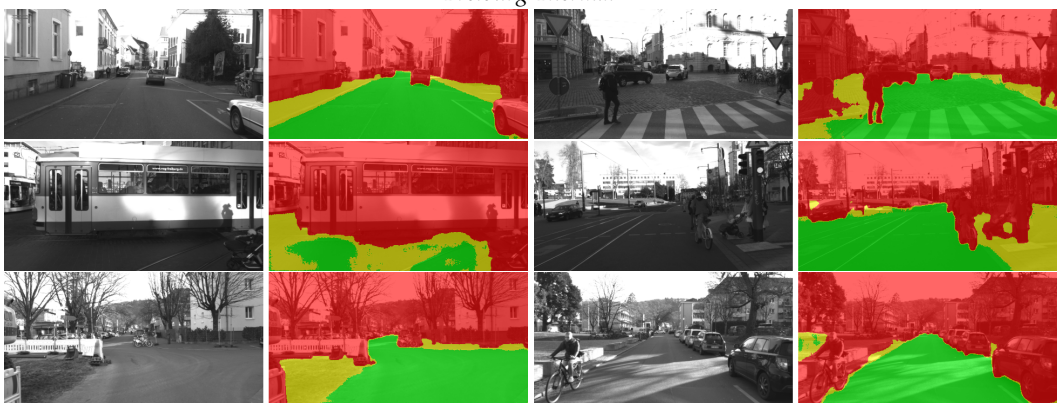
Cityscapes



Kitti



Freiburg Thermal



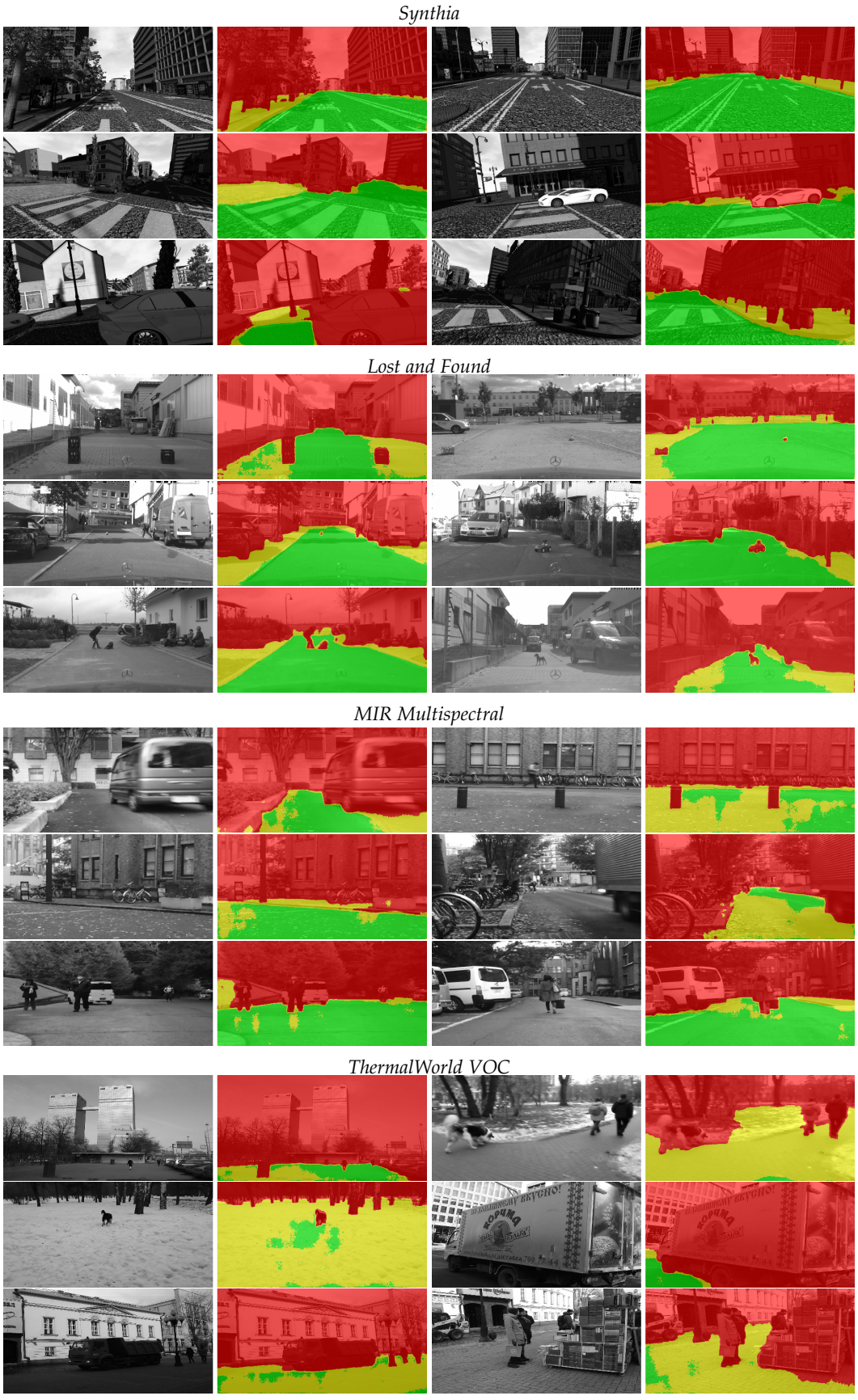


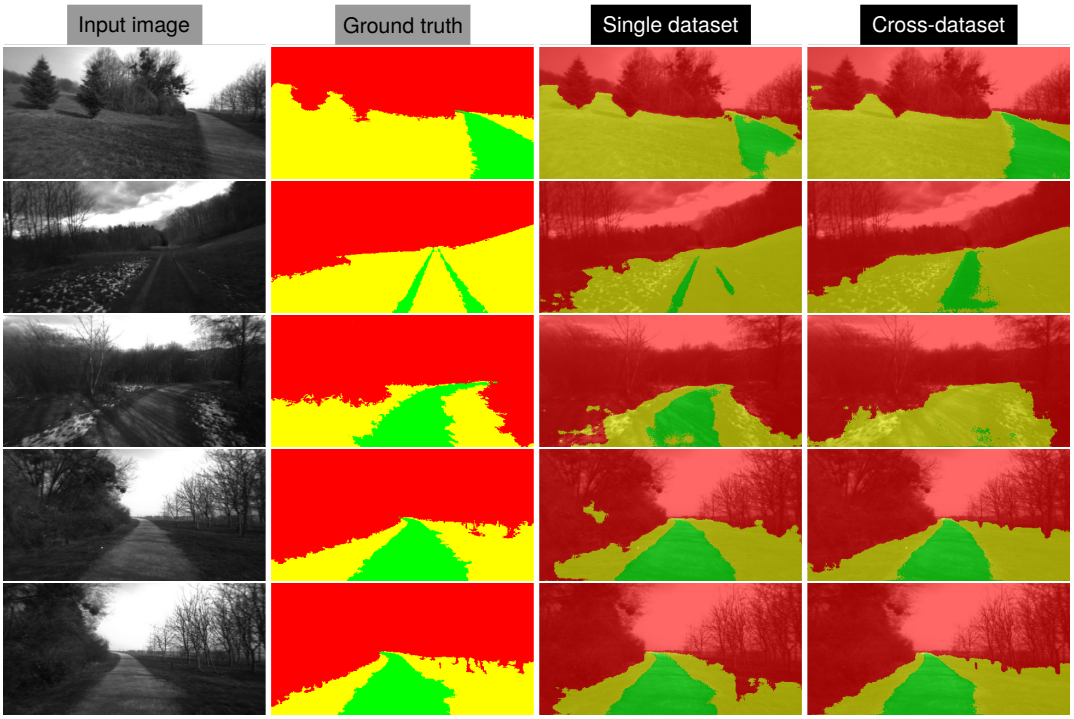
Figure 9.5: Prediction by the cross-dataset model after loss-weighted (LW) training on selected samples from individual test sets.

Single- vs cross-dataset learning

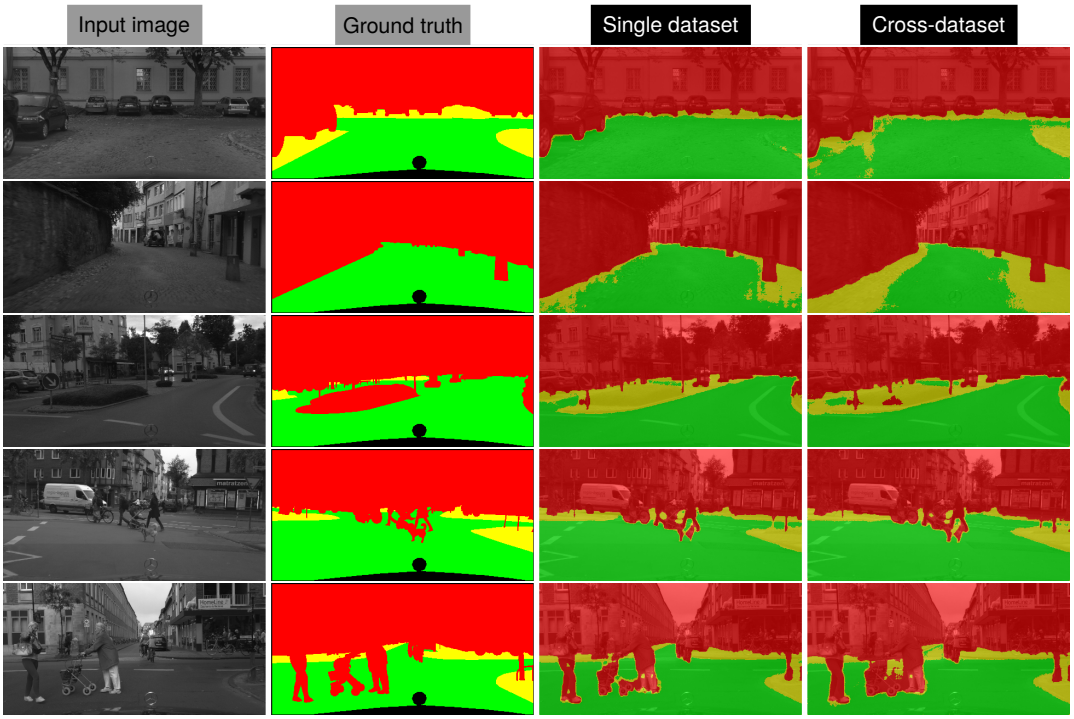
Figure 9.6 compares predictions by the cross-dataset model (blue in Table 9.1) with the single-dataset models from our soft labelling experiment in Section 6.3.4, where the model is trained and evaluated separately on the Freiburg and Cityscapes dataset (gray in Table 9.1).

Looking at quantitative metrics, cross-dataset learning results in a drop in accuracy on both test sets, primarily across the two driveable levels and . During single-dataset training, the model learns to recognize the specific features of the paths in Freiburg Forest for instance, which are quite different from the paved streets of the other datasets included in the cross-dataset training. Thus, in some cases, cross-dataset learning results in a fuller path (eg. the first two rows in Figure 9.6), since the model is accustomed to wide driveable areas from urban scenes - however, when the path is too dissimilar from the paved roads in the other datasets, it is wrongly categorized as possible rather than preferable to drive on. On Cityscapes, road areas with irregular terrain are more likely to be classified as when learning cross-dataset representations.

However, as can be seen in Figure 9.6's visual comparison, cross-dataset learning results noticeably improves segmentation of obstacles: in Freiburg Forest, trees are more clearly delineated and in Cityscapes, previously under-segmented objects are recovered (at the expense of precision, hence the lower IoU score).



(a) Freiburg Forest



(b) Cityscapes

Figure 9.6: Qualitative comparison of predictions by single-dataset models on Freiburg Forest and Cityscapes with the cross-dataset model (both trained without loss weighting).

Overall, learning a variety of objects from other datasets seems to be beneficial for recognizing undriveable areas, however the notion of a driveable path is more ambiguous depending on the type of scene and annotation.

Effect of loss weighting

Comparing the results of the cross-dataset model trained with uniformly weighted loss (blue in Table 9.1) and after applying loss weighting (green), we note that loss weighting improves accuracy on every dataset except Freiburg Forest. Figure 9.7 shows predictions by both models for qualitative comparison. Similarly to our single dataset experiment from Section 6.4, we find that loss weighting yields a slightly smoother and more cohesive segmentation. We especially find that it improves recall for ■ areas low in the image (at the expense of precision) - e.g. the traffic sign, baby stroller and dog's head in Figure 9.7. The increase in mistake severity on most test sets is likely due to over-segmentation of obstacles or loss of detail in distant areas: loss weighting greatly improves mistake severity on ThermalWorld VOC, which contains many obstacles captured at close range.

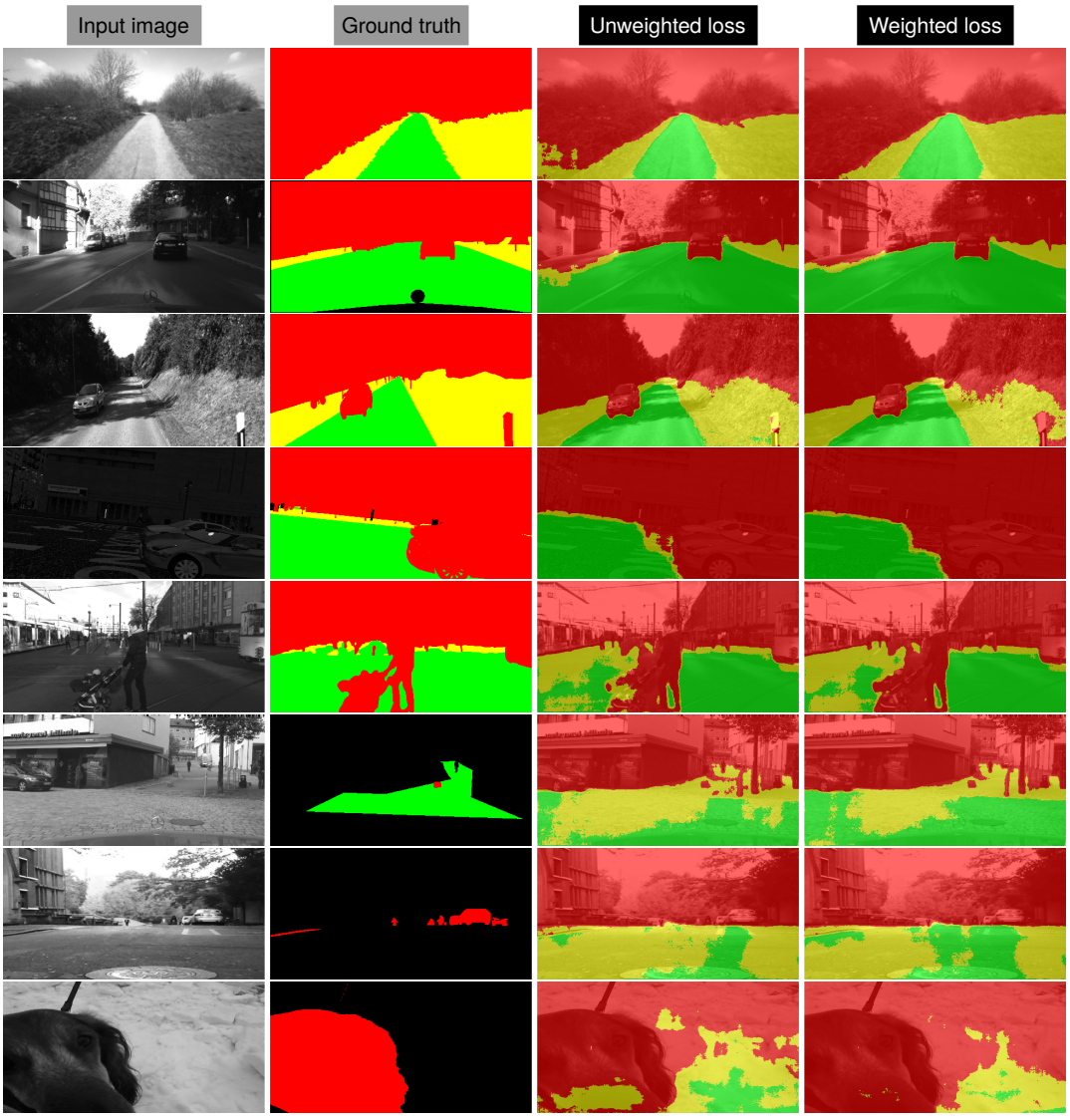


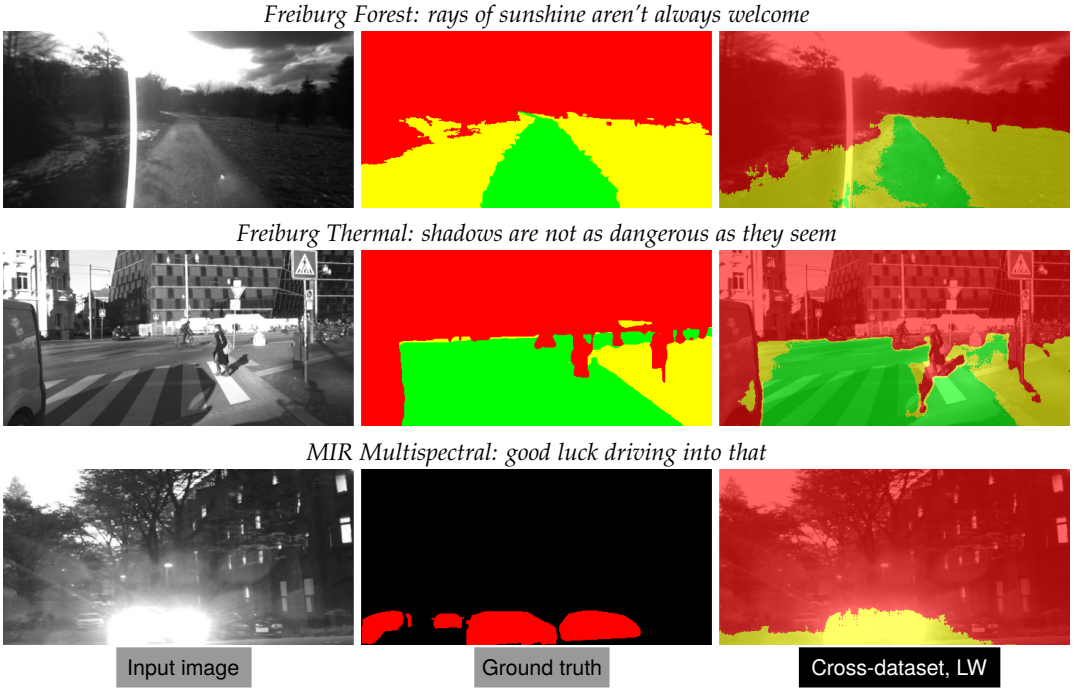
Figure 9.7: Comparison of predictions by the cross-dataset model, trained with and without loss weighting, with a sample from each of the test sets in Table 9.1 (shown in the same order).

Failure cases

We note that the model can be prone to severe mistakes when confronted with challenging illumination conditions or unexpected obstacles on the road - Figure 9.8 shows examples for both cases.

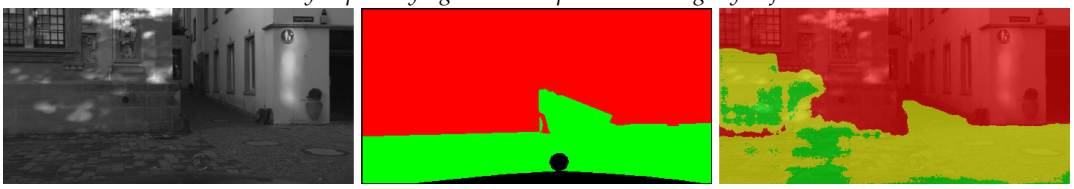
For large obstacles in close vicinity to the vehicle, this can partly be explained by the fact that in most images seen during training, obstacles appear starting at a reasonable distance from the vehicle. A contributing factor to these errors might also be that due to the front of the ego-vehicle being visible at the bottom of the image in Cityscapes images (which the model was pre-trained on, and which constitute 1/8 of training samples), the model learns to ignore object features in its immediate vicinity. In addition, some surfaces like smooth concrete barriers or walls are difficult to distinguish from the road or sidewalk when relying on grayscale visual information alone.

The poor IoU score for pixels on the Lost & Found dataset (where all the pixels in this class correspond to small road obstacles) also raises a key contradiction faced by the model during learning: the model is encouraged to ignore road irregularities such as potholes, shadows, lane markings (since all of these are considered driveable), yet should still be able to identify anomalies and potential hazards on the vehicle's path. Distinguishing the two can be tricky when perception is limited to monocular vision.

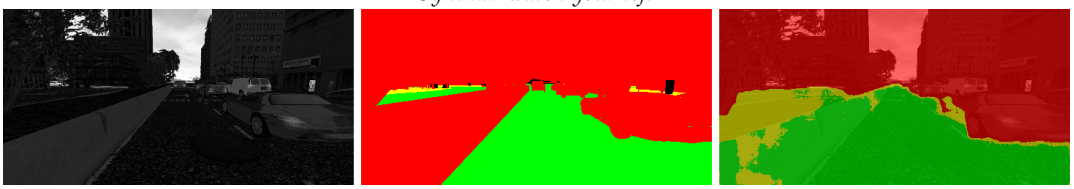


(a) Shadows and lights confuse the best of us

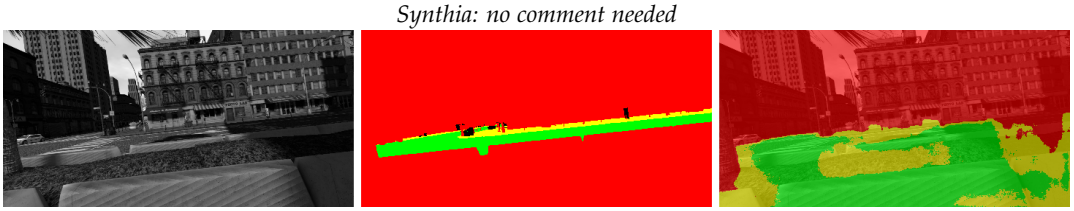
Cityscapes: trying to climb up walls won't get you far



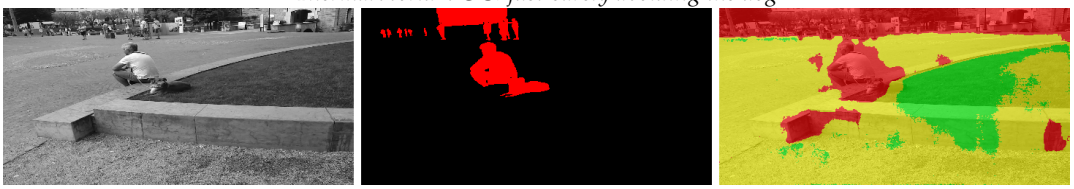
Synthia: watch your left



Synthia: no comment needed

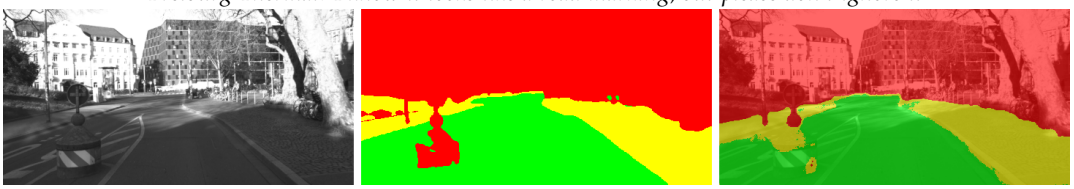


ThermalWorld VOC: just barely avoiding the dog

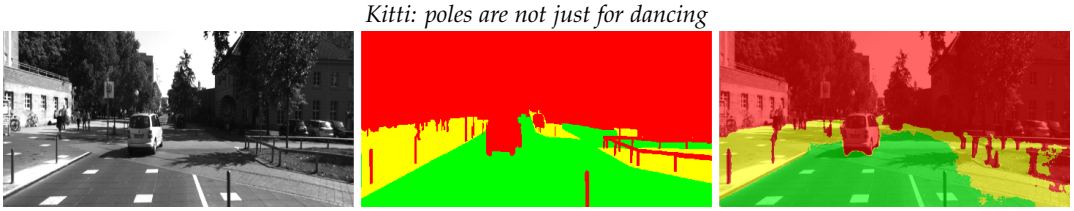


(b) Overlooking large obstacles

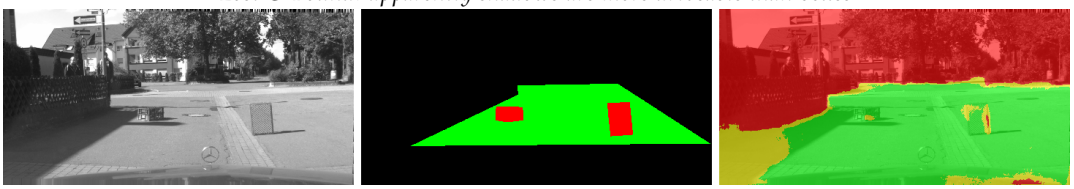
Freiburg Thermal: I know it looks like a road marking, but please don't ignore it



Kitti: poles are not just for dancing



Lost & Found: apparently shadows are more driveable than boxes



(c) Overlooking smaller ones

Figure 9.8: Some examples of unacceptable segmentation results by the cross-dataset model (trained with loss weighting), with at least one sample from each of the test sets in Table 9.1.

These failures highlight the importance of incorporating additional modalities in the model's prediction: depth information has shown to improve segmentation of obstacles (both in our deep fusion experiments from Section 8.2 and in existing literature [108, 101]), while thermal images provide robustness to variations in illumination (we fuse visible spectrum and thermal imaging in the next section). These modalities can help distinguish between driveable surfaces and real obstacles. The addition of color information may also be beneficial to accentuate differences between the road and scene elements which should be avoided.

9.2.4 Bonus content: driveability in the wild

To further display the model's ability to predict driveability in a wide range of settings, we show segmentation results on hand-held pictures taken in diverse locations and conditions in Figure 9.9. Note that the model's training samples do not feature any images captured during night-time or under snow-fall, yet it is able to segment obstacles correctly in these examples.

Figure 9.9: Spot the kitties

9.3 No modality left behind

As a final experiment, we show that the proposed deep fusion architecture can be trained with the soft labelling and loss-weighting scheme to predict driveability from a combination of visible spectrum and thermal images. For training, since this is the only large-scale RGB-T dataset with full-image pixel-level annotations, we make use of the Freiburg Thermal dataset, introduced in Section 5.1.2. Note that this is also a much larger dataset than in the experiments from previous Chapters (12k+ samples vs. less than 200 for Freiburg Forest and around 3.5k for Cityscapes).

The goal of this experiment is not only to record segmentation performance on unseen samples from Freiburg Thermal, but also to assess the model's ability to generalize to out-of-dataset captures. Therefore, for evaluation, we also make use of partially annotated RGB-T datasets from Section 5.1: ThermalWorld VOC and MIR Multispectral. Since the pixel labels in these two datasets almost exclusively correspond to the ■ level (eg. person, building, car - cf. Figure 9.4), we additionally manually annotate 55 images from the KAIST Multispectral Pedestrian Detection dataset (introduced in Section 5.1.2). We randomly select 5 samples from each day-time sequence in the test set (cf. the benchmark's data description[1]). Image pairs are annotated with coarse, full-image, pixel-level class labels for the 3 driveability levels. Our annotated samples are made publicly available[2]. We show a driveability sample from each of these 4 datasets in Figure 9.10.

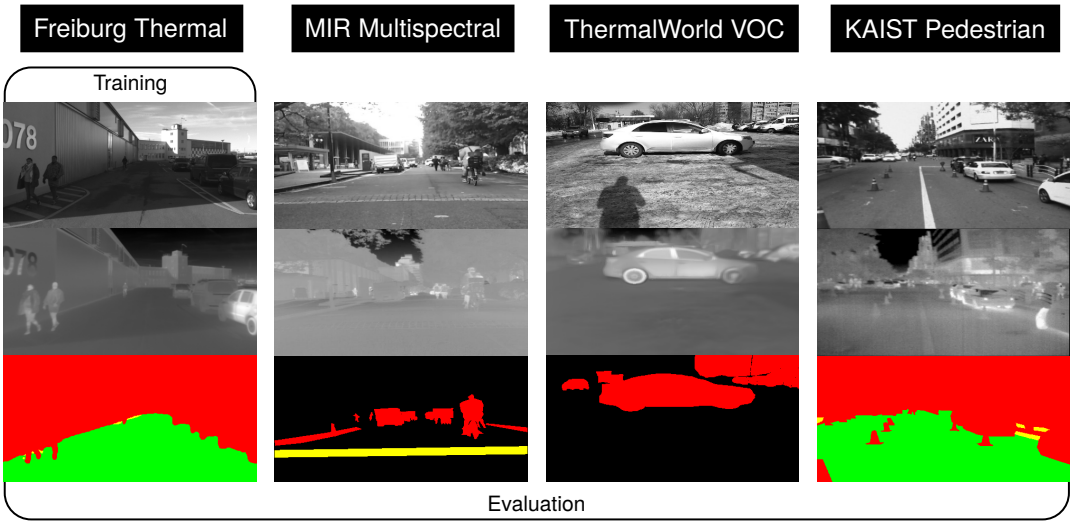


Figure 9.10: Example of an input pair (visible spectrum and thermal image) and ground truth driveability segmentation from the 4 datasets in our thermal fusion experiment.

[1] https://soonminhwang.github.io/rgbt-ped-detection/data/

[2] https://www.kaggle.com/glhr00/rgbt-driveability-segmentation-kaist

9.3.1 Training procedure

Pre-trained models As a starting point (step 1 in Figure 9.1), we train two single-modality SegNet models from scratch to segment object classes in visible spectrum and thermal images from the Freiburg Thermal dataset (13 semantic classes). We use the same standard training procedure as in Section 6.2, with one-hot labels and no loss weighting. We refer to these two models as $V_{\text{FreiburgThermal}}$ (visible spectrum model) and $T_{\text{FreiburgThermal}}$ (thermal model). We also make use of the SegNet model trained on Cityscape's object classes in Section 6.2, which we refer to as $V_{\text{Cityscapes}}$.

Baseline model Following the transfer learning scheme, we use the pre-trained $V_{\text{FreiburgThermal}}$ model as a basis to learn driveability from visible spectrum images alone (step 2 in Figure 9.1). Similarly to the previous experiment, training is performed with the $SLD_{\alpha=1}$ soft labelling scheme. We take this as our baseline, to compare against deep fusion.

Fusion models Step 3 in Figure 9.1 illustrates how we incorporate pre-trained SegNet models into the $dual_{\text{Custom}}$ fusion architecture. The visible spectrum and thermal branches are initialized with the weights from the encoder and decoder of their respective pre-trained SegNet models: only the prediction layer of each SegNet model is discarded. To investigate the effect of initializing the deep fusion architecture with pre-trained weights from a different dataset, we train two fusion models:

- $V_{\text{FreiburgThermal}}$ + $T_{\text{FreiburgThermal}}$, where the network is initialized with the SegNet models which were both pre-trained on Freiburg Thermal's object classes
- $V_{\text{Cityscapes}}$ + $T_{\text{FreiburgThermal}}$ where the visible spectrum branch is initialized with the SegNet model pre-trained on Cityscapes' object classes instead

In order to learn driveability levels, the fusion architecture's prediction layer is randomly initialized with 3 output channels, and we then train the network end-to-end with the $SLD_{\alpha=1}$ soft labelling scheme. In addition, similarly to the previous experiment, we apply loss weighting as a final training phase, and record the fusion model's performance both before and after applying loss weighting to evaluate its effect.

Data augmentation Based on the results from Section 7.1 which suggest that photometric data augmentation is beneficial for segmentation of NIR images, we similarly apply both photometric and geometric transformations to thermal images during training. However, considering the fundamental differences between NIR and LWIR imaging (cf. Section 2.1.2), the choice of data augmentation for the thermal modality should be investigated further.

9.3.2 Evaluation

Table 9.2 records performance of the baseline (in blue) and deep fusion models (green and yellow) on Freiburg Thermal's test set as well as the three other RGB-T datasets, before and after the loss-weighting (LW) training stage. Similarly to the results of our previous experiment, the level is the most challenging to segment correctly, while all models achieve an IoU of over 90% for undriveable areas.

	pre-trained model per modality	architecture & training	Pixel accuracy A	Pixel accuracy $A_{weighted}$	IoU	IoU	IoU	mistake severity
Trained on Freiburg Thermal dataset, prediction on Freiburg Thermal test set - 1115 samples								
	$V_{\text{FreiburgThermal}}$	SegNet	95.61	95.93	96.96	61.07	90.25	**0.1514**
+	$V_{\text{FreiburgThermal}}$	$dual_{\text{Custom}}$	**95.75**	***96.07***	***97.04***	**62.80**	***90.33***	0.1691
	$T_{\text{FreiburgThermal}}$	$dual_{\text{Custom}}$,LW	***95.74***	***96.07***	**97.05**	***62.37***	***90.33***	0.1677
+	$V_{\text{Cityscapes}}$	$dual_{\text{Custom}}$	95.61	95.96	96.88	61.29	90.18	0.1674
	$T_{\text{FreiburgThermal}}$	$dual_{\text{Custom}}$,LW	95.72	**96.13**	96.97	61.57	**90.45**	***0.1664***
Trained on Freiburg Thermal dataset, prediction on MIR Multispectral - 820 samples								
	$V_{\text{FreiburgThermal}}$	SegNet	87.72	84.61	90.30	3.47	-	0.3711
+	$V_{\text{FreiburgThermal}}$	$dual_{\text{Custom}}$	***89.73***	***87.43***	***92.30***	***5.15***	-	0.3507
	$T_{\text{FreiburgThermal}}$	$dual_{\text{Custom}}$,LW	**89.90**	**87.76**	**92.43**	**5.83**	-	0.3653
+	$V_{\text{Cityscapes}}$	$dual_{\text{Custom}}$	89.43	87.05	92.07	3.49	-	**0.2689**
	$T_{\text{FreiburgThermal}}$	$dual_{\text{Custom}}$,LW	89.40	86.90	92.17	2.01	-	***0.2973***
Trained on Freiburg Thermal dataset, prediction on ThermalWorld VOC - 1466 samples								
	$V_{\text{FreiburgThermal}}$	SegNet	90.38	85.92	90.38	-	-	0.2763
+	$V_{\text{FreiburgThermal}}$	$dual_{\text{Custom}}$	**96.77**	**95.54**	**96.77**	-	-	**0.2107**
	$T_{\text{FreiburgThermal}}$	$dual_{\text{Custom}}$,LW	96.57	***94.92***	96.57	-	-	***0.2340***
+	$V_{\text{Cityscapes}}$	$dual_{\text{Custom}}$	95.61	94.38	95.61	-	-	0.2500
	$T_{\text{FreiburgThermal}}$	$dual_{\text{Custom}}$,LW	***95.65***	94.19	***95.65***	-	-	0.2970
Trained on Freiburg Thermal dataset, prediction on KAIST Pedestrian - 55 samples								
	$V_{\text{FreiburgThermal}}$	SegNet	**94.42**	**96.08**	93.46	**37.70**	**92.18**	0.2582
+	$V_{\text{FreiburgThermal}}$	$dual_{\text{Custom}}$	***93.74***	***95.15***	93.56	***34.11***	***90.92***	0.1953
	$T_{\text{FreiburgThermal}}$	$dual_{\text{Custom}}$,LW	93.68	95.03	93.67	33.63	90.65	0.1928
+	$V_{\text{Cityscapes}}$	$dual_{\text{Custom}}$	91.84	92.55	***93.69***	27.29	86.73	**0.1233**
	$T_{\text{FreiburgThermal}}$	$dual_{\text{Custom}}$,LW	92.86	94.02	**93.93**	29.98	88.66	***0.1514***

Table 9.2: Quantitative results for the SegNet (visible spectrum) and $dual_{\text{Custom}}$ (visible spectrum and thermal) models trained with SORD labels ($SLD_{\alpha=1}$) on the Freiburg Thermal dataset. We highlight the **best** and ***second best***, and worst results.

The following sections specifically examine the effect of incorporating thermal images with deep fusion, network initialization and loss weighting.

Benefit of fusion

We first compare the performance of the single-modality V model and the V+T deep fusion model, both pre-trained on Freiburg Thermal (first two rows in Table 9.2). Looking at the IoU for undriveable areas, the fusion model outperforms the visible spectrum model on all 4 datasets. KAIST Pedestrian is the only dataset in our evaluation where fusion does not improve accuracy for driveable areas (and pixels) - it seems that the lower quality thermal images in this dataset causes confusion between different terrain types.

While quantitative differences between the two models are rather minor on Freiburg Thermal (which the model is trained on), we see noticeable differences when looking at the segmentation output. Figure 9.11 shows examples from Freiburg Thermal's test set where fusion improves segmentation compared to the visible spectrum model, by recovering the outline of important obstacles which should be avoided. Note that this dataset's ground truth masks are very approximate (they seem to have been estimated from visible spectrum images), with frequently inaccurate outlines of objects (e.g. the traffic pole in the first row of Figure 9.11 or people's legs in the second row) - in many cases, the output of the fusion model is more accurate and cohesive than the ground truth. Hence, the quantitative metrics reported for this dataset may not be representative of true performance.

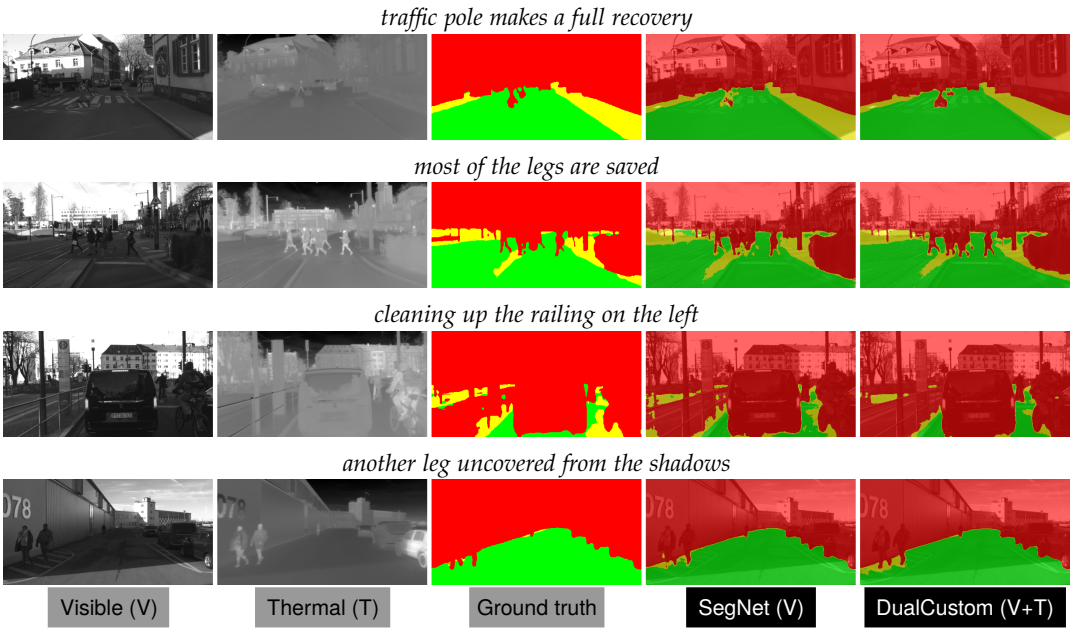
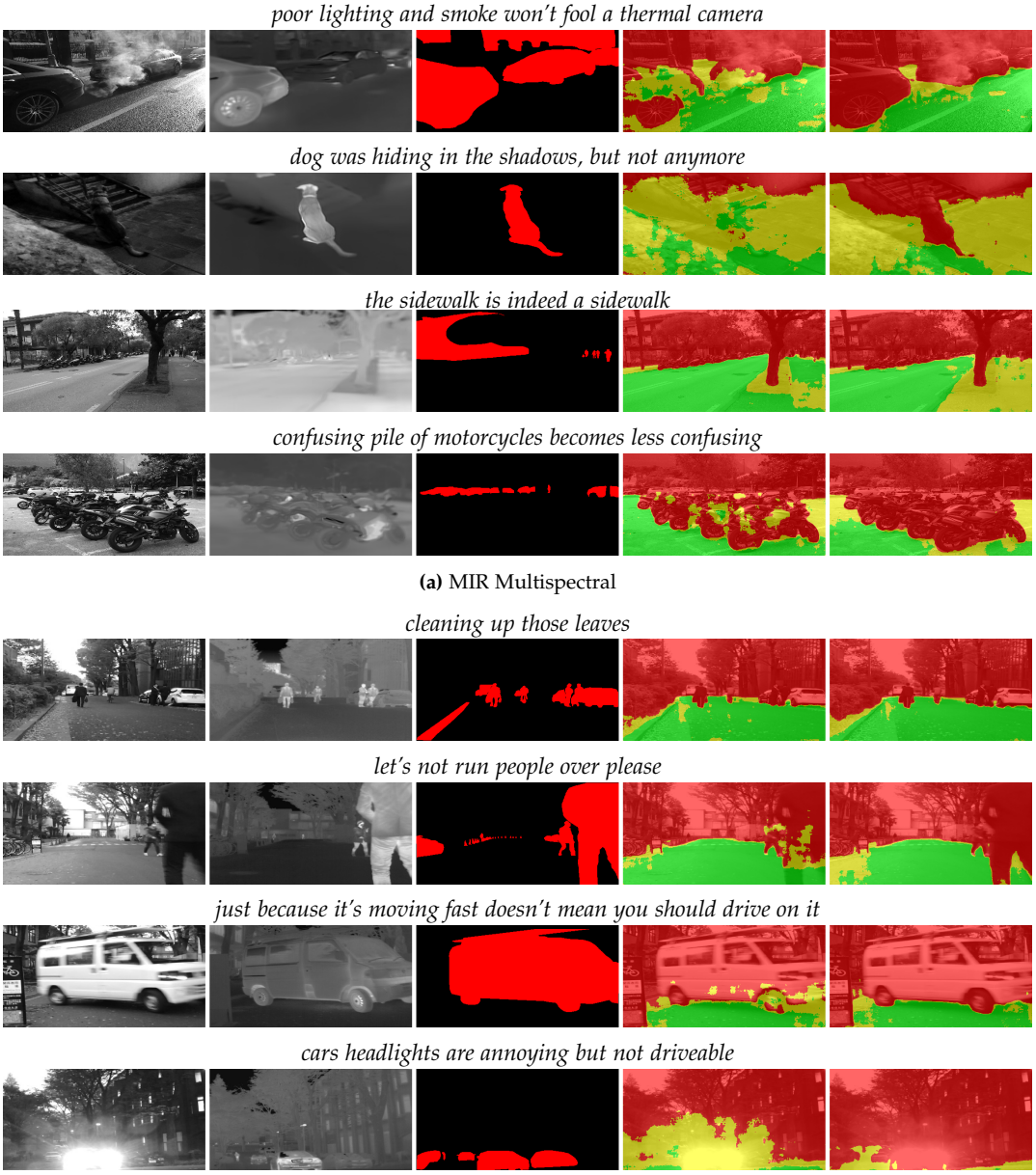


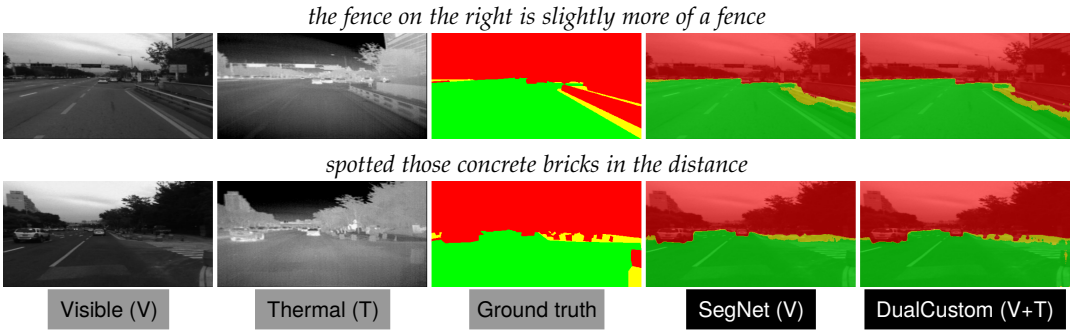
Figure 9.11: Prediction on samples from the Freiburg Thermal test set, comparing deep fusion (V+T) with the single-modality (V) model (first two rows in Table 9.2).



(a) MIR Multispectral

(b) ThermalWorld VOC

The performance gap between the two models significantly widens on out-of-dataset samples, especially on ThermalWorld with an improvement in accuracy of over 6% (and over 9% in weighted accuracy) when fusing modalities. Considering the challenging nature of this dataset, these results suggests that incorporating thermal imaging aids generalization to new scene elements and viewpoints. Figure 9.12 shows predictions on out-of-dataset samples: the visible spectrum model frequently makes severe mistakes which are corrected by visible-thermal fusion.



(c) KAIST Pedestrian

Figure 9.12: Prediction on out-of-dataset samples from the 3 datasets in our evaluation, comparing deep fusion (V+T) with the single-modality (V) model (first two rows in Table 9.2).

Initialization matters

Comparing the two fusion models ($V_{\text{FreiburgThermal}}$ + $T_{\text{FreiburgThermal}}$ and $V_{\text{Cityscapes}}$ + $T_{\text{FreiburgThermal}}$) which differ in the pre-trained model used to initialize the visible spectrum encoder-decoder branch, the first converges faster during training and achieves higher accuracy across all 4 datasets. However, quantitative differences in performance between the models are quite small, both on Freiburg Thermal and out-of-dataset samples. In fact, we find frequent cases where the fusion model initialized with $V_{\text{Cityscapes}}$ yields more coherent segmentation, a few of which are shown in Figure 9.13.

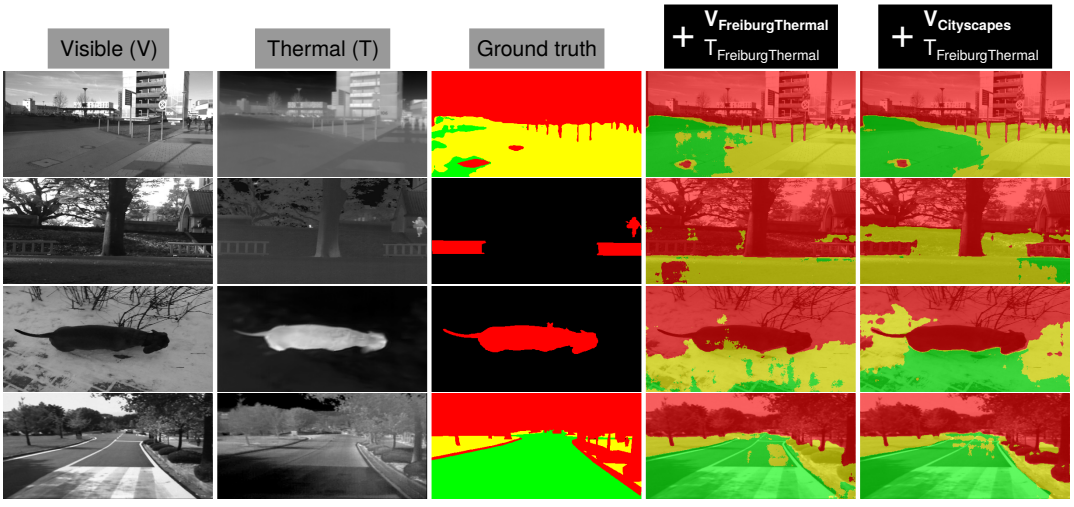
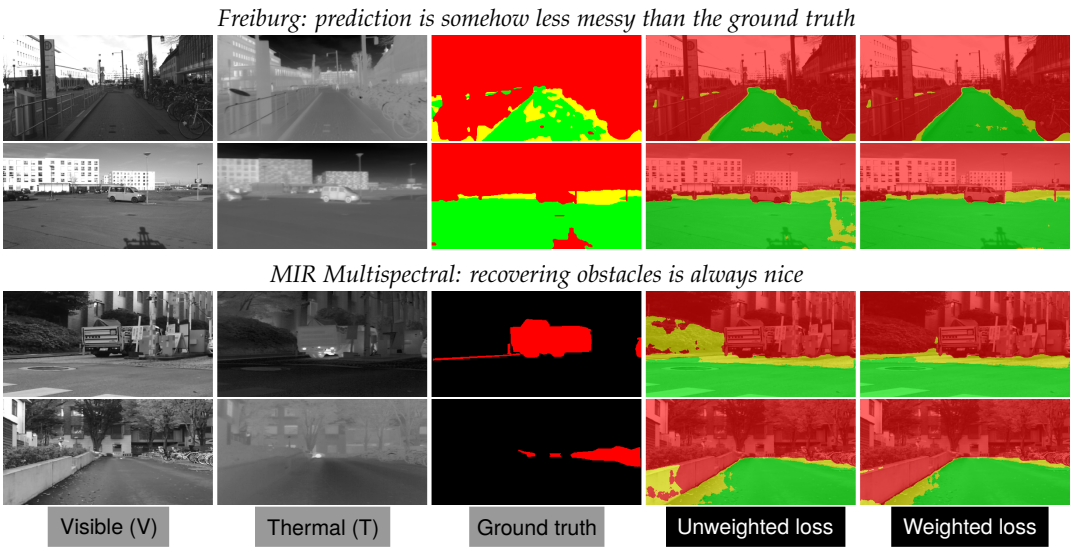


Figure 9.13: Predictions by the two fusion models, with one example per dataset in our evaluation.

These results suggest that the proposed fusion architecture can be successfully initialized with a pre-trained semantic model from a different dataset and achieve comparable performance. Considering the lack of public large-scale fusion datasets and the effort required to pre-train each branch of the fusion network, this could have interesting implications: the fusion network can be initialized with readily available pre-trained SegNet models trained on separate large-scale single-modality datasets, and then learn to fuse modalities for learning driveability on a (potentially smaller) multi-modal dataset.

Effect of loss weighting

Out of the two fusion models, loss weighting seems to be the most influential for $V_{\text{Cityscapes}} + T_{\text{FreiburgThermal}}$, where it consistently improves IoU for undriveable ■ areas across all datasets, therefore we show examples in Figure 9.13 for this model. Although it continues to consolidate the segmentation, visually, we do not see the same level of improvement from applying loss weighting than in the previous cross-dataset experiment, and it fails to improve weighted accuracy for half of the datasets in the evaluation. This could be due to the inaccuracy of Freiburg Thermal's pixel annotations: loss weight maps are generated from ground truth masks, with the assumption that scene elements are correctly outlined. If outlines are incorrectly labelled (eg. patches or shadows on the road in Figure 9.13), the loss weighting scheme may assign low importance to pixels of interest.



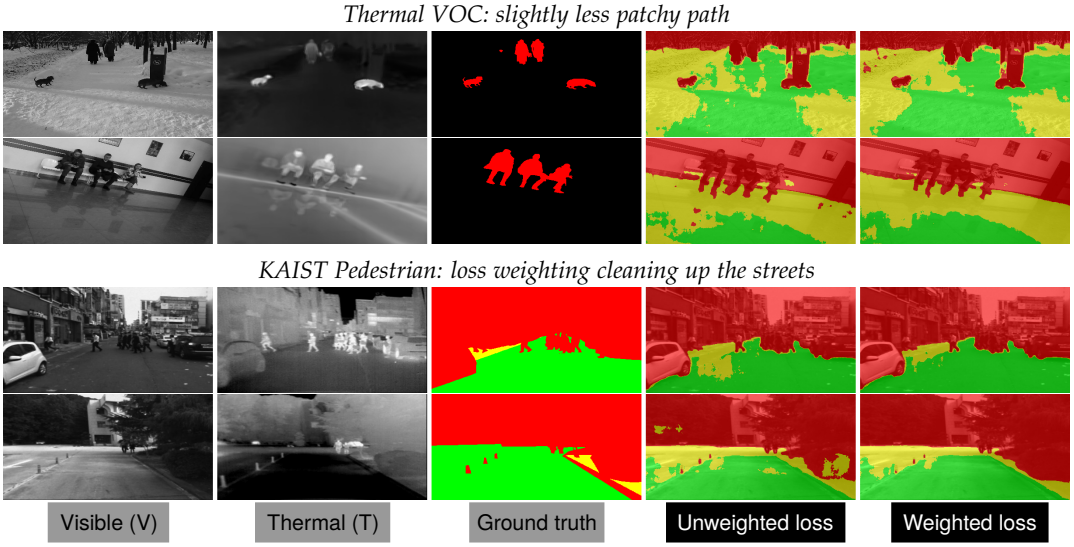


Figure 9.13: Comparison of predictions by the $V_{\text{Cityscapes}}$ + $T_{\text{FreiburgThermal}}$ fusion model, trained with and without loss weighting, with two samples from each of the datasets in Table 9.2.

that's enough pictures I think

Chapter 10

Discussion

In Section 10.1, we first reflect on our findings with respect to the research questions defined in Section 1.2. Section 10.2 raises some over-arching challenges and limitations of our methods, and Section 10.3 zooms in on some specific weaknesses and possible improvements. Section 10.4 considers the broader context of robotic navigation, outlining possible next steps for real-world operation. Lastly, in Section 10.5 we spread out further into research dreams about potential extensions of this work. Apologies in advance for the length - I blame all the papers mentioned here for being too interesting.

10.1 Research questions

To what extent does parsing outdoor scenes in terms of driveability rather than specific semantic classes help when faced with unconstrained environments?

In Section 6.2 we train segmentation models to directly predict 3-level driveability from images (either from scratch or via transfer learning), and compare them to a baseline which predicts object classes followed by a naive mapping to driveability in the post-prediction stage. Looking at rate of convergence, training stability and segmentation performance, our results suggest that training SegNet on specific object classes is an easier learning task, but does not perform on par with models trained to directly predict driveability levels. For generalization to new obstacles from out-of-dataset samples, learning driveability via transfer learning yields the best results: the model first learns fine object-specific features, and then is adapted to learn broader, more generic representations.

Defining the learning problem based on a notion of level (*how driveable is this?*) rather than descriptive categories (*what is this?*) has also opened interesting connections to affordance-based perception in robotic literature, and ordinal classification approaches which incorporate inter-class relations during learning. In a

standard object-based segmentation task, there is no effective difference between mis-classifying the road as a sidewalk or as an elephant - both are simply considered wrong, despite one mistake being much more reasonable than the other. In contrast, our driveability definition clearly expresses a ranking between the classes, and we show that by manipulating the soft label encoding used for training, we can adjust how different mistakes are penalized based on prior knowledge about the task - this proves to be an effective way to reduce mistake severity in different types of scenes.

The cross-dataset experiment in Section 9.2 also shows that our 3-level driveability definition allows segmentation models to learn from a combination of multiple datasets, despite them being labelled with incompatible sets of semantic classes. Applying the same training procedure with an object-based class definition would result in non-overlapping classes across datasets and a potential increase in the number of output channels of the network for every new dataset added: for instance, rather than learning a shared notion of obstacle across different scenes, the model would learn to recognize cats from one dataset, and trees and bicycles from another. Adding training data would then require annotating images for all the object classes present across the combined datasets. In contrast, learning a generic, universal notion of driveability allows each driveability level to encompass training examples from different sources, and from a practical stand-point, makes the labelling process much less laborious than outlining individual objects.

Despite these advantages, as discussed in [73], parsing the scene in terms of affordance or functionality adds a layer of subjectivity to the labelling and the evaluation. For instance, in a descriptive computer vision task, grass can (and should) always be labelled as grass with little to no ambiguity, but in terms of navigational affordance, its driveability level may depend on the vehicle's ability to cope with varying grass lengths. Another semantic category such as water could range from being a shallow puddle which can safely be traversed, to a pond or lake which would be hazardous to drive into. The edge of the sidewalk can range from being a non-issue for a legged robot, a slight bump on the way for a large vehicle, to an insurmountable obstacle for a small mobile robot. While descriptive segmentation can usually be solved and assessed in a vacuum regardless of context or application, the same cannot be said when segmenting the scene with such a nebulous concept as driveability.

How can multiple imaging modalities be combined for this purpose?

A simple approach consists of incorporating modalities as additional image channels at the input of the network. However, this approach is as uncool as it sounds. As described in Section 8.2, a cooler approach which is supported by state-of-the-art literature [25] consists of treating each modality as a separate input stream, and giving them each their own specialized encoder. Modality-specific features can

then either be fed to a common decoder (*mid* fusion) or up-sampled independently by separate decoders and fused right before the prediction stage (*late* fusion). We also propose a third fusion variant, which performs fusion in both locations (*dual* fusion). The question then lies in how to fuse modality-specific features into a shared representation - for this, we adapt the fusion unit in from [109] and propose a modification which results in improved training speed and stability, and more consistent performance across modality combinations and fusion variants.

What are the drawbacks and benefits of a multi-modal architecture for this task, compared to single-modality approaches?

A first hurdle when developing multi-modal fusion methods is data: recording the environment from a combination of multiple sensors brings its own set of challenges compared to simply taking images with a single camera [25]. Although we bypass data collection altogether in favor of existing datasets, we note that the extent of available data quickly narrows when adding additional modalities to the grocery list, especially for less conventional imagery such as infrared. In contrast, RGB-only semantic segmentation datasets are much easier to come by [76].

The addition of a second or third modality for segmentation also requires developing a more complex model. Fusing modalities at the input of the network via channel stacking requires minimal architectural changes, and can be trained end-to-end at little to no additional computational cost, but does not bring consistent improvement compared to relying on visible spectrum images alone, and even degrades segmentation performance for some modality combinations. Employing modality-specific branches in a *mid*, *dual* or *late* configuration comes with a significant cost in terms of inference time and memory (cf. our benchmark in Section 8.3), but has a clear performance advantage over a visible spectrum-only model. We find that deep fusion consistently out-performs early fusion and single-modality baselines across all 3 datasets in our multi-modal experiments, aligning with the findings in [109] - even on the Freiburg Forest and Cityscapes datasets which were captured in favorable illumination conditions, incorporating infrared and/or depth into the prediction brings an improvement in IoU across all driveability levels, and reduces mistake severity. When comparing deep fusion variants, we achieve remarkable performance with a *mid* architecture for a bi-modal input - however, its performance hinges on the quality of visible spectrum images, whose intermediate-level features in the encoder are used to guide up-sampling in the common decoder. The *dual* architecture's performance proves to be more robust to challenging lighting, and scales better to an increasing number of modalities.

Nevertheless, the visible spectrum SegNet model achieves good performance of its own when predicting driveability (pixel accuracy in the order of 95% in urban and forested scenes), and one could argue that the addition of depth or infrared imaging is not worth the trouble for normal day-time operation. Indeed,

when trained on a single dataset, the model learns to specialize in a particular domain and rarely makes flagrant mistakes when evaluated on images from the same dataset. However, we point to our experiments in Chapter 9 which reveal significant limitations of visible spectrum imaging for jointly learning driveability across multiple diverse datasets, or when faced with challenging out-of-dataset samples.

10.2 Challenges and limitations

Let's just blame the data Due to the lack of available multi-modal data recorded in truly challenging conditions, we cannot speak for the models' performance under different weathers or at night-time, for instance. All the visible spectrum images used in our experiments are clear enough that the scene can be labelled by a human annotator looking at this modality alone. Thus, we believe to not have explored the full potential of deep adaptive fusion, in cases where the visible spectrum cannot be relied upon for interpreting the scene. We also note that our cross-dataset experiment was performed with a single-modality model; it may be more difficult to combine infrared or depth data from different datasets due to the variety in sensor characteristics, but would be worth investigating.

We have also encountered limitations in the way that we generate ground truth labels for driveability segmentation. A blind mapping from descriptive object classes to driveability ignores the subjective and contextual nature of affordance labelling which we have discussed in the previous section. In addition, when trying to train a classifier on a combination of different datasets, we encounter many of same the issues addressed in [39] (although we tackle a very different task): a lack of global agreement on label definitions (resulting in conflicting ground truth examples), heterogeneous annotation styles, data bias (a majority of urban scenes), and imbalanced class distributions.

Or let's blame the metrics We have largely evaluated our approach as a standard segmentation task based on pixel-wise correctness (with accuracy and IoU). However, this assessment operates under the assumption that finer segmentation which maximises quantitative segmentation metrics (with respect to densely annotated ground truth masks) is desirable for path planning. It does not capture qualitative or subjective factors which may in fact make a less accurate model more suited for navigation-oriented scene understanding. For instance, the improvements brought by pixel-wise loss weighting in our experiments are primarily qualitative in nature and do not necessarily translate to improvement in pixel-wise correctness, yet could be valuable if a smoother, less detailed segmentation is desired. [122] raises an interesting discussion about the limitations of widely-used semantic segmentation metrics for scene understanding in real-world applications, and proposes

region-based metrics which separately account for under- and over-segmentation while allowing ambiguity along segmentation boundaries. In addition, as recommended by [25], assessing multi-modal segmentation approaches for autonomous driving should extend beyond quantifying deviations from a ground truth mask, and include systematic measures of how the model copes with different types of scenes, outdoor conditions, or sensor failures, for instance.

Unlike existing work tackling pixel-wise classification for scene understanding, we draw from ordinal classification literature [9] and incorporate a mistake severity metric as a first step towards ranking mis-classifications rather than considering them all equally unacceptable. However, this remains an overly simplistic measure, since it only quantifies the magnitude of errors and is blind to their direction - which, as mentioned in Section 6.3.4, we believe to be relevant for a navigation-oriented task due to the severity of under-segmenting obstacles as opposed to over-segmenting them. Much like pixel accuracy, this metric is also skewed by class imbalance, and in our case can simply be minimized by always predicting the low-risk middle level  . Lastly, being a regression metric, it captures a distance between a target and predicted level, which in our case is defined rather arbitrarily; the rank-based metrics used in [93] may be more appropriate.

A more comprehensive evaluation of our approach would first require defining what we would like the segmentation to look like when used as a basis for planning, and reasoning more carefully about what kind of mistakes can be tolerated vs. should be highly penalized in the context of real-world navigation.

10.3 Going further down the rabbit hole

Tuning things is a bore In our experiments, we have paid little to no emphasis on tuning hyper-parameters for maximizing performance; we stuck with the same set of hyper-parameters across all experiments (except for the reduction in learning rate for transfer learning) to maintain fair grounds for comparison. However, this may not have done justice to certain models which would have achieved better performance had they been given the hyper-parameter tuning efforts they deserve (eg. [23] mentions that the learning rate should be carefully chosen based on the entropy of the chosen soft label encoding, and [109] trains different parts of their fusion network at different rates). We also found our results to be significantly influenced by weight initialization (eg. when comparing the effect of using a pre-trained layer in our fusion unit in Section 8.2.3, or when initializing a fusion branch with pre-trained weights from a different dataset in paragraph 9.3.1); initializing depth and infrared segmentation models from Chapter 7 with pre-trained weights learned from visible spectrum images may have given an unfair advantage to the visible spectrum model, and limited the potential of the deep fusion models.

Multi-modal data augmentation When training our fusion models on RGB+IR images, we apply the same random data augmentation to both modalities. For geometric transformations, this is necessary to preserve spatial alignment. However, considering the properties of infrared imaging, we question whether applying the same photometric transformation makes sense: in reality, a change in brightness captured with a visible spectrum camera would not translate to the same change in brightness in thermal images, since LWIR cameras are blind to lighting conditions - and conversely, an increase in the thermal intensity of an object would not translate to the same change in visible light intensity in an RGB image. Thus, we stress that cross-modal data augmentation techniques should be investigated further.

Loss weighting for weight loss Our loss weighting scheme was developed purely as an initial proof of concept, and we have not evaluated the effect of the parameters in the weight map formulation (eg. rate at which the pixel weight decays from boundaries, or the overall loss scaling factor) on learning or segmentation quality; the work in [88] which inspired our approach uses a large scaling factor of 10, but without any justification for this value, or comparison to a standard uniformly weighted loss scheme for reference. We also note that our current implementation blindly generates weight maps regardless of label type, rather than ignoring void areas (cf. the examples in Section A.6), which is not great, but a pretty easy fix.

Soft problems are hard to solve As we have mentioned in our analysis of Section 6.3.2, and is further argued in [93, 78], the approach we have taken for learning a ranking between classes (based on [23]) is poorly suited for a task where the predicted levels are not based on a measurable quantity, as it requires defining inter-class distances which are completely arbitrary yet directly influence classification performance. In our case, for generating soft labels, we evaluated 3 distance metrics on 2 different inter-class difference definitions, with a close look at the effect on mistake severity and segmentation quality, but the optimal choice of soft label encoding for learning driveability remains an open question. To avoid having to define these inter-class distances as a prior, [78] shows that the optimal soft label encoding can be learned rather than pre-defined, while [93] proposes to predict ranking probabilities rather than class probabilities, using a rank-based loss for training, and thus bypassing the notion of inter-class distance altogether.

Cooler fusion for indexed un-pooling When designing the fusion architecture, our main goal was to preserve SegNet's topology and indexed un-pooling technique, in order to leverage previously trained models. To do this, we have taken a simple (rather lazy) approach, which only considers the input and output of the decoder as possible points of fusion, such that SegNet's encoder-decoder connections (which encode max-pooling locations) are left undisturbed; unlike [109], no

intermediate representations are shared during up-sampling. Techniques for fusing sets of pooling indices from multiple encoders would be worth investigating, especially for our mid-fusion configuration which currently only uses the pooling indices from a single modality-specific encoder for up-sampling. Extending the SegNet model, [69] shows that pooling indices can be learned adaptively as a function of feature maps - a technique which could lend itself well in combination with the adaptive fusion unit used in our architecture.

More small questions Does color in visible spectrum images matter for estimating driveability? How does cross-modal spatial misalignment at the input affect performance for different fusion configurations, and how much of it can we get away with? Is it necessary to up-sample feature maps back to the full original image resolution if we don't care much about details? Call me if you find out.

10.4 Towards navigation: next steps

From colours to planning We have tackled robotic perception purely as a computer vision task, and have not explored how the resulting scene representation can be used for path planning and navigation. However, the viability of our approach and its applicability to autonomous navigation systems are supported by existing work. For instance, [45] plans safe collision-free routes for a mobile robot by generating artificial potential fields from obstacle segmentation masks. [86] shows that augmenting classical geometric maps with pixel-wise navigability affordance remarkably improves goal-directed planning in dynamic environments. In predicting a driveability level per pixel, our method directly outputs a quantity which a planning algorithm can try to maximise when sampling and selecting trajectories.

Whenever, Wherever? While this work specifically considers outdoor scenes since they bring interesting challenges (ambient conditions, complex scenes, diversity of objects), our methods could analogously be applied to images captured in indoor environments. Indeed, unlike descriptive computer vision approaches which train a network to recognize scene-specific elements withing a particular domain (eg. {tree, road, bike} outdoors, and {chair, floor, table} indoors), learning driveability affordance places no prior constraints on the type of environment in which the system can operate, enabling perception in mixed scenarios rather than treating indoor/outdoor scenes as completely separate domains. Extending our cross-dataset experiment from Section 9.2, it would be interesting to investigate how a consistent notion of driveability can be learned across a combination of indoor and outdoor scenes, paying closer attention to cross-dataset bias and adaptation methods as described in [39].

Need for speed Computational constraints for real-time operation on an embedded system were not considered within the scope of this work, and the benchmarking experiment in Section 8.3 shows that inference with our SegNet-based deep fusion architecture at 20+ FPS requires a high-end (exorbitantly priced) GPU, making it prohibitive for resource-constrained applications. However, the proposed training strategies and fusion methods are not bound to a specific segmentation architecture, and could be applied to other more lightweight networks with a lower computational footprint. In this direction, [98] systematically compares different encoder-decoder combinations with a focus on efficiency for autonomous driving. Network pruning can also be applied to deep fusion architectures for parameter reduction with a minimal performance trade-off [109].

10.5 Future directions: dream big or go home

Once upon a time Aligning with related work, we treat each input image as an independent sample. However, on autonomous navigation platforms, images are captured at a high frame-rate with significant overlap between successive captures. Thus, it may be beneficial to rather consider the input as a sequence of time-adjacent frames, in order to incorporate a temporal dimension in the prediction, using previous frames to guide the segmentation of the current frame and enforce temporal consistency. Recurrent networks are a common choice for modelling spatio-temporal dependencies between pixels [76], however multi-modal scene understanding approaches in this direction remain scarce [25]. Such an approach places additional requirements on the datasets used for training (we would have to rely on video sequence datasets like Synthia [89] or Freiburg Thermal [113]).

Pixels holding hands Our method predicts driveability at the pixel level based on local, appearance-based information, and by employing standard classification loss, essentially treats images as bags of unrelated pixels. However, this often results in speckled or fragmented segmentation, unnecessarily detailed contours, and areas being labelled as driveable even though they would not be reachable by a robot in practice, depending on its size and movement capabilities (eg. a road on the other side of a barrier, a narrow passage between obstacles). Simplifying and correcting the scene representation for planning could be performed as a post-prediction step, but could also be incorporated during the learning itself, by applying geometrical and topological constraints on the prediction, such that, for instance, the driveable area starts from the bottom-center of the image (where the vehicle is), and spreads out maintaining a single, smooth cohesive shape. Such constraints can be incorporated in the loss term itself, as proposed in [8, 77].

Less baby-sitting, more self-supervision Similarly to [71], we have taken a fully-supervised approach for affordance segmentation, relying on pixel-wise ground truth segmentation masks for training, which either limits us to existing image segmentation datasets, or requires us to label new data ourselves, which is a rather tedious process. This tempts us to explore weakly or self-supervised labelling approaches, where driveability labels are generated with minimal human intervention, by simply driving. For instance, [53] generates navigational affordance labels by detecting collisions or bumpy terrain with on-board odometry and range sensors. Another interesting yet under-exploited modality is audio: the acoustic signature of a vehicle's surroundings and of its own movement can reveal informative scene properties such as the type of surface that it is driving on, or the direction of approaching vehicles with a 360° field of view, making auditory perception an interesting candidate, not only as a form of weak or self-supervision for labelling driveability, but also as a complementary modality during online operation [74]. However, as argued in Section 3.1.4, labelling data from experience by letting a robot explore an outdoor environment is time-consuming and risky, if not impossible altogether (eg. on public roads). To learn from the physical world without having to put a robot in harm's way, an idea would be to label driveability by observing the behaviour of other agents in a scene, following the intuition that areas which are the most driveable are also the most likely to be driven/walked on by others. Learning affordance from observation or demonstration is an active field of research, however currently seems to be limited to human-object interaction rather than locomotion [2, 40].

Robots should be confused when it matters Much like humans navigate the world with varying degrees of confidence depending on how familiar and clear their surroundings are, deep learning models should ideally exhibit lower certainty in their predictions when encountering previously unseen or low-quality data, in order to better inform control decisions, detect failures and exert necessary caution in challenging conditions. Ideally, prediction confidence should be a reliable indicator of prediction accuracy, such that the network is less likely to make severe mistakes with a high level of certainty. However, modern neural networks trained with a standard softmax classifier only output relative probabilities between classes rather than any meaningful measure of uncertainty, and have shown to produce over-confident predictions even when faced with ambiguous or out-of-distribution samples, making them poorly suited for safety-critical applications [1, 36, 81, 20]. In our experiments, we have simply taken the predicted driveability level for each pixel as the class with the highest probability, without any consideration for data or model uncertainty [68]. Motivated by the aforementioned works, extending our approach with a probabilistic interpretation of driveability estimation would be a interesting direction to pursue, with direct relevance to real-world operation.

Chapter 11

Conclusion

This project was primarily undertaken as a curiosity-driven exploration, drawing from recent developments in computer vision and scene understanding to try to interpret ego-centric images in a useful and reliable way. We specifically work with images taken out in the wild - compared to indoor scenes, the environment tends to be more diverse, unpredictable and ambiguous, motivating the use of depth and infrared as complementary modalities to visible spectrum imaging. Although the robots and vehicles considered in this work remain nothing more than figments of our imagination, framing the problem of scene understanding within the context of autonomous navigation allows us to deviate from conventional task-agnostic vision, and develop methods which are tailored to an outdoor driving task. Our starting point is a conventional one: fully-supervised image segmentation with a well-established encoder-decoder CNN, using public datasets for training and evaluation. Our contribution lies in the way that we formulate the learning problem, and integrate the network into a multi-modal architecture for adaptive fusion.

For *useful* perception, following the concept of affordances, we propose to segment images at the functional level: for every pixel in the image, the network directly predicts an estimate of how suitable it is to drive on. We generate ground truth labels by blindly mapping object labels from pixel-annotated datasets to 3 driveability levels, and soften the labels into a class probability distribution which encodes a ranking between the 3 levels. We show this to be an effective way to reduce error and mistake severity compared to a standard one-hot labelling approach, while requiring no architectural changes. We also explore the idea of adjusting the contribution of each pixel based on its position and distance from segmentation boundaries: correctly segmenting areas closest to the ego-vehicle is more important than precisely delineating objects across the whole image. We implement this as an importance map which weighs the loss at the pixel-level during learning, and find it most beneficial as a final training step to consolidate the segmentation, as done in our cross-dataset experiment.

For *reliable* perception, we evaluate deep fusion approaches in a bi-modal and tri-modal configuration across urban and forested scenes. Compared to single-modality prediction, our early fusion baseline shows mixed results, depending on the modalities stacked at the input: combining visible spectrum with near-infrared images improves segmentation for instance, while the addition of less correlated data such as noisy stereo depth degrades it. By giving each modality a dedicated encoder for feature extraction, and shifting the point of fusion deeper into the network, we out-perform early fusion and single-modality baselines for every modality combination in our experiments, especially in the segmentation of obstacles. In our final experiments, we highlight the limitations of solely relying on a monocular RGB camera for learning to segment driveability across a wide range of challenging datasets, and show that the incorporation of thermal imaging brings substantial benefits for out-of-dataset generalization to new scenes and obstacles.

In sum, considering the lack of prior work jointly tackling affordance-based and multi-modal approaches for outdoor perception, we would like to think of this pile of pages as a pretty cool (albeit mildly useful and definitely too long) first step towards something potentially much cooler.

Bibliography

[1] Vijay Badrinarayanan Alex Kendall and Roberto Cipolla. "Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding". In: *Proceedings of the British Machine Vision Conference (BMVC)*. Ed. by Gabriel Brostow Tae-Kyun Kim Stefanos Zafeiriou and Krystian Mikolajczyk. BMVA Press, Sept. 2017, pp. 57.1–57.12. ISBN: 1-901725-60-X. DOI: 10.5244/C.31.57.

[2] Paola Ardón et al. *Affordances in Robotic Tasks - A Survey*. 2020. arXiv: 2004.07400 [cs.RO].

[3] Amir Atapour-Abarghouei and Toby P. Breckon. "A comparative review of plausible hole filling strategies in the context of scene depth image completion". In: *Computers & Graphics* 72 (2018), pp. 39–58. ISSN: 0097-8493. DOI: doi.org/10.1016/j.cag.2018.02.001.

[4] V. Badrinarayanan, A. Kendall, and R. 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. DOI: 10.1109/TPAMI.2016.2644615.

[5] D. Barnes, W. Maddern, and I. Posner. "Find your own way: Weakly-supervised segmentation of path proposals for urban autonomy". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2017, pp. 203–210. DOI: 10.1109/ICRA.2017.7989025.

[6] A. Behl et al. "Label Efficient Visual Abstractions for Autonomous Driving". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020, pp. 2338–2345. DOI: 10.1109/IROS45743.2020.9340641.

[7] M. Bennamoun et al. "Guest Editors' Introduction to the Special Issue on RGB-D Vision: Methods and Applications". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42.10 (2020), pp. 2329–2332. DOI: 10.1109/TPAMI.2020.2976227.

[8] Aïcha BenTaieb and Ghassan Hamarneh. "Topology Aware Fully Convolutional Networks for Histology Gland Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*. Ed. by Sebastien Ourselin et al. Cham: Springer International Publishing, 2016, pp. 460–468. ISBN: 978-3-319-46723-8. DOI: 10.1007/978-3-319-46723-8_53.

[9] Luca Bertinetto et al. "Making Better Mistakes: Leveraging Class Hierarchies With Deep Networks". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 12503–12512. DOI: 10.1109/CVPR42600.2020.01252.

[10] Lukas Biewald. *Experiment Tracking with Weights and Biases*. 2020. URL: https://www.wandb.com/.

[11] G. Borgefors. "Distance transformations in digital images". In: *Comput. Vis. Graph. Image Process.* 34 (1986), pp. 344–371.

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

[13] Matthew Brown and Sabine Süsstrunk. "Multi-spectral SIFT for scene category recognition". In: *CVPR 2011*. 2011, pp. 177–184. DOI: 10.1109/CVPR.2011.5995637.

[14] Alexander Buslaev et al. "Albumentations: Fast and Flexible Image Augmentations". In: *Information* 11.2 (2020). ISSN: 2078-2489. DOI: 10.3390/info11020125.

[15] C. Chen et al. "DeepDriving: Learning Affordance for Direct Perception in Autonomous Driving". In: *Proceedings of the IEEE International Conference on Computer Vision* (2015), pp. 2722–2730. DOI: 10.1109/ICCV.2015.312.

[16] L. Chen et al. "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.4 (2018), pp. 834–848. DOI: 10.1109/TPAMI.2017.2699184.

[17] Jaehoon Cho et al. *A Large RGB-D Dataset for Semi-supervised Monocular Depth Estimation*. 2019. arXiv: 1904.10230 [cs.CV].

[18] Gyeongmin Choe et al. "RANUS: RGB and NIR Urban Scene Dataset for Deep Scene Parsing". In: *IEEE Robotics and Automation Letters* 3.3 (2018), pp. 1808–1815. DOI: 10.1109/LRA.2018.2801390.

[19] F. Codevilla et al. "Exploring the Limitations of Behavior Cloning for Autonomous Driving". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 9328–9337. DOI: 10.1109/ICCV.2019.00942.

[20] Charles Corbière et al. "Addressing Failure Prediction by Learning Model Confidence". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019.

[21] Marius Cordts et al. "The Cityscapes Dataset for Semantic Urban Scene Understanding". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 3213–3223. DOI: 10.1109/CVPR.2016.350.

[22] Vincent Dumoulin and Francesco Visin. "A guide to convolution arithmetic for deep learning". In: *ArXiv e-prints* (Mar. 2016). eprint: 1603.07285.

[23] R. Díaz and A. Marathe. "Soft Labels for Ordinal Regression". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 4733–4742. DOI: 10.1109/CVPR.2019.00487.

[24] WA Falcon and .al. *PyTorch Lightning*. 2019. URL: https://github.com/PyTorchLightning/pytorch-lightning.

[25] Di Feng et al. "Deep Multi-Modal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges". In: *IEEE Transactions on Intelligent Transportation Systems* 22.3 (2021), pp. 1341–1360. DOI: 10.1109/TITS.2020.2972974.

[26] Rikke Gade and Thomas B. Moeslund. "Thermal cameras and applications: a survey". In: *Machine Vision and Applications* 25 (2014), pp. 245–262. ISSN: 0932-8092. DOI: 10.1007/s00138-013-0570-5.

[27] Aram Galstyan and Paul R. Cohen. "Empirical Comparison of "Hard" and "Soft" Label Propagation for Relational Classification". In: *Inductive Logic Programming*. Ed. by Hendrik Blockeel et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 98–111. ISBN: 978-3-540-78469-2. DOI: 10.1007/978-3-540-78469-2_13.

[28] B. Gao et al. "Deep Label Distribution Learning With Label Ambiguity". In: *IEEE Transactions on Image Processing* 26.6 (2017), pp. 2825–2838. DOI: 10.1109/TIP.2017.2689998.

[29] Alberto Garcia-Garcia et al. "A Review on Deep Learning Techniques Applied to Semantic Segmentation". In: *CoRR* abs/1704.06857 (2017). arXiv: 1704.06857.

[30] Andreas Geiger et al. "Vision meets Robotics: The KITTI Dataset". In: *International Journal of Robotics Research (IJRR)* (2013). DOI: 10.1177/0278364913491297.

[31] Swarnendu Ghosh et al. "Understanding Deep Learning Techniques for Image Segmentation". In: *ACM Comput. Surv.* 52.4 (Aug. 2019). ISSN: 0360-0300. DOI: 10.1145/3329784.

[32] James J. Gibson. *The Ecological Approach to Visual Perception*. Psychology Press, 1979. DOI: 10.4324/9781315740218.

[33] Xavier Glorot and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Yee Whye Teh and Mike Titterington. Vol. 9. Proceedings of Machine Learning Research. PMLR, May 2010, pp. 249–256.

[34] C. Gros, Andreanne Lemay, and J. Cohen-Adad. "SoftSeg: Advantages of soft versus binary training for image segmentation". In: *Medical image analysis* 71 (2021), p. 102038. DOI: 10.1016/j.media.2021.102038.

[35] F. A. Guerrero-Peña et al. "Multiclass Weighted Loss for Instance Segmentation of Cluttered Cells". In: *2018 25th IEEE International Conference on Image Processing (ICIP)*. 2018, pp. 2451–2455. DOI: 10.1109/ICIP.2018.8451187.

[36] Chuan Guo et al. "On Calibration of Modern Neural Networks". In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, Aug. 2017, pp. 1321–1330. DOI: 10.5555/3305381.3305518.

[37] Yulan Guo et al. "Deep Learning for 3D Point Clouds: A Survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020), pp. 1–1. DOI: 10.1109/TPAMI.2020.3005434.

[38] Q. Ha et al. "MFNet: Towards real-time semantic segmentation for autonomous vehicles with multi-spectral scenes". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017, pp. 5108–5115. DOI: 10.1109/IROS.2017.8206396.

[39] Byungok Han et al. "Toward Unbiased Facial Expression Recognition in the Wild via Cross-Dataset Adaptation". In: *IEEE Access* 8 (2020), pp. 159172–159181. DOI: 10.1109/ACCESS.2020.3018738.

[40] Mohammed Hassanin, Salman Khan, and Murat Tahtali. "Visual Affordance and Function Understanding: A Survey". In: *ACM Comput. Surv.* 54.3 (Apr. 2021). ISSN: 0360-0300. DOI: 10.1145/3446370.

[41] Caner Hazirbas et al. "FuseNet: Incorporating Depth into Semantic Segmentation via Fusion-Based CNN Architecture". In: *Computer Vision – ACCV 2016*. Ed. by Shang-Hong Lai et al. Cham: Springer International Publishing, 2017, pp. 213–228. ISBN: 978-3-319-54181-5. DOI: 10.1007/978-3-319-54181-5_14.

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

[43] Kaiming He et al. “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification”. In: *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*. ICCV ’15. USA: IEEE Computer Society, 2015, 1026–1034. ISBN: 9781467383912. DOI: 10.1109/ICCV.2015.123.

[44] 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.

[45] M. Hua, Y. Nan, and S. Lian. “Small Obstacle Avoidance Based on RGB-D Semantic Segmentation”. In: *IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. 2019, pp. 886–894. DOI: 10.1109/ICCVW.2019.00117.

[46] G. Humblot-Renaux et al. “Thermal Imaging on Smart Vehicles for Person and Road Detection: Can a Lazy Approach Work?” In: *IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. 2020, pp. 1–6. DOI: 10.1109/ITSC45102.2020.9294671.

[47] Soonmin Hwang et al. “Multispectral pedestrian detection: Benchmark dataset and baseline”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 1037–1045. DOI: 10.1109/CVPR.2015.7298706.

[48] Sergey Ioffe and Christian Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*. ICML’15. Lille, France: JMLR.org, 2015, 448–456.

[49] L. Jamone et al. “Affordances in Psychology, Neuroscience, and Robotics: A Survey”. In: *IEEE Transactions on Cognitive and Developmental Systems* 10.1 (2018), pp. 4–25. DOI: 10.1109/TCDS.2016.2594134.

[50] Joel Janai et al. “Computer Vision for Autonomous Vehicles: Problems, Datasets and State of the Art”. In: *Foundations and Trends® in Computer Graphics and Vision* 12.1–3 (2020), pp. 1–308. ISSN: 1572-2740. DOI: 10.1561/0600000079.

[51] M. Jaritz et al. “Sparse and Dense Data with CNNs: Depth Completion and Semantic Segmentation”. In: *2018 International Conference on 3D Vision (3DV)*. 2018, pp. 52–60. DOI: 10.1109/3DV.2018.00017.

[52] C. Jing et al. “A comparison and analysis of RGB-D cameras’ depth performance for robotics application”. In: *24th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*. 2017, pp. 1–6. DOI: 10.1109/M2VIP.2017.8211432.

[53] G. Kahn, P. Abbeel, and S. Levine. "BADGR: An Autonomous Self-Supervised Learning-Based Navigation System". In: *IEEE Robotics and Automation Letters* (2021), pp. 1–1. DOI: 10.1109/LRA.2021.3057023.

[54] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015.

[55] Vladimir V. Kniaz and Vladimir A. Knyaz. "Chapter 6 - Multispectral Person Re-Identification Using GAN for Color-to-Thermal Image Translation". In: *Multimodal Scene Understanding*. Ed. by Michael Ying Yang, Bodo Rosenhahn, and Vittorio Murino. Academic Press, 2019, pp. 135 –158. ISBN: 978-0-12-817358-9. DOI: 10.1016/B978-0-12-817358-9.00012-3.

[56] Zoltan Koppanyi et al. "Chapter 3 - Multimodal Semantic Segmentation: Fusion of RGB and Depth Data in Convolutional Neural Networks". In: *Multimodal Scene Understanding*. Ed. by Michael Ying Yang, Bodo Rosenhahn, and Vittorio Murino. Academic Press, 2019, pp. 41 –64. ISBN: 978-0-12-817358-9. DOI: 10.1016/B978-0-12-817358-9.00009-3.

[57] Mikkel Fly Kragh et al. "FieldSAFE: Dataset for Obstacle Detection in Agriculture". In: *arXiv preprint arXiv:1709.03526* (2017).

[58] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc., 2012.

[59] Jason Ku, Ali Harakeh, and Steven L. Waslander. "In Defense of Classical Image Processing: Fast Depth Completion on the CPU". In: *2018 15th Conference on Computer and Robot Vision (CRV)*. 2018, pp. 16–22. DOI: 10.1109/CRV.2018.00013.

[60] Jan Kukacka, Vladimir Golkov, and Daniel Cremers. "Regularization for Deep Learning: A Taxonomy". In: *CoRR* abs/1710.10686 (2017). arXiv: 1710.10686. URL: http://arxiv.org/abs/1710.10686.

[61] *Labelbox*. 2021. URL: https://labelbox.com.

[62] A. J. Lee et al. "ViViD : Vision for Visibility Dataset". In: *IEEE International Conference on Robotics and Automation (ICRA). Dataset Generation and Benchmarking of SLAM Algorithms for Robotics and VR/AR workshop*. 2019.

[63] Dan Levi et al. "StixelNet: A Deep Convolutional Network for Obstacle Detection and Road Segmentation." In: *Proceedings of the British Machine Vision Conference 2015, BMVC*. Vol. 1. 2. 2015, p. 4. DOI: 10.5244/C.29.109.

[64] Xiaoxiao Li et al. “Not All Pixels Are Equal: Difficulty-Aware Semantic Segmentation via Deep Layer Cascade”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 6459–6468. DOI: 10.1109/CVPR.2017.684.

[65] A. Ligocki, A. Jelinek, and L. Zalud. “Brno Urban Dataset - The New Data for Self-Driving Agents and Mapping Tasks”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2020, pp. 3284–3290. DOI: 10.1109/ICRA40945.2020.9197277.

[66] Xiaofeng Liu et al. “Unimodal-Uniform Constrained Wasserstein Training for Medical Diagnosis”. In: *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. 2019, pp. 332–341. DOI: 10.1109/ICCVW.2019.00044.

[67] J. Long, E. Shelhamer, and T. Darrell. “Fully convolutional networks for semantic segmentation”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 3431–3440. DOI: 10.1109/CVPR.2015.7298965.

[68] Antonio Loquercio, Mattia Segu, and Davide Scaramuzza. “A General Framework for Uncertainty Estimation in Deep Learning”. In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 3153–3160. DOI: 10.1109/LRA.2020.2974682.

[69] Hao Lu et al. “Indices Matter: Learning to Index for Deep Image Matting”. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 3265–3274. DOI: 10.1109/ICCV.2019.00336.

[70] Wenjie Luo et al. “Understanding the Effective Receptive Field in Deep Convolutional Neural Networks”. In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. NIPS’16. Curran Associates Inc., 2016, 4905–4913. ISBN: 9781510838819.

[71] Timo Lüddecke, Tomas Kulvicius, and Florentin Wörgötter. “Context-based affordance segmentation from 2D images for robot actions”. In: *Robotics and Autonomous Systems* 119 (2019), pp. 92–107. ISSN: 0921-8890. DOI: 10.1016/j.robot.2019.05.005.

[72] Timo Lüddecke and Florentin Wörgötter. “Fine-grained action plausibility rating”. In: *Robotics and Autonomous Systems* 129 (2020), p. 103511. ISSN: 0921-8890. DOI: 10.1016/j.robot.2020.103511.

[73] Timo Julian Lüddecke. “Action-oriented Scene Understanding”. PhD thesis. Georg-August-Universität Göttingen, 2019.

[74] Letizia Marchegiani and Xenofon Fafoutis. "How Well Can Driverless Vehicles Hear? A Gentle Introduction to Auditory Perception for Autonomous and Smart Vehicles". In: *IEEE Intelligent Transportation Systems Magazine* (2021), pp. 0–0. DOI: 10.1109/MITS.2021.3049425.

[75] Paul McManamon. *Introduction to LiDAR*. SPIE Press, July 2019. Chap. 1, pp. 1–20. ISBN: 9781510625396. DOI: 10.1117/3.2518254.

[76] Shervin Minaee et al. "Image Segmentation Using Deep Learning: A Survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021), pp. 1–1. DOI: 10.1109/TPAMI.2021.3059968.

[77] Agata Mosinska et al. "Beyond the Pixel-Wise Loss for Topology-Aware Delineation". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 3136–3145. DOI: 10.1109/CVPR.2018.00331.

[78] Raouf Muhamedrahimov, Amir Bar, and Ayelet Akselrod-Ballin. "Learning Interclass Relations for Intravenous Contrast Phase Classification in CT". In: *Medical Imaging with Deep Learning*. 2021.

[79] J. Muñoz-Bulnes et al. "Deep fully convolutional networks with random data augmentation for enhanced generalization in road detection". In: *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. 2017, pp. 366–371. DOI: 10.1109/ITSC.2017.8317901.

[80] Gerhard Neuhold et al. "The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes". In: *International Conference on Computer Vision (ICCV)*. 2017. DOI: 10.1109/ICCV.2017.534. URL: https://www.mapillary.com/dataset/vistas.

[81] Lukás Neumann, Andrew Zisserman, and A. Vedaldi. "Relaxed Softmax: Efficient Confidence Auto-Calibration for Safe Pedestrian Detection". In: *Conference on Neural Information Processing Systems (NIPS) - Workshop on Machine Learning for Intelligent Transportation Systems (MLITS)*. 2018.

[82] G. Papandreou, I. Kokkinos, and P. Savalle. "Modeling local and global deformations in Deep Learning: Epitomic convolution, Multiple Instance Learning, and sliding window detection". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 390–399. DOI: 10.1109/CVPR.2015.7298636.

[83] 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.

[84] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[85] P. Pinggera et al. “Lost and Found: detecting small road hazards for self-driving vehicles”. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2016, pp. 1099–1106. DOI: 10.1109/IROS.2016.7759186.

[86] William Qi et al. “Learning to Move with Affordance Maps”. In: *International Conference on Learning Representations (ICLR)*. 2020.

[87] Morgan Quigley et al. “ROS: an open-source Robot Operating System”. In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA) Workshop on Open Source Robotics*. Kobe, Japan, May 2009.

[88] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by Nassir Navab et al. Cham: Springer International Publishing, 2015, pp. 234–241. DOI: 10.1007/978-3-319-24574-4_28.

[89] German Ros et al. “The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 3234–3243. DOI: 10.1109/CVPR.2016.352.

[90] Anirban Roy and Sinisa Todorovic. “A Multi-scale CNN for Affordance Segmentation in RGB Images”. In: *Computer Vision – ECCV 2016*. Ed. by Bastian Leibe et al. Cham: Springer International Publishing, 2016, pp. 186–201. ISBN: 978-3-319-46493-0. DOI: 10.1007/978-3-319-46493-0_12.

[91] Sebastian Ruder. “An overview of gradient descent optimization algorithms”. In: *CoRR* abs/1609.04747 (2016). arXiv: 1609.04747. URL: http://arxiv.org/abs/1609.04747.

[92] Lukas Schneider et al. “Multimodal Neural Networks: RGB-D for Semantic Segmentation and Object Detection”. In: *Image Analysis*. Ed. by Puneet Sharma and Filippo Maria Bianchi. Cham: Springer International Publishing, 2017, pp. 98–109. ISBN: 978-3-319-59126-1. DOI: 10.1007/978-3-319-59126-1_9.

[93] Joan Serrat and Idoia Ruiz. “Rank-based ordinal classification”. In: *2020 25th International Conference on Pattern Recognition (ICPR)*. 2021, pp. 8069–8076. DOI: 10.1109/ICPR48806.2021.9412846.

[94] Marcel Sheeny et al. “RADIATE: A Radar Dataset for Automotive Perception”. In: *arXiv preprint arXiv:2010.09076* (2020).

[95] E. Shelhamer, J. Long, and T. Darrell. “Fully Convolutional Networks for Semantic Segmentation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.4 (2017), pp. 640–651. DOI: 10.1109/TPAMI.2016.2572683.

[96] S. S. Shivakumar et al. "PST900: RGB-Thermal Calibration, Dataset and Segmentation Network". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 2020, pp. 9441–9447. DOI: 10.1109/ICRA40945.2020.9196831.

[97] Connor Shorten and T. Khoshgoftaar. "A survey on Image Data Augmentation for Deep Learning". In: *Journal of Big Data* 6 (2019), pp. 1–48. DOI: doi.org/10.1186/s40537-019-0197-0.

[98] Mennatullah Siam et al. "A Comparative Study of Real-Time Semantic Segmentation for Autonomous Driving". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2018, pp. 700–70010. DOI: 10.1109/CVPRW.2018.00101.

[99] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *International Conference on Learning Representations*. 2015.

[100] Nitish Srivastava et al. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *J. Mach. Learn. Res.* 15.1 (Jan. 2014), 1929–1958. ISSN: 1532-4435.

[101] Lei Sun et al. "Real-Time Fusion Network for RGB-D Semantic Segmentation Incorporating Unexpected Obstacle Detection for Road-Driving Images". In: *IEEE Robotics and Automation Letters* 5.4 (2020), pp. 5558–5565. DOI: 10.1109/LRA.2020.3007457.

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

[103] C. Szegedy et al. "Rethinking the Inception Architecture for Computer Vision". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 2818–2826. DOI: 10.1109/CVPR.2016.308.

[104] Karasawa Takumi et al. "Multispectral Object Detection for Autonomous Vehicles". In: *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*. Thematic Workshops '17. Association for Computing Machinery, 2017, 35–43. ISBN: 9781450354165. DOI: 10.1145/3126686.3126727.

[105] M. Teichmann et al. "MultiNet: Real-time Joint Semantic Reasoning for Autonomous Driving". In: *IEEE Intelligent Vehicles Symposium (IV)*. 2018, pp. 1013–1020. DOI: 10.1109/IVS.2018.8500504.

[106] Wayne Treible et al. "CATS: A Color and Thermal Stereo Benchmark". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 134–142. DOI: 10.1109/CVPR.2017.22.

[107] J. Uhrig et al. “Sparsity Invariant CNNs”. In: *2017 International Conference on 3D Vision (3DV)*. 2017, pp. 11–20. DOI: 10.1109/3DV.2017.00012.

[108] A. Valada et al. “AdapNet: Adaptive semantic segmentation in adverse environmental conditions”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2017, pp. 4644–4651. DOI: 10.1109/ICRA.2017.7989540.

[109] Abhinav Valada, Rohit Mohan, and Wolfram Burgard. “Self-Supervised Model Adaptation for Multimodal Semantic Segmentation”. In: *International Journal of Computer Vision (IJCV)* (July 2019). Special Issue: Deep Learning for Robotic Vision. ISSN: 1573-1405. DOI: 10.1007/s11263-019-01188-y.

[110] Abhinav Valada et al. “Deep Multispectral Semantic Scene Understanding of Forested Environments Using Multimodal Fusion”. In: *2016 International Symposium on Experimental Robotics*. Ed. by Dana Kulić et al. Springer International Publishing, 2017, pp. 465–477. ISBN: 978-3-319-50115-4. DOI: 10.1007/978-3-319-50115-4_41.

[111] G. Varma et al. “IDD: A Dataset for Exploring Problems of Autonomous Navigation in Unconstrained Environments”. In: *IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2019, pp. 1743–1751. DOI: 10.1109/WACV.2019.00190.

[112] Igor Vasiljevic et al. *DIODE: A Dense Indoor and Outdoor DEpth Dataset*. 2019. arXiv: 1908.00463 [cs.CV].

[113] Johan Vertens, Jannik Zürn, and Wolfram Burgard. “HeatNet: Bridging the Day-Night Domain Gap in Semantic Segmentation with Thermal Images”. In: *arXiv preprint arXiv:2003.04645* (2020).

[114] Michael Vollmer and Klaus-Peter Möllmann. *Infrared thermal imaging: Fundamentals, research and applications*. 2nd ed. Oct. 2018. Chap. 6, pp. 447–474. ISBN: 978-3-527-41351-5. DOI: 10.1002/9783527693306.

[115] Maggie Wigness et al. “A RUGD Dataset for Autonomous Navigation and Visual Perception in Unstructured Outdoor Environments”. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2019, pp. 5000–5007. DOI: 10.1109/IROS40897.2019.8968283.

[116] Jae Shin Yoon et al. “Thermal-infrared based drivable region detection”. In: *2016 IEEE Intelligent Vehicles Symposium (IV)*. 2016, pp. 978–985. DOI: 10.1109/IVS.2016.7535507.

[117] Fisher Yu and Vladlen Koltun. “Multi-Scale Context Aggregation by Dilated Convolutions”. In: *4th International Conference on Learning Representations, ICLR*. Ed. by Yoshua Bengio and Yann LeCun. 2016.

[118] Ekim Yurtsever et al. "A Survey of Autonomous Driving: Common Practices and Emerging Technologies". In: *IEEE Access* 8 (2020), pp. 58443–58469. DOI: 10.1109/ACCESS.2020.2983149.

[119] Oliver Zendel et al. "WildDash - Creating Hazard-Aware Benchmarks". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Sept. 2018. DOI: 10.1007/978-3-030-01231-1_25.

[120] Chang-Bin Zhang et al. "Delving Deep into Label Smoothing". In: *CoRR* abs/2011.12562 (2020). arXiv: 2011.12562.

[121] Yifei Zhang et al. "Deep multimodal fusion for semantic image segmentation: A survey". In: *Image and Vision Computing* 105 (2021), p. 104042. ISSN: 0262-8856. DOI: 10.1016/j.imavis.2020.104042.

[122] Yuxiang Zhang, Sachin Mehta, and Anat Caspi. *Rethinking Semantic Segmentation Evaluation for Explainability and Model Selection*. 2021. arXiv: 2101.08418 [cs.CV].

[123] H. 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.

[124] Brady Zhou, Philipp Krähenbühl, and V. Koltun. "Does computer vision matter for action?" In: *Science Robotics* 4 (2019). DOI: 10.1126/scirobotics.aaw6661.

[125] Yi Zhu et al. "Improving Semantic Segmentation via Video Propagation and Label Relaxation". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 8848–8857. DOI: 10.1109/CVPR.2019.00906.

Appendix A

Miscellaneous mess

A.1 Mapping from object classes to driveability

driveability level	**original object class**			
void/unlabeled	ignore ego-vehicle	out of ROI rectification border	unlabeled	void
impossible	background bicycle boat bridge building bus car caravan	cat cone curve dog dynamic fence guardrail human	license plate minibus motorcycle obstacle pole sky static traffic light	traffic sign trailer train tree truck tunnel vegetation wall
possible	bump grass	parking railtrack	sidewalk	terrain
preferable	curb ground	lane marking	path	road

A.2 Dataset overview

	Platform	Sensors	Scene	Annotation	Conditions	Diversity	Size
ROB7 [46]	golf cart	**stereo RGB, LWIR**	university campus	N/A	day-time w/ glare & shadows	2 days	800
Freiburg Forest [110]	mobile robot	**stereo RGB, NIR**, NRG, NDVI, EVI	unstructured forest environments	pixel-level for: obstacle, trail, sky, grass, vegetation, void	day-time w/ shadows, various sun angles	3 days, same area	266
ViViD [62]	vehicle	**stereo RGB, LWIR**, event, LIDAR, IMU	indoor and outdoor trajectories	N/A	day and night-time	same area	8k
FieldSAFE [57]	tractor	**stereo RGB, LWIR**, webcam, 360° camera, LIDAR, radar, IMU, GNSS	grass field	geo coordinates and labels for obstacles	day-time w/ glare & shadows	2 hours, single day, same area	?
CATS [106]	hand-held	**stereo RGB, stereo LWIR**	static outdoor	N/A	day-time	10 different locations	100
PST900 [96]	hand-held	**stereo RGB, LWIR**	underground	pixel-level for: background, fire extinguisher, backpack, drill, survivor	poor illumination & visibility	single location	894
Brno Urban [65]	car	**RGB, LWIR, 3D LIDAR**, IMU, GNSS	urban, suburban & country roads	N/A	day-time	375.7 km, 10 hours	?

Table A.1: RGB-D-IR datasets

	Platform	Sensors	Scene	Annotation	Conditions	Diversity	Size
Freiburg Thermal [113]	car	RGB, LWIR	regular traffic	pixel-level for: road, sidewalk, building, curb, fence, pole, vegetation, terrain, sky, person, car, bicycle, background	day-time, good weather	diverse driving scenarios, same city	10k+
KAIST pedestrian [47]	car	RGB, LWIR	regular traffic	bounding boxes for: person, people, cyclist	day and night time	3 types of roads / locations	95k
ThermalWorld VOC [55]	hand-held	RGB, LWIR	indoor and outdoor	pixel-level for: person, car, truck, van, bus, building, cat, dog, tram, boat	different temperatures and weathers	different cities, all seasons	5098
FLIR ADAS	vehicle	RGB, LWIR	regular traffic	bounding boxes for: car, bike, person, dog, other vehicle	day and night-time	same city, several months	14k
RGB-NIR Scene [13]	fixed	RGB, NIR	static indoor & outdoor	N/A	day-time, good weather	wide range of scenes / locations	954
MIR Semantic Segmentation [38]	mobile cart	RGB, LWIR	traffic scenes	pixel-level for: car, person, bike, curve, car stop, guardrail, cone, bump	day and night-time	same city	1569
MIR Object Detection [104]	mobile cart	RGB, NIR, MWIR, LWIR	traffic scenes	bounding boxes for: car, person, bike, curve, car stop, guardrail, cone, bump	day and night-time, motion blur	same city	7521
Driveable region [116]	vehicle	RGB, LWIR	campus roads	pixel-level driveable region	night-time	3 types of roads / locations	191

Table A.2: RGB-IR datasets

	Platform	Sensors	Scene	Annotation	Conditions	Diversity	Size
IDD Multimodal [111]	car	RGB, LIDAR, GPS	unstructured traffic in India	N/A	daytime, different weathers, illumination and air quality	2 cities	10k+
DIML/CVLAB [17]	hand-held	stereo RGB	indoor & outdoor	N/A	daytime, variations in illumination	diverse scenes and perspectives	1M
SYNTHIA video [89]	simulation	RGB, depth	virtual realistic urban environment	pixel-wise for: misc, sky, building, road, sidewalk, fence, vegetation, pole, car, sign, pedestrian, cyclist, lane-marking	different weathers, illuminations, seasons	7 sequences, different settings and cities	560k
Cityscapes [21]	car	stereo RGB, odometry, GPS, temp.	traffic scenes	pixel-wise for 30 classes, 8 categories: flat surfaces, humans, vehicles, constructions, objects, nature, sky, void	daytime, good weather	50 cities, several months	3k+
Kitti [30]	car	RGB, LIDAR	traffic scenes incl. rural areas & highways	same as Cityscapes	daytime, good weather	same city	200
RADIATE [94]	car	stereo RGB, LIDAR, RADAR, GPS/IMU	traffic scenes	bounding boxes for: car, van, truck, bus, motorbike, bicycle, pedestrian	sunny, overcast, night, rain, fog, snow	same city	200k+
Lost and Found [85]	car	stereo RGB	road with small obstacles	coarse pixel-wise for road & different obstacle categories	daytime, good weather	same city	2k+

Table A.3: RGB-D datasets

A.3 Depth completion

We first define thresholds for the pixel-wise depth i (in meters) to distinguish between invalid pixels (eg. $i < 0.1$ for Cityscapes), near pixels ($0.1 < i \leq 15.0$), mid-range pixels ($15.0 < i \leq 30.0$) and far pixels ($30.0 < i$). We also define a ROI beyond which any depth value is treated as invalid. These thresholds and ROI are manually adjusted for each dataset (eg. for Cityscapes, we exclude the front of the ego-vehicle and the top of the image from the ROI, since they always contain invalid or missing values). Figure A.1 illustrates these zones, followed by the steps in the depth completion step, which we detail them below using the same numbering:

1. select valid pixels, and invert + offset them such that they remain much larger than invalid ones during the whole process
2. select near, mid, and far pixels and dilate them with different cross kernels (3×3 for far, 5×5 for mid, 7×7 for near); then combine the dilated maps, starting from farthest to nearest
3. small hole closure with full 5×5 kernel, followed by 5×5 median blur to remove outliers
4. fill holes with masked dilations (full 9×9 kernel)
5. extrapolate from the edges of the ROI: starting from the top edge, then bottom, left, and right
6. fill holes with masked dilations (full 5×5 kernel) - repeated 6 times
7. median blur followed by bilateral blur for valid pixels
8. final hole filling step with with masked dilations (full 51×51 kernel)
9. invert and offset (to revert the 1st step)
10. finito :D

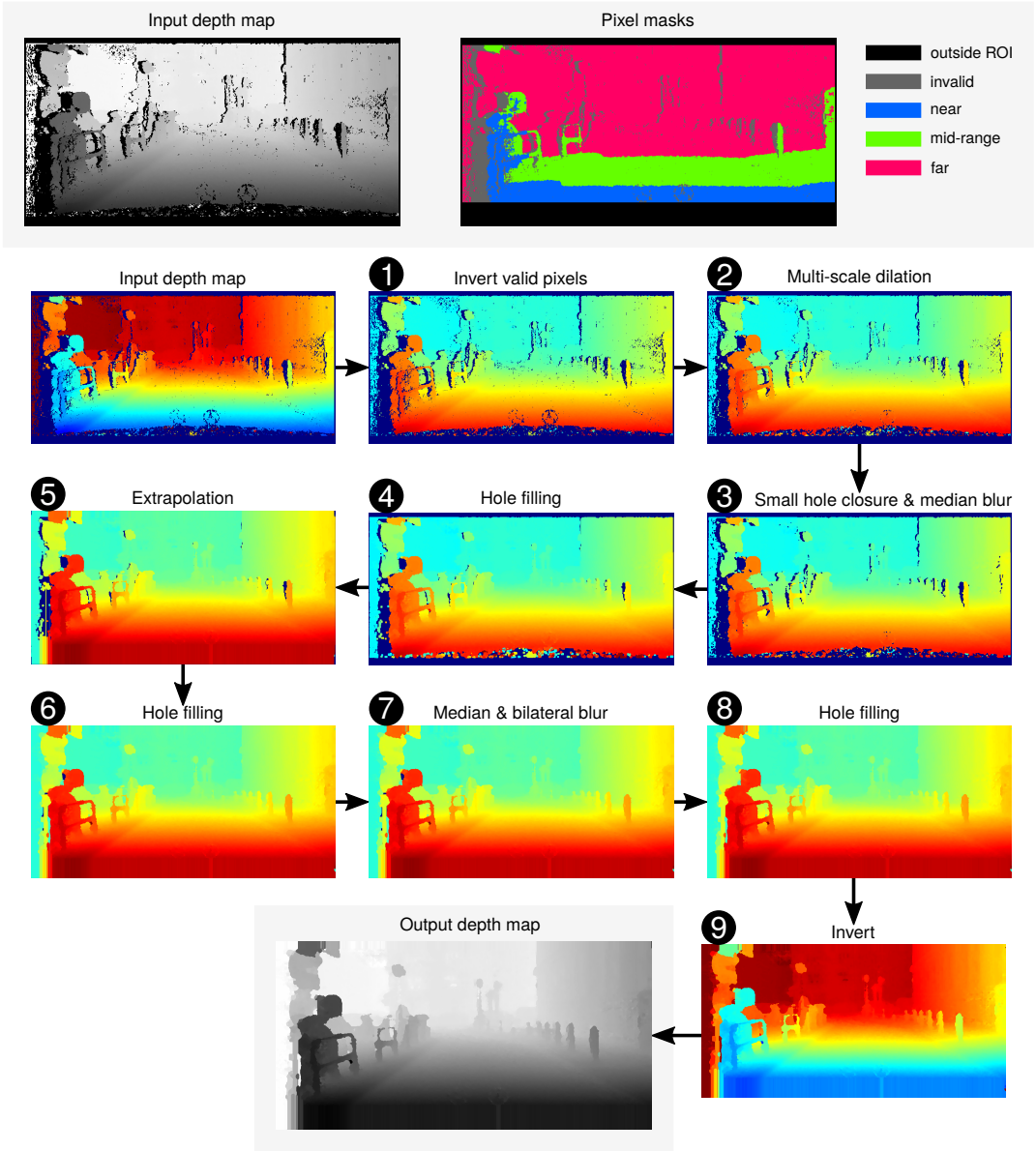
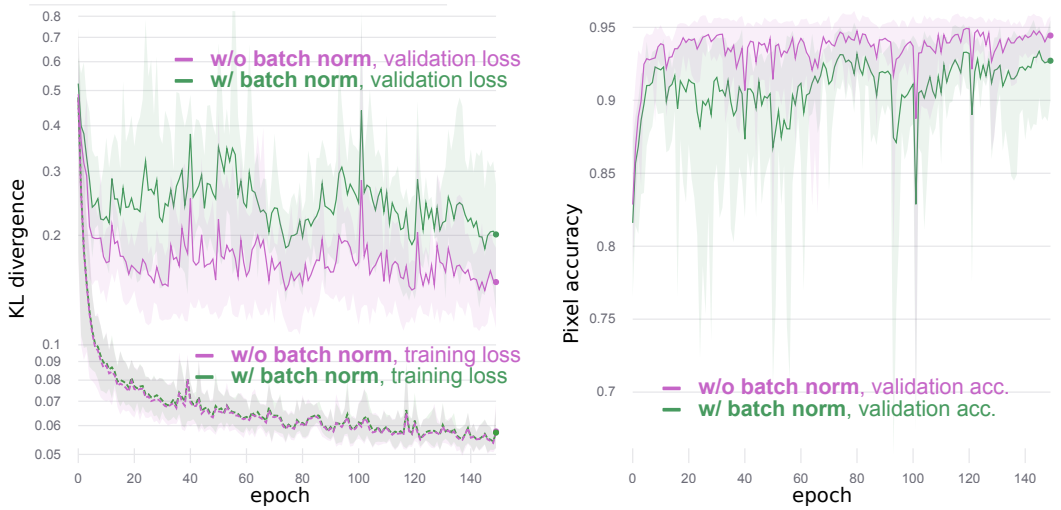


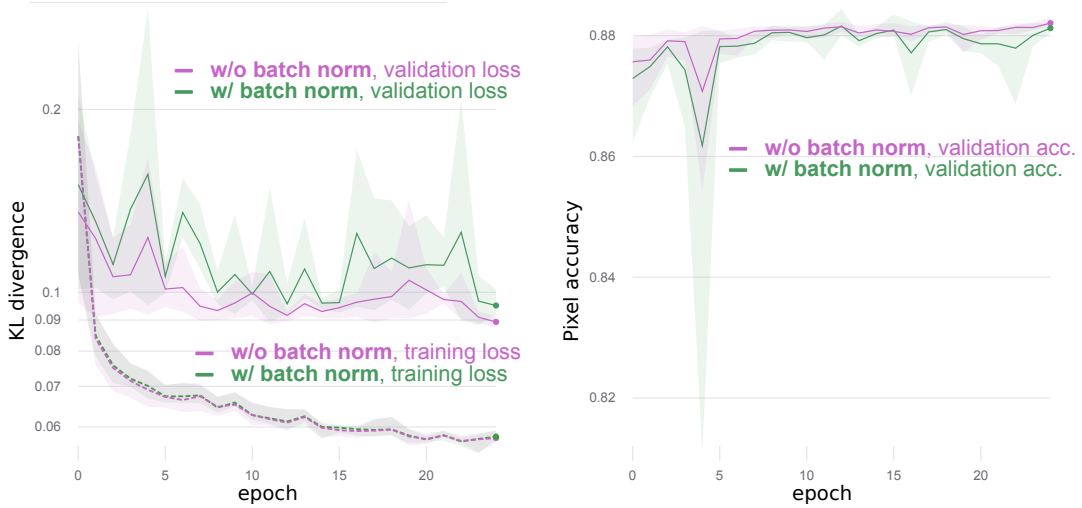
Figure A.1: Steps in the depth completion process, illustrated with a sample from the Cityscapes dataset. Depth maps in the intermediate steps are colorized for visualization.

A.4 Effect of batch normalization when training SSMA fusion units

As evidenced by the model learning curves in Figure A.2, omitting the batch normalization layer from the SSMA unit (presented in [109]) improves validation loss stability and pixel accuracy.



(a) Freiburg dataset (12 models)



(b) Cityscapes dataset (8 models)

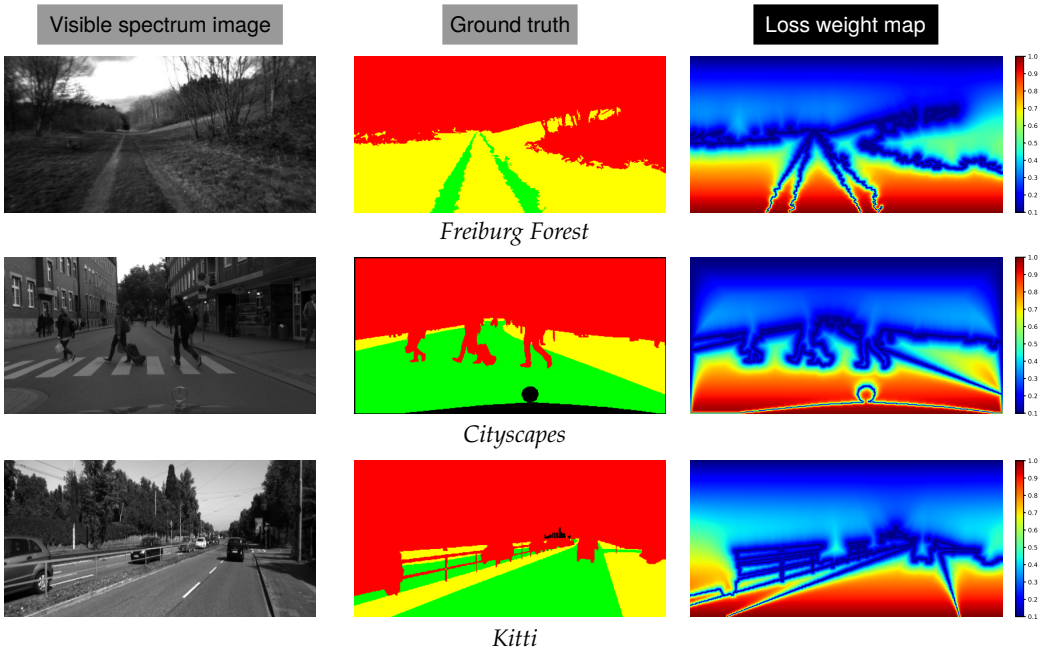
Figure A.2: Learning curves of the middle and dual architecture variants on different modality combinations (cf. Table 8.2). The models are grouped based on whether the $SSMA_{\text{Mid}}$ fusion unit applies batch normalization after its last convolution layer. We show the mean and min-max range for the loss (left) and pixel accuracy (right) per epoch. The loss is plotted on a log-scale for clarity.

A.5 Benchmarking - implementation details

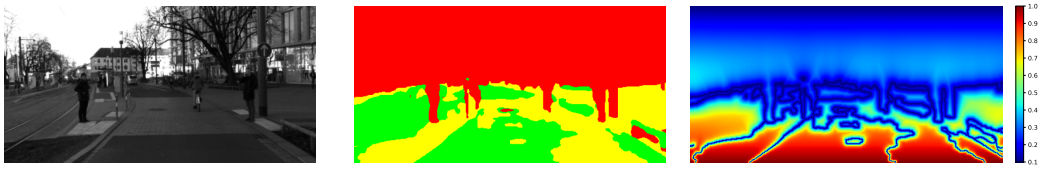
Inference time For measuring and comparing inference time of the different segmentation architectures in our evaluation, we use Pytorch's Benchmark module[1], which automatically handles warm-up and CUDA synchronization. Measurements are performed with the models in evaluation mode (no gradient calculation, drop-out layer, or computation of running statistics for batch normalization). Input tensors are generated randomly, with batch size of 1, channel size equal to the number of modalities, and height $\times$ width of 240×480 (the same image size used in all our experiments) and values ranging from 0 to 255. For a given model, the runtime of a forward pass is repeatedly measured until the variance is low enough to be confident in the measurement.

Memory consumption We make use of Pytorch's Profiler[2] which reports the amount of (CPU or CUDA) memory used by the model's tensors. For each model, measurements are taken for a forward pass on a single input sample, since the amount of memory allocated is not influenced by system activity.

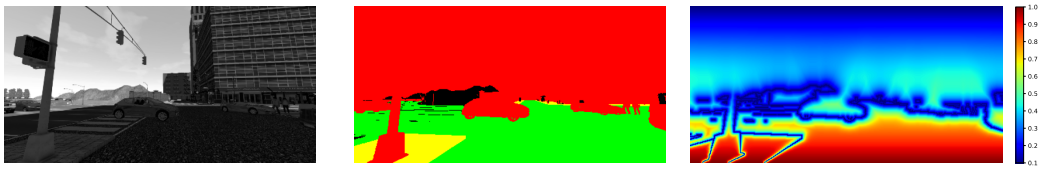
A.6 Examples of loss weight maps



[1] https://pytorch.org/docs/stable/benchmark_utils.html
[2] https://pytorch.org/docs/stable/autograd.html#profiler

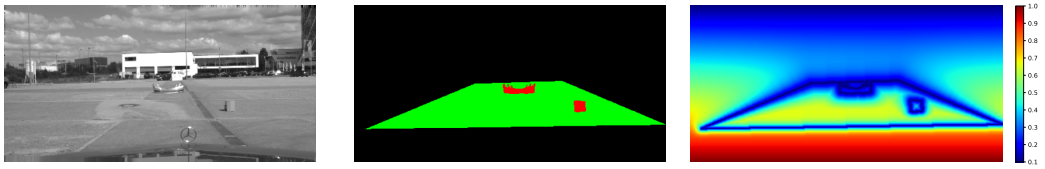


Freiburg Thermal

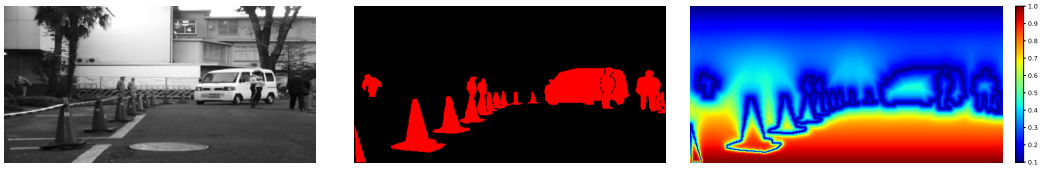


Synthia

KAIST Pedestrian (own annotation)



Lost and Found



MIR Multispectral Segmentation

ThermalWorld VOC

Figure A.2: Example of a loss weight map generated for each dataset in our experiments. The loss weight map is only generated based on the ground truth segmentation mask; the input image is shown for reference only.

A.7 Demo videos

We showcase some of our main results on publicly available video sequences for qualitative comparison. Links and details for each video are included below.

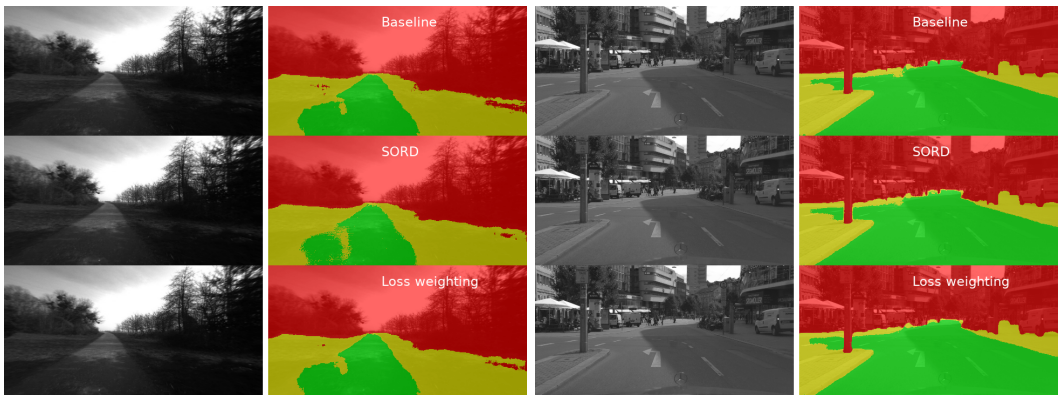
A.7.1 Visible spectrum models

We compare the following 3 visible spectrum models from our evaluation:

- the baseline model from Section 6.3.4, trained to learn driveability via transfer learning with **standard one-hot labels**
- the SORD $SLD_{\alpha=1}$ model from Section 6.3.4, trained to learn driveability via transfer learning with **soft ordinal labels**
- the loss weighting model from Section 6.4.3, trained to learn driveability via transfer learning with **one-hot labels** and our **pixel-wise loss weighting** scheme

Freiburg Forest We show predictions on RGB video sequences from Freiburg Forest Raw[3]. This is the raw large-scale (non-annotated) dataset from which the authors selected and annotated the samples which ended up in Freiburg Forest. Note that the model was trained from scratch on less than 230 images.

Cityscapes We use the RGB video sequences from the `leftImg8bit_demoVideo` set[4]; similarly to Freiburg Forest Raw, these are sequences from which a small number of samples were selected for annotation in the Cityscapes semantic benchmark. The model was trained from scratch on approximately 3000 images.



(a) Freiburg Forest: `https://youtu.be/R2zTY3hGKQg`

(b) Cityscapes: `https://youtu.be/n8BYNp3wfH4`

Figure A.3: Preview and links to video demos for the visible spectrum models

[3] `http://deepscene.cs.uni-freiburg.de/`
[4] `https://www.cityscapes-dataset.com/downloads/`

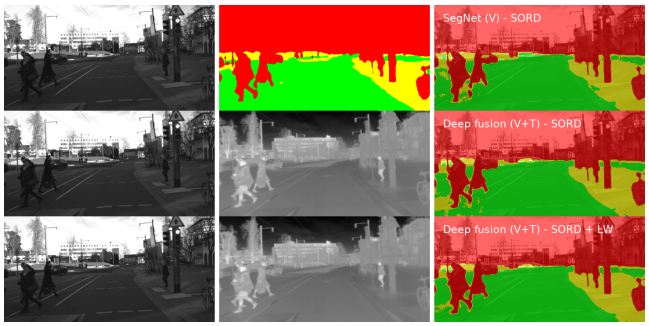
A.7.2 Visible+Thermal fusion models

Freiburg Thermal We compare 3 models from our final visible-thermal fusion experiment in Section 9.3.2:

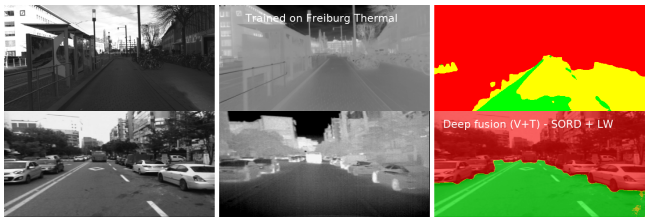
- the baseline $V_{\text{FreiburgThermal}}$ SegNet model, trained to learn driveability from visible spectrum (V) images via transfer learning with **soft ordinal labels** ($SLD_{\alpha=1}$)
- the $V_{\text{FreiburgThermal}}$ + $T_{\text{FreiburgThermal}}$ $dual_{\text{Custom}}$ fusion model trained to learn driveability from a combination of visible spectrum (V) and thermal (T) images with **soft ordinal labels** ($SLD_{\alpha=1}$)
- the $V_{\text{FreiburgThermal}}$ + $T_{\text{FreiburgThermal}}$ $dual_{\text{Custom}}$ fusion model trained to learn driveability from a combination of visible spectrum (V) and thermal (T) images with **soft ordinal labels** ($SLD_{\alpha=1}$) combined with **pixel-wise loss weighting**

Since the dataset was captured as a video sequence, we simply show prediction on the test set. Note that the video includes the ground truth for reference, shown in the top-center, to highlight the fact that the network was trained on very approximate (and often incorrect) segmentation masks generated by an RGB teacher network (cf. [113]).

KAIST Pedestrian Lastly, we show predictions by the $V_{\text{FreiburgThermal}}$ + $T_{\text{FreiburgThermal}}$ $dual_{\text{Custom}}$ model on out-of-dataset video sequences from KAIST Pedestrian[5] (cf. the dataset overview in Section 5.1.2). We select day-time sequences from the test set. In the video, we show samples from the Freiburg Thermal dataset on the top row, to highlight the fact that the model was trained on approximate ground truth data and different thermal sensor characteristics, making this a challenging task.



(a) Freiburg Thermal:
`https://youtu.be/b-r5lAvPwr8`



(b) KAIST Pedestrian:
`https://youtu.be/aJxhqTdemCQ`

Figure A.4: Preview and links to video demos for the visible spectrum + thermal fusion models

[5] `https://soonminhwang.github.io/rgbt-ped-detection/`