Technische Universität München
Photogrammetrie und Fernerkundung
Prof. Dr.-Ing. U. Stilla

Domain adaptation of HDR training data for semantic road scene segmentation by deep learning

Michael Weiher

Master's Thesis

Bearbeitung:	01.04.2019 – 30.09.2019
Studiengang:	Geodäsie und Geoinformation (Master)
Betreuer:	Prof. Dr.-Ing. Uwe Stilla Christian Albrecht, M.Sc. Alexander Hanel, M.Sc. (BIT Technology Solutions GmbH)
Kooperation:	BIT TECHNOLOGY SOLUTIONS

2019

Abstract

Training and validating systems which involve the use of artificial intelligence (AI) is an important task for many applications. Especially automotive applications require to be safe and reliable. In this thesis it is aimed to improve the safety of such systems demonstrated by the performance of a semantic segmentation network. This allows a computer to gain a comprehensive scene understanding by classifying and localizing important semantic classes like pedestrians or road in an image. High dynamic range (HDR) images contain more brightness information than regular images, which are considered as low dynamic range (LDR) images. They are able to display scenes with a much higher contrast. In this thesis it is hypothesized that HDR images are a remarkable component to achieve an increased semantic segmentation performance. This will be demonstrated by three approaches. The first one allows to generate a set of LDR from HDR images with different tone mapping operators. Additionally, there is proposed a ranking method to estimate the semantic segmentation performance for each of them. The second approach enables to evaluate semantic segmentation networks trained on LDR and HDR datasets. The third attempt aims to improve the performance on a real-world test set, as the training set consists of solely synthetic images. This is approached with a domain adaptation. In this thesis it is shown that components of the proposed ranking method are reasonably correlated with the semantic segmentation performances. Furthermore, experiments demonstrate that HDR images outperform LDR images for a semantic segmentation of real-world scenes. In addition, it is demonstrated that HDR images are less sensitive to different domains.

Kurzfassung

Das Trainieren und Testen von Systemen, welche auf dem Prinzip von Künstlicher Intelligenz (KI) beruhen, bemisst sich einer hohen Bedeutung für viele Anwendungen. Insbesondere im Automobilbereich ist ein hohes Maß an Sicherheit und Zuverlässigkeit gefordert. Ziel dieser Arbeit ist es die Sicherheit solcher Systeme zu verbessern, untersucht anhand der Leistung eines Neuronalen Netzwerkes zur semantischen Segmentierung. Dieses ermöglicht dem Computer ein Szenen-Verständnis, durch die Klassifizierung und Lokalisierung wichtiger semantischer Klassen, wie Fußgänger oder Fahrbahn, in einem Bild. High Dynamic Range (HDR) Bilder oder Hochkontrastbilder enthalten mehr Farbinformationen als herkömmliche Bilder, welche deshalb auch als Low Dynamic Range (LDR) Bilder bezeichnet werden. In dieser Untersuchung wird angenommen, dass HDR Bilder zur Verbesserung von semantischen Segmentierungs-Netzwerken beitragen. Dies wird anhand von drei Ansätzen demonstriert. Der Erste ermöglicht es, einen Datensatz von LDR Bildern durch verschiedene *tone mapping* Operatoren aus einem HDR Datensatz zu erzeugen. Des Weiteren wird eine ranking-Methode vorgestellt, welche Erwartungswerte für die semantische Segmentierungs-Leistung liefert. Der zweite Ansatz erlaubt die Evaluierung von semantischen Segmentierungs-Netzen, welche auf LDR und HDR Bildern trainiert worden sind. Durch den dritten Ansatz wird versucht, die Segmentierungs-Leistung für reale Testbilder zu erhören, da die Trainingsdatensätze aus synthetischen Bildern bestehen. Dies wird durch eine *domain adaptation* umgesetzt. In dieser Arbeit wird gezeigt, dass einzelne Komponenten der vorgestellten ranking-Methode eine deutliche Korrelation mit den Ergebnissen aus der semantischen Segmentierung aufweisen. Außerdem zeigen die Ergebnisse, dass HDR Bilder eine höhere Segmentierungs-Leistung für reale Testbilder liefern, als LDR Bilder. Zusätzlich konnte gezeigt werden, dass HDR Bilder weniger Domänen-spezifisch sind.

Contents

List of Abbreviations

AI	artificial intelligence
CNN	convolutional neural network
CRF	conditional random field
DCNN	deep convolutional neural network
DNN	deep neural network
DRN	dense relation network
FCN	fully convolutional network
GPU	graphics processing unit
HDR	high dynamic range
HDRI	high dynamic range imaging
LDR	low dynamic range
mIoU	mean intersection over union
NN	neural network
RNN	recurrent neural network
SGD	stochastic gradient descent
TMO	tone mapping operator
TMQI	tone mapped quality index

List of Figures

List of Tables

1 Introduction

1.1 Motivation and objective of the thesis

Every day there die about 3,700 people on the world's roads with tens of millions of people being injured or disabled every year [WHO, 2019]. It would be an immense improvement for safety, if vehicles were able to avoid crashes. Self-driving cars may be an answer and they are also a great alternative to people which are not able to drive on their own for reasons of age, disability or loss of driving license. Furthermore, the impact on traffic and environment might decline with increasing amount of intelligence during driving, such as acceleration or breaking behaviour and the time spent in traffic could be used more meaningful.

The path to autonomous driving is a very challenging task for the automotive industry. For this purpose, cars are equipped with different kind of sensors in order to capture all of the required information of the environment to safely guide from a starting point to a destination point. One of the utilised sensors is a camera sensor which can be seen as a substitute of the human eye for the machine. Gaining a high-level understanding from images with computers is referred to as computer vision. While it was very difficult and time-consuming to solve important computer vision tasks in the field of autonomous driving for a long time, the improvement on such tasks has increased dramatically with the ongoing trend of artificial intelligence (AI).

Neural networks (NNs) allow to automatically learn features from images instead of designing them with expert knowledge and are a great tool of using AI. In simplest scenarios like image classification they have already shown to outperform humans [He et al., 2015], as they are able to predict classes for images, where only human experts might be able to do so. In general, deep neural networks (DNNs) are able to perform near perfection on many supervised learning problems, as their millions of parameters can be continuously updated via backpropagation until the error on the training data vanishes. Nevertheless, the learned features might be very data-specific and do not necessarily incorporate a good representation for data from different sources. The associated term addressing this phenomenon is the *generalization capability* of the neural network, which describes how well the learned features can be applied on unseen images. While big companies have access to a huge amount of images for training, which makes it possible to robustly perform on a large space of situations, it has still been shown that DNNs can perform very unexpected on unusual situations, which are not part of the training set [Bolte et al., 2019]. In literature, a relevant object in relevant location which a modern autonomous driving system cannot predict is referred to as a *corner case* [Bolte et al., 2019]. An example for such a corner case could be imagined as a human in front of the autonomous driving car which gets predicted as road.

It is essential to test the components of an autonomous driving system against corner cases in order to verify the generalization capability and to guarantee the safety of the system. But in reality it is very time-consuming and expensive to collect images that can be used for the assurance of the system, as many different situations should be covered and the respective ground truth must be manually produced in order to evaluate the performance of the system. Furthermore, for many situations it is very difficult and dangerous to capture images. This could be for example a child running directly in front of the car. Capturing images of such situations and testing the computer vision system against it is essential to proof that the situation shows no corner case for the self-driving car and thus not running over the child.

Modern computer graphics allow to create synthetic images with a very high degree of realism. This makes it possible to create images of street scenes by modelling 3-dimensional objects and assembling them together into a scene. This enables the generation of images of very dangerous situations in a secure environment and allows to create error-free pixel-wise ground truth annotations on the fly. Using these images to test the autonomous driving system for functionality against corner cases may be an useful addition to real-world images. It may be also a good complement for the training of the system itself, as scenes in numerous variations can be generated. Another problem which needs to be addressed for the safety of autonomous driving systems is shown in figure 1.1. It shows a typical scenario where displaying the scene content is limited by the dynamic range of images. The figure shows subsequent frames of a camera video, which was recorded during a passage through a tunnel. Regular images are heavily limited by the number of different grey values for each color channel and are therefore also known as *low dynamic range (LDR)* images. For certain situations the dynamic range might not be high enough. For example when leaving a tunnel with the car, as seen in the images. Thereby, the contrast of the scene is too high, which makes the system blind for what's happening outside of the tunnel. *High dynamic range (HDR)* images are able to represent much more different grey values and overcome the limitations of LDR images.

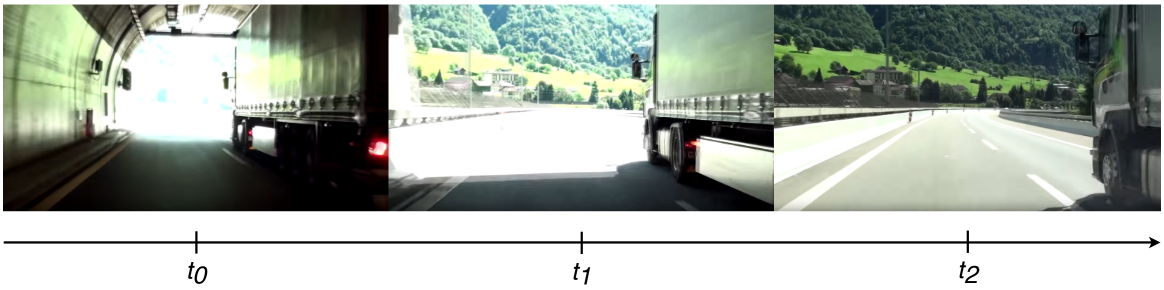


Figure 1.1 The figure shows the limitation of low dynamic range images. The frames were extracted from a video captured by a regular camera sensor and are shown in temporal sequence, as a car is leaving a tunnel. At time step t_0 the camera is blended by the high contrast of the scene. At time step t_1, directly before leaving the tunnel, the contrast is still too large to properly display the road in the image. Only at time step t_2 the image displays the scene content properly. The frames of the video material were taken online from Astey Highways [2013].

The topic of this thesis is to accomplish a complex scene understanding for autonomous driving with synthetic images for the task of semantic segmentation. Since synthetic images can be rendered from 3D computer graphics with a high dynamic range, one important question to answer is, if the high dynamic range can be used to improve the semantic segmentation performance of modern neural networks. Furthermore, HDR images can be used to generate LDR images, which is done with so called tone mapping operators (TMOs). Through tone mapping the high dynamic range image gets compressed into a low dynamic range image of the same scene. In this work multiple operators will be used for the tone mapping, to answer which ones generate the best performing LDR images for a semantic segmentation. To answer these questions a highly realistic synthetic dataset will be used. It has already been shown, that neural networks perform modest on real-world images if they were trained on synthetic ones [Hoffman et al., 2016]. One common approach to improve the semantic segmentation performance is by carrying out a domain adaptation, which tries to align the distribution of the synthetic and the real-world images. In this thesis it is aimed to use HDR and LDR images from both domains to find out, whether the performance of a semantic segmentation network can be improved through a domain adaptation, after training on a highly realistic synthetic dataset.

1.2 Main contributions

The main contributions of this work are: i) a new method that compares a set of tone mapped images from different tone mapping operators and ranks them according to their expected semantic segmentation performance (section 3.1); ii) an evaluation of the semantic segmentation performance between neural networks trained on images with high dynamic range and low dynamic range, as well as images from different tone mapping operators (section 5.2); iii) empirical studies validating the performance of a domain adaptation with synthetic and real-world HDR images (section 5.4).

1.3 Structure of the thesis

The structure of this work is organized as follows. First, in chapter 2 the state of the art of high dynamic range imaging (HDRI), tone mapping and the evaluation of tone mapping are introduced. Other concepts like the semantic segmentation and the domain adaptation using DNNs are also reviewed. Next, chapter 3 describes the methodology of this work, discussing the methods which are used for the experiments. Chapter 4 provides a description of the experiments, which are carried out in this thesis and the results are being provided in chapter 5. Thereafter, the experimental results are discussed in chapter 6. At last, chapter 7 summarizes the work and describes possible future work.

2 State of the Art

This chapter covers a detailed look on concepts and related published methods for this work. The first section contains information about high dynamic range images and explains the difference to standard low dynamic range images. In the following, semantic segmentation with deep neural networks will be introduced and finally in section 2.3 it will be shown how a domain adaptation can be accomplished with neural networks.

2.1 High dynamic range imaging

There are a lot of applications making use of digital images. In the field of autonomous driving images are for example used to accomplish a scene understanding. With that, the computer vision system will be enabled to recognize and distinguish between different kind of objects in the scene, like pedestrians, traffic signs or the road. But while many real-world scenes have a large amount of brightness variation, a typical digital camera provides only 8 bits (256 levels) of brightness information for each color channel of every pixel. This might be inadequate to describe many scenes and therefore may harm the scene understanding capability. This section explains the difference between low and high dynamic range images and describes how HDR images can be generated. Furthermore, methods to transform images from HDR to LDR will be presented and how the transformation can be evaluated.

2.1.1 Low vs. high dynamic range imaging

The dynamic range of an image is defined as the ratio between the lightest and darkest pixel [Reinhard et al., 2010]. The dynamic range of a display is correspondingly defined as the ratio of the minimum and maximum luminance that can be emitted [Reinhard et al., 2010]. In figure 2.1 it is shown that the real world exhibits a wide luminance range [Mantiuk et al., 2015]. For instance, the sun at midday might be 100 million times brighter than starlight [Reinhard et al., 2010]. In contrast, the human visual system is only capable of perceiving luminance values in a range between 4 orders of magnitude [Mantiuk, 2013]. In addition to that, a conventional display can only reproduce a range of luminances by 2 orders of magnitude.

The range of conventional images is limited to 256 integer values for each of the red, green and blue color channel and is called *low dynamic range* or simply *LDR* image [Reinhard et al., 2010]. This number is unsuitable to represent many scenes [Reinhard et al., 2010]. *High dynamic range imaging* or *HDRI* overcomes those limitations, as it allows to represent all colors that can be perceived by the human eye in the real world [Mantiuk et al., 2015]. In figure 2.2 it is attempted to show the advantage of using HDRI with an example. While the left image shows an optimally exposed conventional image, the right image was created using high dynamic range imaging techniques as described in this thesis. The difference between the two images is mainly reasoned due to the fact, that traditional imaging is not able to represent such high-contrast scenes [Mantiuk et al., 2015]. The difference would be even more clear, if the images were

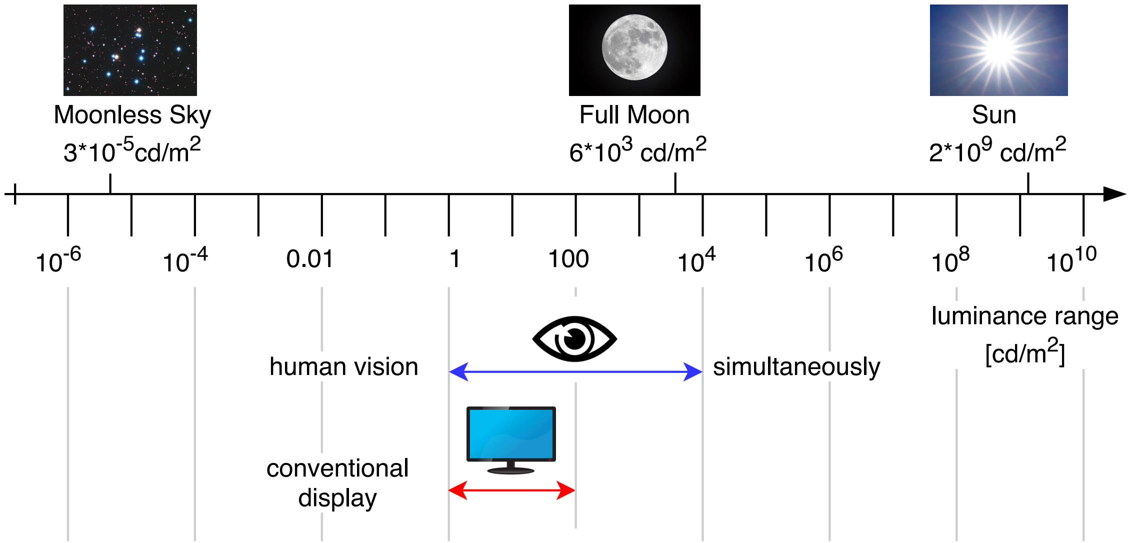


Figure 2.1 Overview of the relation between the luminance range in the real world and different components. This figure was replicated from Mantiuk [2013].

displayed on special display devices that are capable of reproducing a much larger luminance range than conventional displays or photographic prints [Reinhard et al., 2010]. But as it is not possible to show high dynamic range images without special devices, both images are LDR ones. To clarify, HDRI or high dynamic range imaging is the technique which is necessary in order to create HDR images. The resulting high dynamic range images will then contain more different color values than LDR images but also require special displays to be viewed. Every image which is displayed in this thesis will be a LDR one. However, it may be produced from an HDR image and may therefore provide a better impression of the real scene than a regular LDR image.

Figure 2.3 shows the differences between a high dynamic range and a low dynamic range image. The quality of the LDR image was reduced in order to show potential differences between the visual contents as seen on an HDR display. One difference is the number of different grey values for each color channel, which ranges from 8 to 16 bit for LDR images and up to 64 bit for high dynamic range images. The difference between the HDR and LDR image is actually more than just the color bit depth [Reinhard et al., 2010]. Most HDR images are *scene-referred* as their pixel values have a direct relation to the radiance of a scene [Reinhard et al., 2010]. This means, that the pixel values in the HDR image are linearly related to the photometric quantity luminance, which describes the perceived intensity of the light per surface area [Mantiuk et al., 2015]. In contrast, LDR images are *display-referred* because their colors are associated with an output display device. The pixel values in a low dynamic range image are non-linearly related to luminance and the term luma is used to describe them [Mantiuk et al., 2015]. Instead of a linear relationship the pixel values are usually corrected by a power function in the form of $signal = intensity^{(1/\gamma)}$, where the value of γ lies typically between 1.8 and 2.8 [Mantiuk et al., 2015]. This so called inverse gamma correction is performed in order to compensate for the gamma correction in the form of $intensity = signal^{\gamma}$, which is performed by CRT or LCD monitors [Mantiuk et al., 2015]. The complete transformation from a high dynamic range image stored in a scene-referred representation to a LDR image stored in a display-referred representation is called *tone mapping*

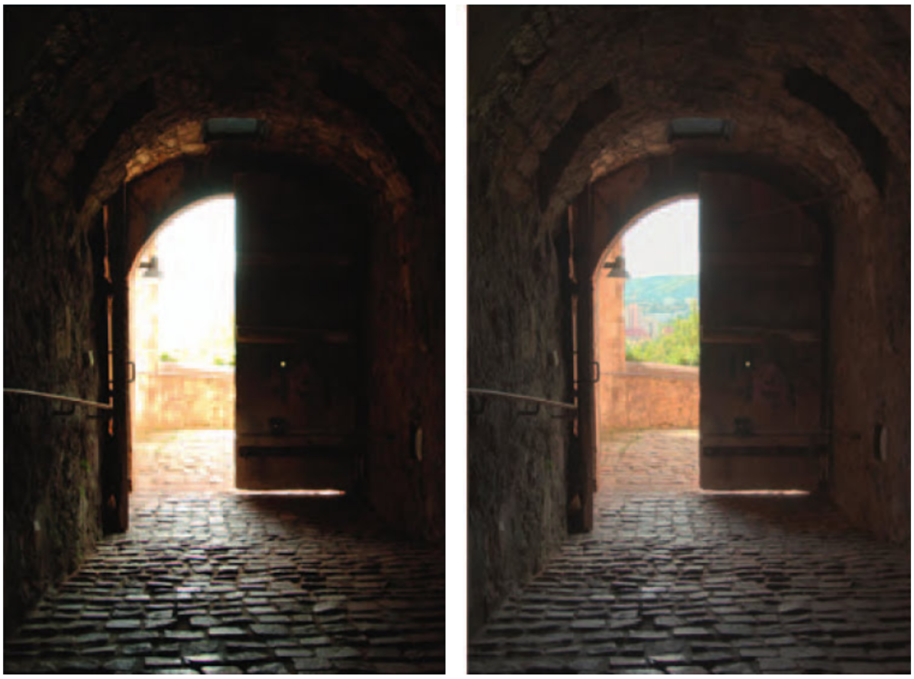
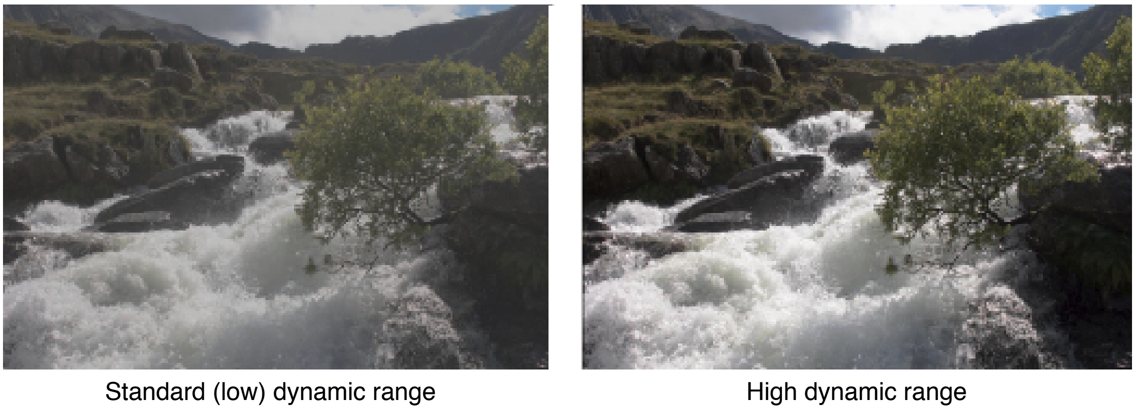


Figure 2.2 Reinhard et al. [2010] shows the benefit of high dynamic range imaging. Left can be seen an optimally exposed conventional image. On the right side, the image was generated with techniques (tone mapping) described in this work from an HDR image. Nevertheless, both images are LDR ones.

[Reinhard et al., 2010]. It transforms the HDR image into a low dynamic range image, which can be shown on a computer display [Mantiuk et al., 2015]. The transformation itself is performed with a tone mapping operator (TMO).

Figure 2.4 provides a complete overview of the high dynamic range imaging pipeline and available HDR technologies [Mantiuk et al., 2015]. Taking an image of a real scene with a conventional camera will produce an image stored in the LDR format. This image can be viewed offhand on a conventional display. In contrast, taking an image of the same scene with an HDR camera will create an HDR image of that scene, which can be viewed on an HDR display without further ado. In order to view a high dynamic range image on a conventional display it has to be tone mapped.



Standard (low) dynamic range

High dynamic range

Figure 2.3 Imitation of the difference between a low dynamic range and a high dynamic range image by purposefully reducing the quality of the left image [Mantiuk et al., 2015].

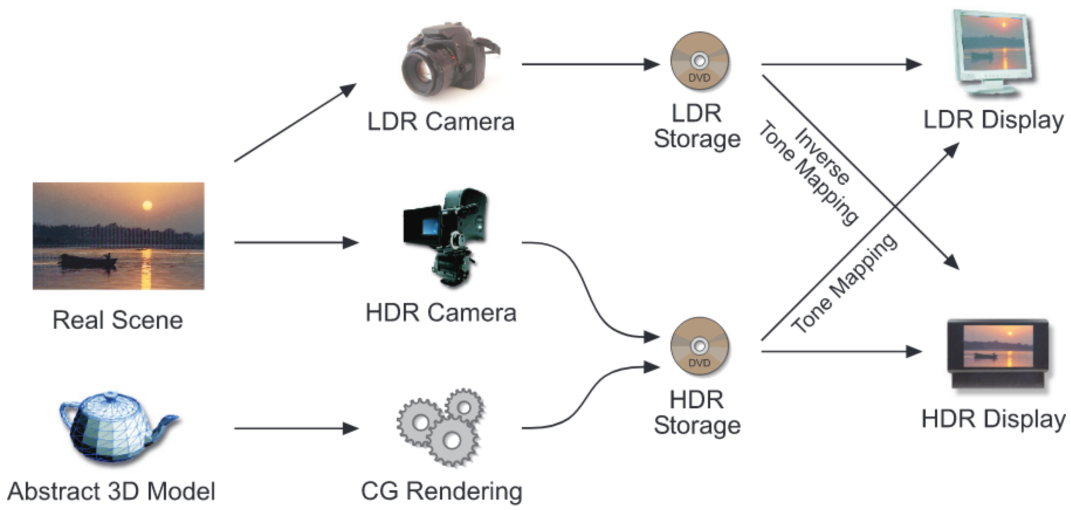


Figure 2.4 Overview of the high dynamic range imaging pipeline (adapted from Mantiuk et al. [2015]). HDR images can be either captured from real scenes using camera sensors or using 3D models and computer graphics (CG) rendering techniques.

The process of recovering the high dynamic range of a tone mapped LDR image is called inverse tone mapping and tries to approximate the lost information in the image. Another approach to generate HDR images is by utilizing computer graphics, where abstract 3D models are used to generate the high dynamic range images via rendering techniques.

2.1.2 Approaches for high dynamic range imaging

High dynamic range images can be either captured from real scenes or via rendering using 3D computer graphics [Reinhard et al., 2010; Mantiuk et al., 2015]. The topic, especially when using computer graphics, is very large. Therefore this section provides a basic overview of high dynamic range imaging techniques.

HDR images from camera sensors

There are different possibilities to get a high dynamic range image from a real scene. When using a conventional camera equipment, relying on a 12 bit or 14 bit RAW mode will only end up recording the noise more precisely. Therefore it is essential to use multiple exposures in order to generate an HDR image with a non-professional equipment [Reinhard et al., 2010].

According to Reinhard et al. [2010], creating a high dynamic range image from multiple LDR images requires several steps: First, the scene and camera should be completely static. Then multiple images of that scene can be recorded with different exposures. Another necessary step is to invert the system response to recover a linear relation between scene radiances and pixel values. Afterwards, each exposure can be brought into the same domain by dividing every pixel in the image by it's exposure time. The linear exposures have to be averaged in order to generate the HDR image. However, the lightest and darkest pixels should be excluded as they are under- or overexposed. The question therefore is how the pixels in between should be weighted.

There are many methods available [e.g. Debevec & Malik, 2008; Granados et al., 2010; De Neve et al., 2009] that are able to finally create the HDR image from the linear pixel values, each using different weighting functions. Which one to choose depends on various factors, which can be different for many kind of applications.

Reinhard et al. [2010] name several disadvantages in this high dynamic range imaging pipeline. Hence, even slight camera shifts between the exposures result in a blurring of the HDR image, which can be eliminated by image alignment methods (e.g. SIFT, SURF, BRIEF, BRISK). If the camera response function is unknown, it has to be estimated, using one of many available methods [e.g Ng et al., 2007; Grossberg & Nayar, 2003]. Another problem is that objects may move in the scene between exposures, which causes errors in the final HDR image. Even these effects can be eliminated using deghosting algorithms.

Of course, it would be easier to record the high dynamic range of a scene in a single shot. According to Mantiuk et al. [2015], the simplest approach for a single-shot HDR camera is to introduce sensor sensitivity. This method does not require novel sensor design, but affects the spatial resolution negatively. However, HDR sensors can also be explicitly designed. There are several commercially available HDR video cameras based on sensors that do not require exposure time control [Mantiuk et al., 2015].

HDR images from computer graphics

The definitions and statements in this section were taken from the textbook "Physically based rendering: From theory to implementation" from Pharr et al. [2016]. The process of generating images from the description of a 3D scene is called *rendering*. There are physically based approaches that try to simulate the reality by modelling the interaction of light and matter using the principles of physics. Nearly all rendering systems that aim to create photorealistic images are based on the ray-tracing algorithm. Here, the path of a ray of light is followed through a scene as it intersects with objects. There are various components which have to be implemented in such a ray-tracing system:

- *Camera*: Positioning of a camera that determines from where the scene is being viewed. Simulation of a specific camera model, which is in the simplest case a pinhole camera.
- *Ray-object intersections*: Exact determination of the position where a ray intersects with an object in the scene. Additionally, specific properties, like material or surface normal, must be determined at the intersection point.
- *Light sources*: Modelling of the lighting inside the scene, in particular position and the way of energy distribution of the lights.
- *Visibility*: Construction of the ray from a surface to the light determines whether a point on the surface inherits energy from the light source.
- *Surface scattering*: Describes how surfaces from objects interact with light, typically by a set of parameters in order to simulate a variety of appearances.
- *Indirect light transport*: Enclosure of indirect specular reflection and transmission. Reflection of the ray, e.g. about the surface normal when hitting a mirror, to recursively find the amount of arriving light at a specific point.
- *Ray propagation*: Designation how the energy of rays changes when passing through space. There are different behaviours for rays in many environments, for example, when traced through fog, smoke or the Earth's atmosphere.

According to Mantiuk et al. [2015], computer graphics are nowadays a very important source of HDR content, as the rendering can be executed with a floating point precision and even though physically-based lighting simulations are mostly ignored, the generated images look conceivable.

2.1.3 Tone mapping of high dynamic range images

The dynamic range of illumination of a real-world scene can be extremely large, especially when the scene includes an outdoor area illuminated by sunlight and an indoor scene with much fewer illumination [Reinhard et al., 2010]. Techniques as discussed in the subsection 2.1.2 can be used in order to capture this high range with full precision.

Since the fact, that most displays emit light only in a very fixed range, but HDR images contain rather wide-ranging scene luminances, there is the need for a conversion in order to display HDR images on typical monitors. Tone mapping refers to the mapping of the potentially high dynamic range of a real world scene to the low dynamic range of a photographic print or a screen [Reinhard et al., 2002] or respectively to the mapping from HDR to LDR images. The mapping itself is carried out with a so called tone mapping operator (TMO). Looking at figure 2.2 it can be noted, that the right image was generated by such a tone mapping operation for the best possible presentation of the high dynamic range from the observed scene in the respective low dynamic range image. Figure 2.5 shows the tone mapping problem prepared from Tumblin & Rushmeier [1993]. They defined the goal of tone mapping by building a general framework. Consequently, the desired tone mapping operator causes the closest match between a real-world observer, which is a mathematical model of the human visual system, looking at a real-world scene and the tone mapped image being viewed on a display device. While the display device converts display input values to viewed luminance values, the human observer converts luminance values into perceived brightness values. Accordingly, the tone mapping operator has to be defined such, that the resulting tone mapped image looks as close as possible on the display to the human

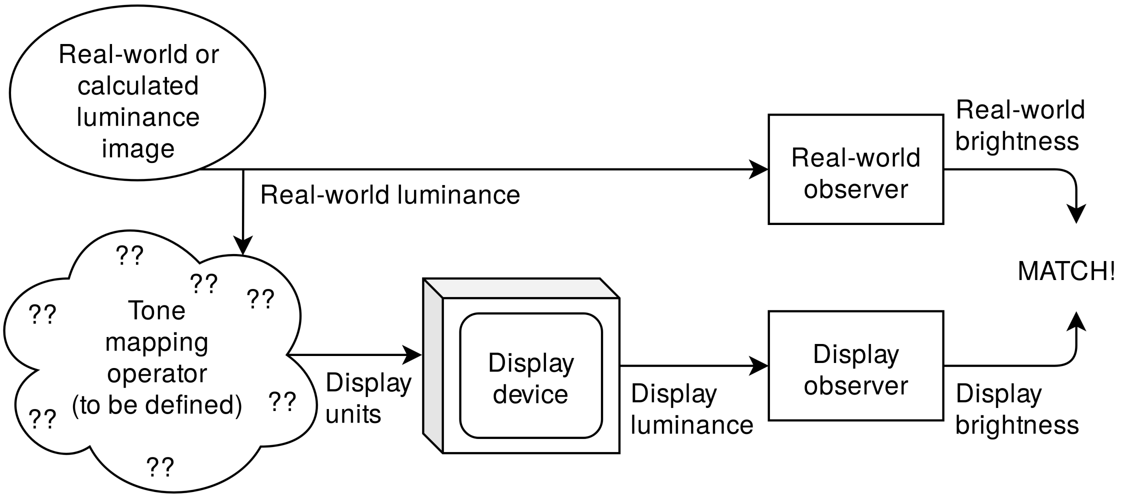


Figure 2.5 Demonstration of the tone mapping problem, replicated from Tumblin & Rushmeier [1993]. The aim is to find a tone mapping operator, which causes the closest match between real-world and display brightness.

observer, as the same observer looking at the real scene.

According to Devlin [2002], the first experiments on tone mapping were carried out in the mid 1980s to match perceived brightness values from a real scene to the brightness values of the corresponding displayed image. The large amount of work on tone mapping has been already addressed by Mantiuk et al. [2015], who classified tone mapping operators by their intents as recognized from Eilertsen et al. [2016]:

- *Visual system simulators*: Refers to tone mapping operators, which try to simulate the limitations and properties of the human visual system. The operator could, for example, try to simulate limitations of the human vision at night.
- *Scene reproduction*: These operators try to preserve the original scene appearance. This refers especially to the contrast, sharpness and colors of the image being viewed on a display device and matches with the tone mapping definition from Tumblin & Rushmeier [1993]. Instead of simulating the limitations of the visual system, they focus on achieving the best match between display output and the original scene.
- *Best subjective quality*: The goal of such operators is to generate the most preferred images for subjective preferences or artistic goals.

Devlin [2002] provides an overview on existing tone mapping operators for HDR-images and classifies them as the following:

- Spatial dependent
  - *Spatially uniform*: Also known as *global* operators. Spatially uniform TMOs apply the same transformation to each pixel in the image.
  - *Spatially varying*: Also known as *local* operators. Spatially varying TMOs apply different transformations to different parts of an image. Those have the disadvantage that they can cause artifacts around high contrast edges.
- Time dependent
  - *Time independent*: Many tone mapping operators are independent of time. They try to find one specific mapping from one image to another.
  - *Time dependent*: Time dependent tone mapping operators either try to find more than just one mapping for a single image and thus generate an animation as a result or the other way around, i.e. to find one specific mapping for a sequence of images. The later one is also referred to as *tone mapping for HDR-video* and is a more complex process than the tone mapping of single HDR-images.

Tone mapping can therefore not only be classified by their intents but also by their mathematical approach or their dependency on time. It can be seen that there has been done a lot of work on tone mapping and that it is non-trivial to choose an operator for a specific task. To tackle this problem various evaluation methods were developed in order to find the operator which satisfies one's demand the most.

2.1.4 Tone mapping evaluation methods

There are different strategies to evaluate the quality of tone mapped images. Eilertsen et al. [2016] provide an overview on different ways how to measure the quality of TMOs:

- *Fidelity with reality*: Comparing the tone mapped image being viewed on a display device with the real physical scene. This kind of evaluation requires a lot of effort, as it involves displaying the tone mapped image and the corresponding physical scene in the same experimental setup. It is also not possible to use this evaluation metric for computer graphics rendered scenes.
- *Fidelity with HDR reproduction*: Comparing the tone mapped image with the reference HDR image being displayed on a high dynamic range display, which was proposed by Ledda et al. [2005]. Even though this approach is simpler to carry out, some form of tone mapping is introduced by the HDR display which causes imperfections in the displayed image. Another disadvantage is that similar to the *fidelity with reality* method, it depends on a subjective evaluation of the tone mapped image with a reference.
- *Non-reference methods*: Evaluation without any reference image being shown. The evaluation is performed with respect to individual preferences. Those have the advantage that they are very simple and sufficient for many applications, like artistic usage. Nevertheless, it is only a subjective evaluation method.
- *Appearance match*: Comparison of color appearance between the real scene and the tone mapped image with the help of magnitude estimation methods. This method measures the brightness of square patches in a physical scene and on a display. It also brings some disadvantages, as it is a very difficult method and does not guarantee the overall match of image appearance.

They point out, that none of the evaluation methods is free of problems and that the choice clearly depends on the application. In their study they only list methods which require a human interaction to evaluate the performance of a tone mapping operator. This can be a huge disadvantage for many applications as it requires a lot of work to perform the evaluation.

However, there is also an amount of computer algorithms, which are able to measure the quality of an image. These are considered as *objective image quality metrics* and were not designed in the context of high dynamic range imaging. Because of that, such algorithms are not originally intended to measure the quality of a tone mapped image. Instead, they evaluate the visual quality of an image, which could be affected during acquisition, processing, compression, storage or transmission. Objective image quality metrics can be classified as from Wang et al. [2004]:

- *Full-reference method*: Such an algorithm expects a manipulated image and compares it with the original distortion-free image. One of the most simple methods is the mean squared error.
- *No-reference method*: Also known as blind quality assessment method. It does not require any reference image for the evaluation.
- *Reduced-reference method*: Is only a part of the reference image available, for example, in the form of extracted features, then it is considered as reduced-reference quality assessment.

The success of two design principles of image quality assessment inspired Yeganeh & Wang [2012] to develop an objective model to assess the quality of tone mapped images. This is a full-reference method, which compares the tone mapped LDR image with the original high dynamic range image. It is based on a combination of a multi-scale structural fidelity measure and a statistical naturalness measure. Their experiments have shown that their *tone mapped quality index (TMQI)* is reasonably correlated with subjective evaluations of image quality.

2.2 Deep semantic segmentation

While approaches using deep learning models have been existing quite a time [e.g. Ivakhnenko, 1971], they have been used extensively only since the last years, since modern graphics processing units (GPUs) enabled the fast matrix and vector multiplications, which are required for the training of NNs [Schmidhuber, 2015]. Recently, deep convolutional neural networks (DCNNs) have achieved state of the art performance in high level computer vision tasks, like image classification and object detection [Chen et al., 2018a]. Additionally, the interest in image semantic segmentation increases more and more for computer vision and machine learning researchers [Garcia-Garcia et al., 2018]. Hence, this section covers the recent progress in deep learning based image semantic segmentation and shows the approaches for images of street scenes.

2.2.1 Scene understanding for autonomous driving

The visual understanding of street scenes plays an important role for a wide range of applications. The approaches for a scene understanding can be sorted by the amount of knowledge of a scene, similar as described by Li et al. [2009]:

- *Classification*: In the simplest case, one single label is assigned to an image, e.g. urban street scene or rural street scene, as done by Bosch et al. [2008] or Krizhevsky et al. [2012].
- *Recognition*: One step further, it is possible to gain a basic understanding of the scene content, by assigning a list of annotations without localization, e.g. [Li & Wang, 2003].
- *Object detection*: In this more advanced approach, specific objects are located in the image, typically described by a label and a tight-fitting bounding box around the object [e.g. Redmon et al., 2016; Chen et al., 2018b].
- *Semantic segmentation*: It is considered as one of the most challenging tasks in computer vision [Liu et al., 2015] and belongs to assigning a class label for every pixel in the image [e.g. Chen et al., 2018a; Long et al., 2015].
- *Semantic instance segmentation*: In this scene understanding approach it is aimed to not only assign a label for every pixel in the image, but also to identify object instances and therefore, to distinguish between every single object in the scene [e.g. Dai et al., 2016; Li et al., 2017].

Figure 2.6 shows an example for a semantic segmentation of an urban street scene. The left image shows an image as being recorded from a camera mounted on a driving car while the right image contains the ground truth, which is often referred to as *pixel level annotation* or *label map*. It is color encoded, such that a specific combination of color values correspond to a semantic

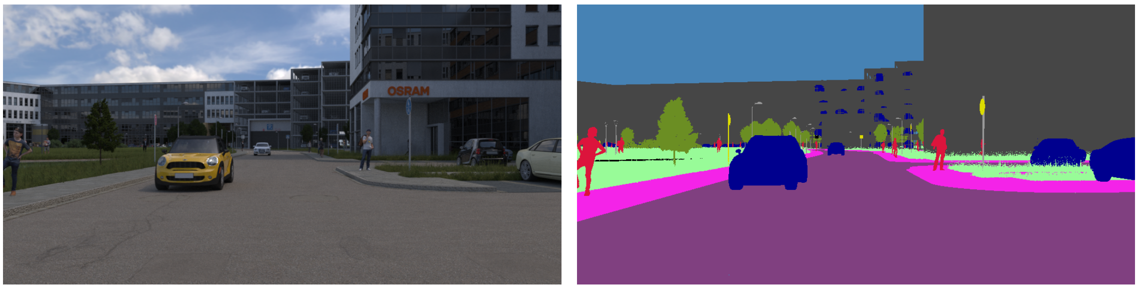


Figure 2.6 Example for the semantic segmentation of an urban street scene. The image was rendered using computer graphics and belongs to a highly realistic synthetic dataset from BIT [2019], which was developed for machine learning and validation.

class, for example car, road or pedestrian. Images like this, with their respective ground truth annotation, are usually used for supervised learning approaches, which is very often the initial situation for deep learning algorithms.

2.2.2 Semantic segmentation for road scene understanding

Garcia-Garcia et al. [2018] reviewed deep learning based approaches for semantic segmentation. They provide a whole characterization of that field and cover topics on common deep network architectures, training tips and details, datasets and challenges as well as methods for the semantic segmentation itself. Their survey shows, that this field is non-trivial and requires a lot of preliminary knowledge on machine learning concepts. This section will therefore give a very general introduction into this field.

Semantic segmentation datasets for driving scenes

In the following, popular datasets that provide RGB images of street scenes with pixel-wise labels will be described. Classifying them may be done for example according to the origin of their frames:

- Real-world datasets
  - *Cityscapes* [Cordts et al., 2016]: The database contains semantic annotations for 30 classes of urban street scenes from 50 different cities. The images were captured at daytime with good weather conditions. It consists of 20,000 coarse annotated images and around 5,000 fine annotated ones.
  - *CamVid* [Brostow et al., 2009]: It consists of 701 frames, in which each pixel is associated with one of 32 semantic classes. The frames originate from three video sequences that were captured in daylight with rather sunny weather conditions and one sequence that was recorded at dusk. The environment is mixed between residential and urban.
  - *KITTI* [Geiger et al., 2013]: The KITTI dataset provides images of street scenes from the German city Karlsruhe captured at sunny weather conditions. The semantic segmentation benchmark consists of 400 semantically annotated frames, which are split into half for training and testing. The data format and metrics are conform with the Cityscapes dataset.
  - *BDD100K* [Yu et al., 2018]: The Berkeley Deep Drive dataset contains 5,683 fine annotated video frames with semantic annotation for 40 different object classes. The images

were recorded mainly in the US and show great variety, as they originate from 100,000 different video sequences (120,000,000 images) captured in multiple cities, at different weather conditions and at various times of the day.

  - *Mapillary Vistas* [Neuhold et al., 2017]: The dataset consists of 25,000 street scene images, which are annotated into 66 classes. The images were recorded from all around the world with different imaging devices (e.g. mobile phones, action cameras, professional capturing rigs) and at various conditions regarding weather, season and daytime.

- Synthetic datasets
  - *BIT dataset* [BIT, 2019]: It consists of synthetically rendered images with a very high degree of realism. The dataset contains images of multiple scenes with different variations, like weather or daytime and is available with the respective pixel-wise ground truth annotations. It currently provides labels for 11 different classes, while the dataset is continuously extended. Additionally, the images from the camera sensors are available as high dynamic range images.
  - *SYNTHIA* [Ros et al., 2016]: The dataset provides annotations for 13 different classes and was rendered from synthetic American and European cities, as well as green areas and highway scenes. It covers different illumination conditions, multiple weather scenarios and provides images for every season. The dataset is divided into different sequences summing up to over 200,000 images in total.
  - *GTA-5* [Richter et al., 2016]: The dataset "Playing for data: Ground truth from computer games" is also known as GTA-5 dataset, named after the video game where the frames were captured from. It consists of 24,966 manually labeled synthetic street scene images divided into 19 different classes. The images show different weather conditions, daytimes and locations.
  - *Synscapes* [Wrenninge & Unger, 2018]: It is a photorealistic synthetic dataset containing 25,000 images. The images do not follow a driven path through a single virtual world, but were procedurally generated from an entirely unique scene. Synscapes was designed to be similar in structure and content to the Cityscapes dataset and includes all of the 19 training classes for semantic segmentation.
  - *Virtual KITTI* [Gaidon et al., 2016]: It is a synthetic dataset which consists of 21,260 frames [Naverlabs, 2019] generated from five different virtual worlds. The dataset covers multiple imaging and weather conditions in urban setting and provides ground truth annotations for 14 different semantic classes. Some of the synthetic sequences were cloned from the original real-world KITTI dataset, giving the dataset its name.

As this list could be extended [e.g. Huang et al., 2018; Pinggera et al., 2016], there are only few datasets providing high dynamic range images. The Cityscapes dataset, which provides 16 bit HDR images, is the only dataset to the best of the authors knowledge providing a higher dynamic range than usual datasets for automotive applications with corresponding fine-annotated semantic class labels for the whole scene. According to the specifications of the CMOS camera sensor (OnSemi AR0331), which was used to acquire the Cityscapes dataset, the sensor enables high dynamic range imaging via interlaced multi-exposure readout. This makes it possibly the only free available HDR dataset for a complex street scene understanding right now.

Deep network architectures

According to Garcia-Garcia et al. [2018], certain architectures of DNNs have become widely known standards in the field of semantic segmentation, which are currently used as building blocks for a lot of segmentation architectures.

- *AlexNet*, which was presented by Krizhevsky et al. [2012], consists of only five convolutional layers and three fully-connected ones and is considered as the pioneering DNN [Garcia-Garcia et al., 2018].
- *VGG* is a convolutional neural network (CNN) introduced by the Visual Geometry Group (VGG) [Simonyan & Zisserman, 2014]. The group proposed multiple models, but the one consisting of 16 weight layers became most popular and is known as VGG-16 [Garcia-Garcia et al., 2018].
- *GoogLeNet*, also known as *Inception Network*, is characterized by its complexity and a newly introduced block called inception module from Szegedy et al. [2015]. The network consists of 22 layers and got famous for instance through the "deep dream" experiments, where the architecture was used to help understanding and visualizing how learned features in a neural network look like [Mordvintsec et al., 2015].
- *ResNet*, introduced by He et al. [2016], is a very deep neural network which consists of up to 152 layers. The model can also be used with a reduced amount of depth, for example with 101 layers (ResNet-101). The introduction of residual blocks in the network improved learning capabilities and helped overcoming the vanishing gradients problem [Garcia-Garcia et al., 2018].

Training characteristics

While the following characteristics may also apply for other tasks, Garcia-Garcia et al. [2018] give some suggestions when training NNs for semantic segmentation:

- *Transfer learning*: In many cases training a network may not be feasible, for example, because of the non-availability of a dataset with sufficient size or as it could take too long for the experiments to reach convergence [Garcia-Garcia et al., 2018]. It has been shown, that deep neural networks tend to learn similar features in their first layers [Yosinski et al., 2014]. The idea behind transfer learning in deep learning is to use a pre-trained model instead of training it from scratch [Garcia-Garcia et al., 2018]. Especially, when the target dataset is significantly smaller than the base dataset, transfer learning can improve the generalization of a network, even when the target task is different [Yosinski et al., 2014]. One of the major transfer learning scenarios is to fine-tune the weights of a pre-trained network through continuing with the training [Garcia-Garcia et al., 2018].
- *Data augmentation*: It is a technique to improve the generalization capability of a model, where the most common approach is to apply a set of transformations to the existing data [Garcia-Garcia et al., 2018]. The goal is to artificially increase the dataset, which acts as a regularization and prevents the model from overfitting [Garcia-Garcia et al., 2018]. Typical

transformations are translation, rotation, warping, scaling, color space shifts or crops [Garcia-Garcia et al., 2018]. Latest, Zhu et al. [2019] proposed a video prediction-based data synthesis method to improve the accuracy of semantic segmentation models as a kind of data augmentation.

Methods

The key idea behind deep learning based approaches for semantic segmentation is to automatically learn appropriate features with CNNs instead of hand-crafting them, which requires domain expertise and a lot of effort [Garcia-Garcia et al., 2018]. There are different deep-learning techniques, which are utilized to perform the semantic segmentation.

Considered by Garcia-Garcia et al. [2018] as the cornerstone of the most successful deep-learning techniques applied to semantic segmentation is the *fully convolutional network (FCN)* from Long et al. [2015]. The approach benefits from the ability of existing CNNs learning a rich hierarchy of features [Garcia-Garcia et al., 2018]. Long et al. [2015] replaced the fully connected layers in well-known classification models – AlexNet, VGG, GoogLeNet and ResNet – with convolutional ones, which gives spatial maps as an output, instead of classification scores. Those maps can be used to produce dense per-pixel labeled outputs using deconvolutional layers. Though it is considered as a powerful and flexible model, it still brings certain problems for different situations, for example, when it should take useful global context information into account [Garcia-Garcia et al., 2018].

Recently, there are various state of the art models that overcome this limitation:

- *U-net* [Ronneberger et al., 2015], for example, is an architecture that modifies the FCN network to capture context. It supplements a contracting network, where upsampling operators are used to replace pooling operators which enables the context integration. A symmetric expanding network is used to enable precise localization.
- *RefineNet* [Lin et al., 2016] exploits different levels of detail at various stages of convolutions. It fuses the features from different resolutions to reduce the computational effort but maintain the high-resolution of input images.
- *Structural-RNN* [Jain et al., 2016] is an approach which uses deep *recurrent neural networks (RNNs)* to model temporal interactions. The author's have shown that their method improves diverse spatio-temporal problems including human motion modeling, human-object interaction and driver maneuver anticipation.
- The *DeepLab*-models from Chen et al. [2018a] integrate information from different spatial scales. The model uses *dilated convolutions*, also known as *atrous* convolutions, which are like regular convolutions but make use of upsampled filters. The upsampling factor can thereby be controlled by a parameter called dilation rate.
- The *dense relation network (DRN)* from Zhuang et al. [2018] aggregates multi-scale features to obtain hierarchical contextual information. It provides a context-restricted loss (CRL) to constrict the consistency of contextual representations assigned between images and labels.

Accuracy evaluation metrics

In the following the most popular metrics [Garcia-Garcia et al., 2018] to assess the accuracy of a semantic segmentation will be reported. The notation and formula are consistent with the ones from Garcia-Garcia et al. [2018]. Therefore, p_{ii}, also known as true positives, is the total number of correctly predicted pixels for class i. The first index represents the index for the true semantic class and the second index for the predicted class. The number of false positives is p_{ij} and the number of false negatives is p_{ji}. Assuming a total of $k+1$ classes, the accuracy evaluation metrics are determined as follows:

- *Pixel Accuracy (PA)* is the ratio between the amount of correctly classified pixels and the total amount of them and therefore the simplest metric.

$$PA = \frac{\sum_{i=0}^{k} p_{ii}}{\sum_{i=0}^{k} \sum_{j=0}^{k} p_{ij}} \tag{2.1}$$

- *Mean Pixel Accuracy (MPA)* is a slightly improved version of the pixel accuracy in which the ratio of properly classified pixels is balanced by the number of pixels of each class.

$$MPA = \frac{1}{k+1} \sum_{i=0}^{k} \frac{p_{ii}}{\sum_{j=0}^{k} p_{ij}} \tag{2.2}$$

- *Mean intersection over union (mIoU)* is the standard metric for assessing the accuracy of a semantic segmentation and is computed as a ratio between the intersection and the union of the ground truth and the predicted classes in the image.

$$mIoU = \frac{1}{k+1} \sum_{i=0}^{k} \frac{p_{ii}}{\sum_{j=0}^{k} p_{ij} + \sum_{j=0}^{k} p_{ji} - p_{ii}} \tag{2.3}$$

- *Frequency Weighted Intersection over Union (FWIoU)* is a modified version of the mIoU and weights each class importance according to the appearance frequency.

$$FWIoU = \frac{1}{\sum_{i=0}^{k} \sum_{j=0}^{k} p_{ij}} \sum_{i=0}^{k} \frac{\sum_{j=0}^{k} p_{ij} p_{ii}}{\sum_{j=0}^{k} p_{ij} + \sum_{j=0}^{k} p_{ji} - p_{ii}} \tag{2.4}$$

Deep semantic segmentation with HDR images

As described in subsection 2.2.2 the 16 bit Cityscapes dataset is currently the only HDR dataset for a semantic (road scene) segmentation to the best of the author's knowledge. The Cityscapes Benchmark Suite [2019] offers an evaluation server where researchers can measure the performance of their semantic segmentation method. It lists the performance of every published model and the datasets, which were used for training. Out of all published methods there appear only 2 out of 172 methods (as of July 2019) that use the HDR images for training in addition to the LDR dataset.

One of the methods performs poorly and uses a classical machine learning approach with a random forest framework [Kang & Nguyen, 2019]. The other one is currently one of the best performing methods on the benchmark suite and uses a deep learning approach [Li et al., 2019]. Nevertheless, the methods only use the HDR images for training without giving any details about the HDR dataset in their paper. There are also no other methods known by the author, which use high dynamic range images for the task of semantic segmentation. Therefore, this work may be the first to discover the influence of HDR images for a semantic segmentation using neural networks.

2.3 Deep domain adaptation for semantic segmentation

Domain adaptation is a special case of transfer learning, where labeled data is available in only one domain, which is usually referred to as the *source* domain [Csurka, 2017]. This data can be used to train a prediction model for unlabeled data in the so called *target* domain, where the task is usually the same. In this scenario it is assumed that both data distributions are independent and identically distributed, which makes it a classical machine learning problem [Csurka, 2017]. Therefore, a clear drop in performance can be obtained, in case the distributions do not align. This can be seen in figure 2.7, where images from the source domain (top) are used to train a semantic segmentation network, which predicts poorly (bottom left) on images from the target domain.

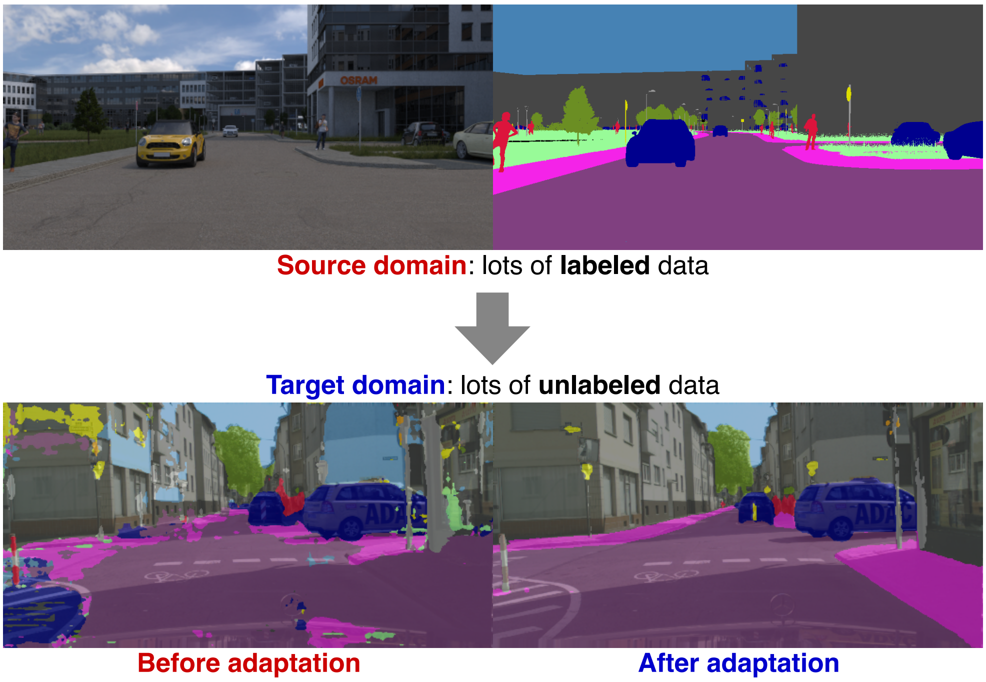

Source domain: lots of labeled data
Target domain: lots of unlabeled data
Before adaptation
After adaptation

Figure 2.7 Top shows an image from the source domain dataset (left), which is available with its corresponding ground truth (right). After training a neural network on those images to predict pixel-wise semantic classes, on the bottom can be exemplary seen the output of a target image before (left) and after adaptation (right). Figure reproduced from Hoffman et al. [2016].

2.3.1 Principle of domain adaptation

In general, the strong performance of deep neural networks on tasks such as semantic segmentation can be attributed to the availability of abundant labeled training data [Sankaranarayanan et al., 2017]. As there is often a lack of carefully annotated pixel-wise labels, the utility of synthetically generated images for the training is an approach to overcome this problem. Also the assurance against corner cases is an important application, where synthetic images could be very helpful, in case the generalization between the domains could be certified. Therefore, in the context of semantic segmentation the domain adaptation is often an attempt to improve the poor generalization capability of deep semantic segmentation networks when trained on synthetic images and tested on real ones (see figure 2.7, bottom right).

There exist some literature reviews on the domain adaptation problem to some extent. In chronological order, Patel et al. [2015] state that domain adaptation is a fundamental learning problem and that it has gained a lot of attention in scenarios like natural language processing, statistics, machine learning and recently, also in computer vision. For that, they provide an overview of the visual domain adaptation. Later, Csurka [2017] provides a comprehensive survey on the domain adaptation for more complex visual applications, like object detection. Nevertheless, they focus on non-deep learning based methods and only mention the task of semantic segmentation. Latest, Wang & Deng [2018] summarize deep learning based domain adaptation approaches in their work and include recent progress about the application of semantic segmentation in their survey. Still, their review on methods for the task of semantic segmentation is rather short. These few available literature reviews indicate that this type of field is rather new.

2.3.2 Domain adaptation for semantic segmentation with neural networks

In the field of domain adaptation for semantic segmentation using deep neural networks there can be seen some trends. Firstly, there are many methods available which tackle the problem of domain adaptation for autonomous driving [e.g. Hoffman et al., 2016; Chen et al., 2017; Zhang et al., 2017; Sankaranarayanan et al., 2017; Chen et al., 2018c; Hoffman et al., 2018; Hong et al., 2018; Tsai et al., 2018; Luo et al., 2019a], as they are using training data from road scenes to transfer the learned knowledge. In this case, many methods (see table 2.1) use the real-world Cityscapes dataset [Cordts et al., 2016] for the target domain and the synthetic SYNTHIA dataset [Ros et al., 2016] or the GTA-5 dataset [Richter et al., 2016] for the source domain. Since the Cityscapes dataset comes with labeled ground-truth annotations, the performance of the semantic segmentation after the domain adaptation is obtained via the class-wise mIoU or the pixel accuracy in most cases.

One problem that still remains from state of the art methods is that the final performance of the neural network trained on synthetic images is still much lower after the domain adaptation than if trained directly on real-world images. In comparison, the highest mIoU class score of all published semantic segmentation networks on the Cityscapes Benchmark Suite [2019] is 83.6% (as of July 2019), while the network is trained on only real-world images according to the method overview on the Benchmark Suite. But when using only synthetic data for the training of a segmentation network, the highest performances vary between 18 - 39 % before and 35 - 46 % after the domain adaptation for state of the art methods [e.g. Hoffman et al., 2018; Luo et al., 2019a; Gong et al., 2019].

The following section summarizes some of the most recent methods for the semantic segmentation with domain adaptation in the field of autonomous driving.

2.3.3 Domain adaptation in the field of autonomous driving

The first domain adaptive semantic segmentation method was proposed by Hoffman et al. [2016] and performs three different kind of experiments. First, they apply a domain adaptation from synthetic to real-world, using the SYNTHIA and the GTA-5 as the source and the Cityscapes dataset as the target domain. Second, they perform a cross seasons adaptation by using the SYNTHIA dataset only, where synthetic images are available for all four seasons. Third, they carry out a cross city adaptation by using on the one hand, the Cityscapes dataset only and on the other hand, another real world dataset additionally, which contains thousands of dense annotated dash-cam video frames. The overview is visualized in figure 2.8. Since then, many authors followed their experiments to improve the results on these tasks.

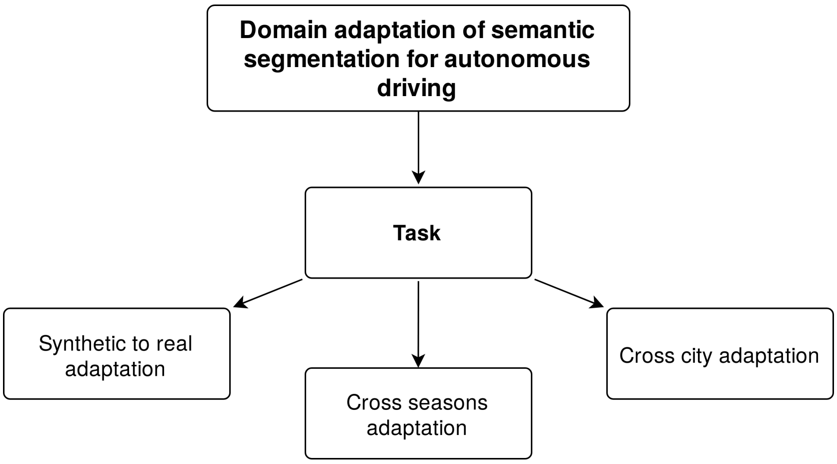
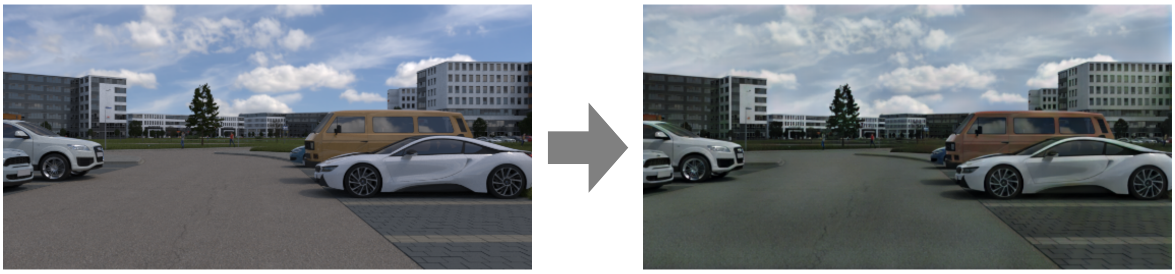

Domain adaptation of semantic segmentation for autonomous driving
Task
Synthetic to real adaptation
Cross seasons adaptation
Cross city adaptation

Figure 2.8 Conceptual scheme of the state of the art approaches for the domain adaptation for semantic segmentation separated by their intent to solve different tasks.

Table 2.1 provides an overview of state of the art methods for domain adaptation for a semantic segmentation in the field of autonomous driving. All of the methods have in common, that they are able to predict semantic label images for street scene images and their aim is to improve the segmentation performance on target domain images through the domain adaptation. Some of the methods are also able to generate domain adapted street scene images. This task is recently known as *neural style transfer* or *image-to-image translation* [Gatys et al., 2016; Isola et al., 2017]. Such a domain adapted image can be seen in figure 2.9, where the appearance of the original source domain image (left) is adapted (right) to the appearance of images from the target domain. In addition to generative and non-generative approaches, there are further differences between state of the art methods. Not all methods aim to solve the same tasks. While all of the methods perform a synthetic to real-world adaptation, either a) from SYNTHIA to Cityscapes, b) from GTA-5 to Cityscapes or e) from Virtual KITTI to KITTI, only some of the methods perform



Original **source domain** image

Target **domain adapted** image

Figure 2.9 Several domain adaptation approaches use image-to-image translation techniques to generate target domain adapted images (right). The difference to the original source domain image (left) can be seen, for example, in the appearance of the road surface or the color of the sky.

a cross city adaptation (c) or a cross seasons adaptation (d). There are also some similarities, for example, that most methods rely on existing segmentation networks (e.g. FCN-8 network or DeepLab-v2 network) with pre-trained classification networks (e.g. VGG-16 or ResNet-101) as a backbone. Those domain adaptation methods extend existing frameworks in order to improve the segmentation performance on target domain images using labeled source domain images and unlabeled target domain images during training.

Method	Task	Base model (backbone)	Generative
FCNs in the Wild [Hoffman et al., 2016]	a,b,c,d	· FCN-8s (VGG-16)	✗
Cross city adaptation [Chen et al., 2017]	a,c	· Front-end dilitated-FCN	✗
Curriculum domain adaptation [Zhang et al., 2017]	a	· FCN-8s (VGG-19)	✗
Unsupervised GAN adaptation [Sankaranarayanan et al., 2017]	a,b	· FCN-8s (VGG-16)	✓
ROAD [Chen et al., 2018c]	a,b	· DeepLab-v2 (VGG-16)	✗
CyCADA [Hoffman et al., 2018]	b,d	· FCN-8s (VGG-16) · DRN-26	✓
cGAN for structured domain adaptation [Hong et al., 2018]	a,b	· FCN-8s (VGG-16)	✗
Structured output space adaptation [Tsai et al., 2018]	a,b,c	· DeepLab-v2 (VGG-16) · DeepLab-v2 (ResNet-101)	✗
DCAN [Wu et al., 2018]	a,b	· FCN-8s (ResNet-101)	✓
Conservative loss adaptation [Zhu et al., 2018]	a,b	· FCN-8s (VGG-16)	✗
CBST [Zou et al., 2018]	a,b,c	· CBST (FCN-8s–VGG16) · CBST (ResNet-38)	✗

CLAN [Luo et al., 2019b]	a,b	· FCN-8s (VGG-16) · DeepLab-v2 (ResNet-101)	✗
SPIGAN [Lee et al., 2019]	a	· FCN-8s (VGG-19)	✓
ADVENT [Vu et al., 2019a]	a,b	· DeepLab-v2 (VGG-16) · DeepLab-v2 (ResNet-101)	✗
DLOW [Gong et al., 2019]	a	· DeepLab-v2 (ResNet-101)	✓
GIO-Ada [Chen et al., 2019]	a,e	· DeepLab-v2 (VGG-16)	✓
DISE [Chang et al., 2019]	a,b	· DeepLab-v2 (ResNet-101)	✓
DADA [Vu et al., 2019b]	a	· DeepLab-v2 (VGG-16) · DeepLab-v2 (ResNet-101)	✗
SIBAN [Luo et al., 2019a]	a,b	· DeepLab-v2 (VGG-16) · DeepLab-v2 (ResNet-101)	✗

Table 2.1 Overview of recent methods which perform a domain adaptation in order to improve the semantic segmentation performance of neural networks trained on a source domain and evaluated on a target domain in the field of autonomous driving. The domain adaptation is performed from a) SYNTHIA to Cityscapes, b) GTA-5 to Cityscapes, c) city to city (using different real world datasets), d) SYNTHIA season to season and e) Virtual KITTI to KITTI [Geiger et al., 2013]. Generative methods are able to perform an image-to-image translation from source to target domain.

2.3.4 Deep domain adaptation of HDR images

All of the methods in table 2.1 perform a domain adaptation between LDR images. There are also approaches, which generate HDR images from LDR images with [e.g. Endo et al., 2017] or without neural networks [e.g. Rempel et al., 2007]. This process is called inverse or reverse tone mapping. These approaches aim to recover the clipped brightness values of an original scene by estimating an HDR image. This might be also considered as a kind of domain adaptation, where the source domain is defined by the LDR images and the target domain by HDR images. Nevertheless, in this scenario the two domains consist of only images from the exact same scenes, whereas the domain adaptation for a semantic segmentation is performed between different scenes. Hence, to the best of the author's knowledge, there are no domain adaptation methods which are performed between HDR images from two different domains.

3 Methodology

In this thesis the influence of synthetic high dynamic range images for several tasks in the field of autonomous driving is investigated. Therefore, in this chapter the methods which are developed to produce the experimental results are explained. The contributions of this thesis are three-folded:

- *LDR dataset generation and evaluation*: One goal is to find out if the high dynamic range of HDR images can improve the semantic segmentation performance. One possibility is to directly train on the HDR images as shown by the upper path in figure 3.1. The first approach of this thesis is to indirectly make use of the high dynamic range. This is done by generating a set of low dynamic range images from different tone mapping operators of the HDR images and using them for training as shown by the lower paths in figure 3.1. Additionally, a ranking method to evaluate the tone mapped datasets is presented. The approach is described in section 3.1.

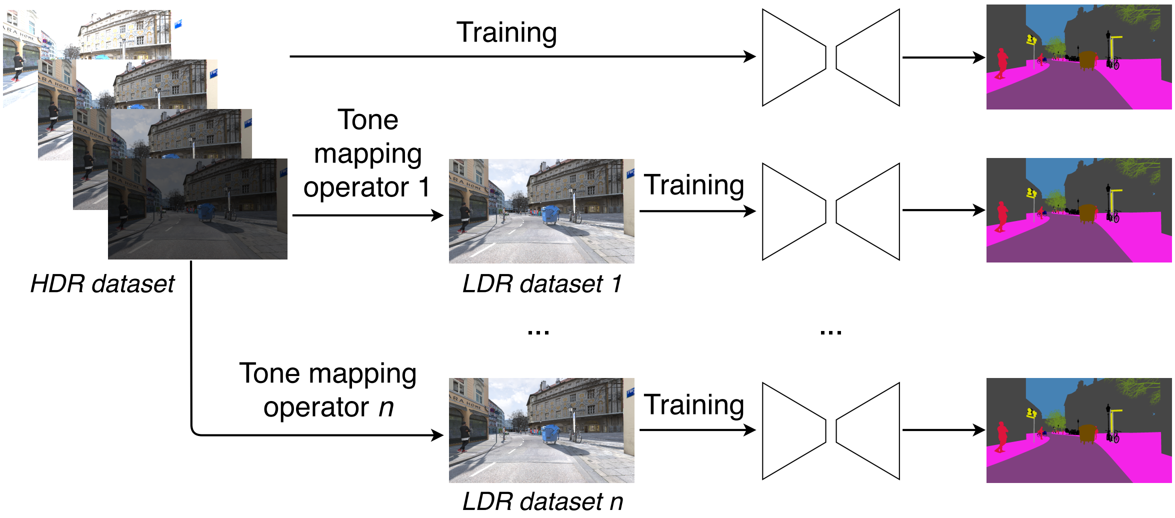


Figure 3.1 Overview of the semantic segmentation networks. For every dataset one network is trained, which is evaluated afterwards to determine the semantic segmentation performance.

- *Evaluation of semantic segmentation performance*: The second approach allows to evaluate the different semantic segmentation networks, which are trained on the HDR dataset and the different LDR datasets as shown in figure 3.1. The method is described in section 3.2.
- *Domain adaptation of synthetic HDR images*: The third approach proposes to improve the semantic segmentation performance of the networks on real-world images, as all networks in figure 3.1 are trained solely on synthetic images. The approach is described in section 3.3.

3.1 Tone mapping of high dynamic range images

In this thesis it is explored whether high dynamic range images can be used to improve the semantic segmentation performance of deep neural networks. The first approach aims to create several low dynamic range image sets out of the investigated HDR dataset with different tone mapping operators. This may allow to compress the most important brightness values of an HDR image for a semantic segmentation into a low dynamic range image. Therefore, it is also explored which tone mapping operators generate better LDR images for a semantic segmentation. Figure 3.1 provides an overview of the networks, which are trained on the HDR and LDR images. There will be *n* different tone mapping operators used to create *n* different LDR datasets to train *n* different semantic segmentation models. Furthermore, it is investigated, which of the tone mapping operators achieve the highest performances at the semantic segmentation. How to evaluate the performance of the models will be explained in the next section.

For the first approach, the process of tone mapping is divided into several steps. The selection of tone mapping operators is described in subsection 3.1.1. A method to determine the parameters for each TMO is described in subsection 3.1.2. In order to approximate the semantic segmentation performance for the different tone mapping operators a ranking method is proposed in subsection 3.1.3. The approximation should help to decide which operators to select or exclude for the training when the amount of GPUs is limited for experiments. Addressing dataset creation it is proposed to increase the dynamic range of the LDR training sets by generating a mixed LDR dataset from different tone mapping operators. The approach is described in subsection 3.1.4.

3.1.1 Selection of tone mapping operators

To the best of the author's knowledge, there are no studies evaluating tone mapping operators for a semantic segmentation. The approach, as shown in figure 3.1, which requires to train a semantic segmentation network until convergence, is very time consuming and requires expensive GPUs. To tackle this problem a method is developed, which is computationally less expensive and provides an expectation for the semantic segmentation performance of different tone mapping operators. The method is described in subsection 3.1.3.

The selection of tone mapping operators is accomplished from the results of the study performed by Čadík et al. [2006]. The authors compared different tone mapping operators with respect to various image quality criteria, such as levels of detail and provided a ranking between the operators according to their overall image quality definition.
The author of this thesis assumes, that images with a high overall quality are better suited for a semantic segmentation with neural networks than images with a low overall quality. For humans it is usually also easier to distinguish between objects in images with a higher quality. Therefore, the seven best performing operators from Čadík et al. [2006] are chosen for this study. Additionally, another operator was selected, which was published by the same author, as the best performing tone mapping operator.

3.1.2 Parameter determination of tone mapping operators

In this thesis, eight different tone mapping operators are used, depending on in total 12 parameters. Handcrafting the parameters for each operator requires a very high effort, as it needs to test different parameter settings and manually evaluating the generated LDR images. To overcome this high effort the process of parameter determination is automatized.
In literature, there is usually no explanation of how to choose the right parameters for a tone mapping operator with respect to a specific objective. For example, Čadík et al. [2006] relied only on a short fine-tuning of the parameters for their investigated TMOs rather than performing a profound investigation. In many publications of different tone mapping operators the authors provide a default parameter setting, which could be used to generate LDR images [e.g. Drago et al., 2003; Schlick, 1995]. But they provide no statement on how well the parameter values adapt for different images. When Čadík et al. [2006] evaluated various tone mapping operators, they used 14 different HDR images for the evaluation. But there is no study, to the best of the author's knowledge, where a complete dataset of thousands of images is tone mapped and where it is described how the operators behave on so many different images.

The most intuitive way to automatically find the best parameters for each tone mapping operator for a semantic segmentation is to train a segmentation network for multiple parameter settings and use the best performing one. Unfortunately, this is not feasible due to limited processing units. Subsection 2.1.4 provides different methods to evaluate the generated images from a tone mapping operator. The tone mapped quality index (TMQI) compares a tone mapped LDR image with its originating HDR image and provides a value for the quality of the LDR image. Finding the parameters of the TMOs which produce the LDR images with the highest TMQI-value can be solved using a constrained nonlinear optimization algorithm.

3.1.3 Ranking method for tone mapped datasets

Addressed in subsection 3.1.1, that there is no method to evaluate tone mapped datasets, an approach to rank the results from different TMOs with respect to the expected semantic segmentation performance is proposed. The aspects of the evaluation contain the following criteria:

- *Overall image quality (OIQ)*: One criterion for the evaluation of the results from the tone mapping operators is the overall image quality. In this measure, an objective evaluation of image quality aspects is performed.
- *Image dynamics (ID)*: Another criterion which should be fulfilled by the tone mapping operators is that the generated images should show an appropriate dynamic. To determine this measure the histogram of the images is considered.
- *Tone mapping sensitivity (TMS)*: The last criterion should discover, how sensitive each tone mapping operator is with respect to different images. Since the operators are applied with a single parameter setting on multiple images, this measure should reveal how well the operators generalize.

Figure 3.2 visualizes the approach of the ranking method. For each tone mapped dataset one score for the overall image quality, one for the image dynamics and one for the tone mapping sensitivity is computed. Finally, the three scores are relatively weighted to achieve one final ranking score for each tone mapping operator.

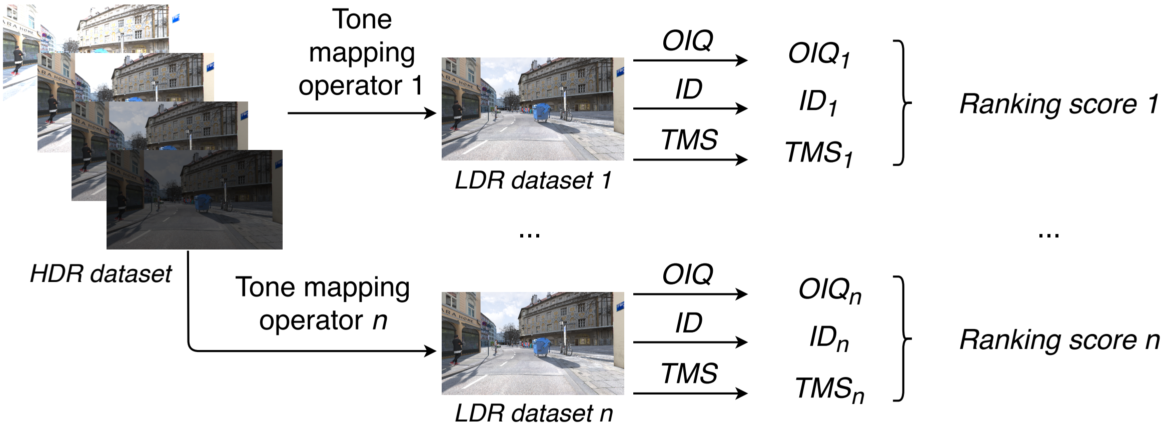


Figure 3.2 The high dynamic range dataset is tone mapped with *n* different operators resulting in *n* different LDR datasets. Every operator gets a ranking score, which consists of a relative weighting between an *overall image quality (OIQ)*, *image dynamics (ID)* and *tone mapping sensitivity (TMS)* score determined on the tone mapped LDR datasets.

Overall image quality

There are many approaches to determine the overall image quality of images in the computer vision community. Image quality assessment (IQA) algorithms take an image as an input and calculate a quality score as an output. For this work, one full-reference and one no-reference method is used, to achieve a more robust estimation of the objective image quality than just using a single method. Figure 3.3 shows, that every tone mapped image is evaluated independently by the full-reference TMQI and the no-reference BRISQUE method for the ranking in this work. To avoid confusion, the term *value* is used as a measure for a single image and the term *score* is used to measure the performance of a tone mapping operator, determined over many values.

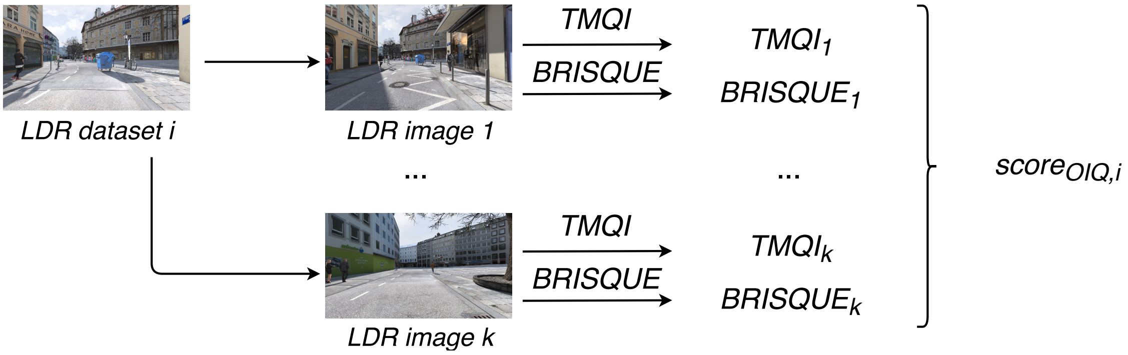


Figure 3.3 Overview of the overall image quality assessment using the tone mapped quality index (TMQI) method from Yeganeh & Wang [2012] and the blind/referenceless image spatial quality evaluator (BRISQUE) method from Mittal et al. [2011].

The tone mapped quality index (TMQI) from Yeganeh & Wang [2012] returns a value for every tone mapped LDR image *j* from the HDR dataset consisting of *k* images. The value provides a number between 0 and 1, where a higher value means better overall image quality. Averaging the TMQI values from every tone mapped image gives an approximation for the image quality by the TMQI-method for a tone mapping operator *i*

$$\mu_{TMQI,i} = \frac{1}{k} \cdot \sum_{j=1}^{k} TMQI_j \qquad with\ i = 1, ..., n \tag{3.1}$$

which is normalized by

$$score_{TMQI,i} = \frac{\mu_{TMQI,i} - min(\mu_{TMQI})}{max(\mu_{TMQI}) - min(\mu_{TMQI})} \qquad with\ i = 1, ..., n \tag{3.2}$$

The normalization corresponds to a linear interpolation between the lowest and highest TMQI mean values, in a way, that the best performing tone mapping operator gets a score of 1 and the worst one a score of 0. The $score_{TMQI,i}$ represents the score for a tone mapping operator i determined by the TMQI-method over the whole tone mapped dataset. In contrast, the blind/referenceless image spatial quality evaluator (BRISQUE) method from Mittal et al. [2011] returns a value larger than 0, whereas a smaller value means higher objective image quality. Similarly, the BRISQUE values are averaged over the tone mapped dataset by

$$\mu_{BRISQUE,i} = \frac{1}{k} \cdot \sum_{j=1}^{k} BRISQUE_j \qquad with\ i = 1, ..., n \tag{3.3}$$

and normalized by

$$score_{BRISQUE,i} = \frac{max(\mu_{BRISQUE}) - \mu_{BRISQUE,i}}{max(\mu_{BRISQUE}) - min(\mu_{BRISQUE})} \qquad with\ i = 1, ..., n \tag{3.4}$$

such that again the best performing operator gets a score of 1 and the worst a score of 0. The final score for the overall image quality can be obtained by

$$score_{OIQ,i} = \frac{1}{2} \cdot score_{TMQI,i} + \frac{1}{2} \cdot score_{BRISQUE,i} \qquad with\ i = 1, ..., n \tag{3.5}$$

which is the arithmetic mean of the TMQI and BRISQUE scores of each operator.

Image dynamics

The author assumes that the tone mapped images should all have a rather similar dynamic, which means that the distribution of grey values should follow specific rules in order to perform well on a semantic segmentation:

- *Equally balanced exposure*: It is desirable to achieve an illumination in the image, such that there are no over- or underexposed areas. Otherwise the classification of pixels in very dark or very bright areas might lead to incorrect class assignments.
- *Equally/normally distributed grey values*: If there are too many pixels with similar grey values in the image, it might be difficult to distinguish different classes between them for a semantic segmentation network. Therefore, it may be desirable to have a rather uniform or normal distribution of grey values in the image instead of having prominent peaks for specific grey values.

The distribution of grey values is given by the histogram of an image, which can be used to extract descriptive statistics such as mean, variance, skewness or kurtosis to define the aforementioned properties. To merge the information from the three color channels, the luminance image *L* may be computed from the ITU-R recommendation BT.709 [ITU, 2015] as follows

$$L = 0.2126 \cdot R + 0.7152 \cdot G + 0.0722 \cdot B \tag{3.6}$$

To compute scores for the image dynamics, the skew and kurtosis may seem very suitable. The skew of the luminance histogram can be used to evaluate the exposure in the image. Therefore, a skew of zero corresponds to an equal balance. The kurtosis is a measure for the peakedness of a probability distribution. It can be used to evaluate the distribution of grey values. A large kurtosis means that there are strong peaks in the probability distribution of the histogram, which might be unfavourable. A uniform distribution has a kurtosis of 1.8, which might be a suitable objective for this evaluation. Therefore, the skew and kurtosis are computed for every tone mapped image in each LDR dataset and scores for the skew and kurtosis are determined in order to get a number for the image dynamics for every tone mapping operator.

Similar as before, the skew values are determined for every image j in the LDR dataset from a tone mapping operator i. Then the values are averaged through

$$\mu_{skew,i} = \frac{1}{k} \cdot \sum_{j=1}^{k} skew_j \qquad with\ i = 1, ..., n \tag{3.7}$$

and normalized by

$$score_{skew,i} = \frac{max(|\mu_{skew}|) - |\mu_{skew,i}|}{max(|\mu_{skew}|) - min(|\mu_{skew}|)} \qquad with\ i = 1, ..., n \tag{3.8}$$

such that the highest reachable score is 1 and the lowest one is 0. Similarly, the score for the kurtosis is computed by averaging the kurtosis values by

$$\mu_{kurtosis,i} = \frac{1}{k} \cdot \sum_{j=1}^{k} kurtosis_j \qquad with\ i = 1, ..., n \tag{3.9}$$

and normalizing them by

$$score_{kurtosis,i} = \frac{max(|\mu_{kurtosis} - 1.8|) - |\mu_{kurtosis,i} - 1.8|}{max(|\mu_{kurtosis} - 1.8|) - min(|\mu_{kurtosis} - 1.8|)} \qquad with\ i = 1, ..., n \tag{3.10}$$

such that the highest score is reached when being close to an average kurtosis of 1.8, which corresponds to a uniform distribution of grey values in the tone mapped images. The final score for the image dynamics is computed by

$$score_{ID,i} = \frac{1}{2} \cdot score_{skew,i} + \frac{1}{2} \cdot score_{kurtosis,i} \qquad with\ i = 1, ..., n \tag{3.11}$$

where the weighting is again equally between the two scores.

Tone mapping sensitivity

The last measure for the evaluation of the tone mapping operators is the sensitivity of the operators with respect to different scenes. This measure intends to show how well each tone mapping operator can be applied to multiple images. This can be defined by the scattering of the objective image quality values of every tone mapped image for each tone mapping operator. Therefore, the determination of the sensitivity is done by considering the standard deviation of the calculated TMQI and BRISQUE values for the tone mapped images of every TMO.

The standard deviation of the TMQI values is calculated by

$$\sigma_{TMQI,i} = \sqrt{\frac{1}{k-1} \cdot \sum_{j=1}^{k} (TMQI_{j,i} - \mu_{TMQI,i})^2} \quad with\; i = 1, ..., n \tag{3.12}$$

where μ_i is the mean of the TMQI values of a tone mapping operator i. Similar to the other evaluations, a linear interpolation between highest and lowest values is performed to calculate scores for the tone mapping sensitivity through

$$score_{\sigma_{TMQI,i}} = \frac{max(\sigma_{TMQI}) - \sigma_{TMQI,i}}{max(\sigma_{TMQI}) - min(\sigma_{TMQI})} \quad with\; i = 1, ..., n \tag{3.13}$$

for the standard deviation from the TMQI evaluation method. The standard deviation of the BRISQUE values can be calculated similarly by replacing the TMQI values with the BRISQUE values of the tone mapped images

$$\sigma_{BRISQUE,i} = \sqrt{\frac{1}{k-1} \cdot \sum_{j=1}^{k} (BRISQUE_{j,i} - \mu_{BRISQUE,i})^2} \quad with\; i = 1, ..., n \tag{3.14}$$

These values are again normalized

$$score_{\sigma_{BRISQUE,i}} = \frac{max(\sigma_{BRISQUE}) - \sigma_{BRISQUE,i}}{max(\sigma_{BRISQUE}) - min(\sigma_{BRISQUE})} \quad with\; i = 1, ..., n \tag{3.15}$$

to obtain scores between 0 and 1. The final score for the tone mapping sensitivity evaluation is obtained by averaging the two scores as follows

$$score_{TMS,i} = \frac{1}{2} \cdot score_{\sigma_{TMQI,i}} + \frac{1}{2} \cdot score_{\sigma_{BRISQUE,i}} \quad with\; i = 1, ..., n \tag{3.16}$$

Final ranking scores

The final score for each tone mapping operator is calculated by relatively weighting the scores from the three different evaluation metrics – namely the *overall image quality (OIQ)*, the *image dynamics (ID)* and the *tone mapping sensitivity (TMS)* – as follows

$$score_{final,i} = \frac{5}{10} \cdot score_{OIQ,i} + \frac{3}{10} \cdot score_{ID,i} + \frac{2}{10} \cdot score_{TMS,i} \quad with\; i = 1, ..., n \tag{3.17}$$

where the weighting scores are chosen manually with respect to subjective importance. As the overall image quality is considered to be the most influencing factor for the performance at a semantic segmentation the weight is chosen 50% of the overall weights. As the overall image quality and the tone mapping sensitivity are both determined by the TMQI and BRISQUE methods, the weight for the tone mapping sensitivity score is chosen lowest in order to avoid too large redundancies within the ranking method.

3.1.4 Mixed LDR dataset for HDR approximation

In this thesis it is approached to improve the semantic segmentation performance of neural networks from HDR images by training a network on a mixed LDR dataset. The idea behind this approach is that neural networks might learn more useful features from images of multiple tone mapping operators which might lead to an increased semantic segmentation performance when training a network on such a dataset. Therefore, the brightness variations in the LDR images which are caused by different TMOs might be an approximation of the original high dynamic range images. For this purpose, the LDR images from multiple tone mapping operators are combined into one single dataset. The dataset is constructed by equally large fractions of every tone mapped dataset, such that the resulting mixed LDR dataset has the same number of images as all other datasets. The frames are consecutively selected from different tone mapping operators. The first image in the new mixed dataset is from TMO 1, the second frame from TMO 2 and so on until the sequence repeats. For datasets with consecutive frames, two following frames are highly correlated, since the content of those images hardly changes. Therefore, selecting these frames from different TMOs might reduce the redundancy and improve the semantic segmentation performance when training a network.

3.2 Evaluation of semantic segmentation performance

In this section, the second approach of this thesis is explained, which allows to determine the semantic segmentation performance of networks trained on HDR images and LDR images from different tone mapping operators. The evaluation approach is divided into three subtasks, which can be interpreted as different scenarios for automotive applications:

- *Closed system performance*: For the first task only images of the synthetic source domain are considered. The evaluation determines the semantic segmentation performance under optimal settings. This means that the training and the test set originate from the same domain and the same post processing operation. This approach intends to show the highest possible performance.
- *LDR generalization capability*: For the second task only images from the target domain are considered. Therefore, this task determines the semantic segmentation performance for real-world scenes. The network is thereby tested on low dynamic range images. This approach aims to show how well the networks can be used for automotive applications with LDR cameras.

- *HDR generalization capability*: The third task shows how well the semantic segmentation networks generalize on real-world scenes that are captured from an HDR sensor.

Figure 3.4 shows an overview of the approach. In this work there is trained one neural network, indicated by two trapezes, for every image dataset. This results in one segmentation network for the HDR training set and *n* networks for the tone mapped LDR training sets, with *k = 1,...,n*. To determine the semantic segmentation performance of the networks there are carried out three different evaluations. The three presented evaluation methods are intended to be rather independent measures to determine the semantic segmentation performance of the different networks. Any test case might be an imaginable scenario for automotive applications like an autonomous-driving system.

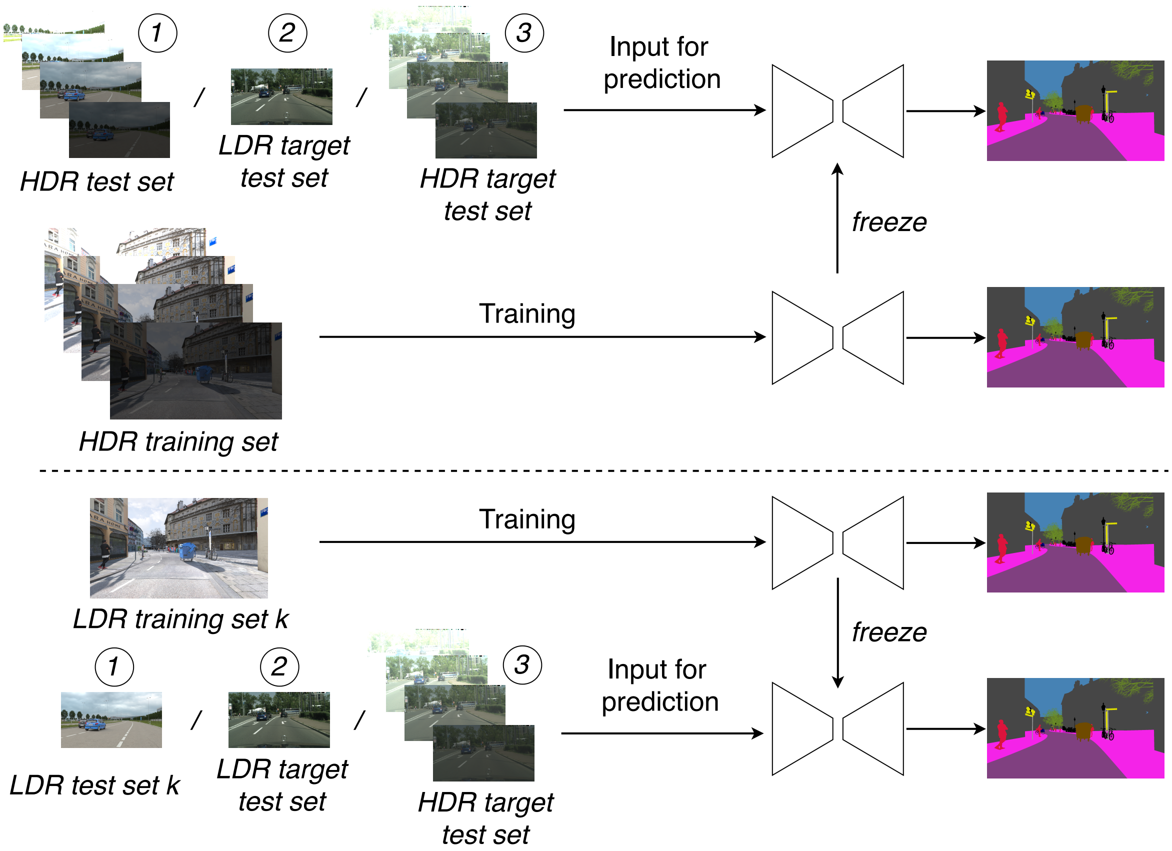


Figure 3.4 Overview of the semantic segmentation evaluation approaches. One semantic segmentation network is trained per image set, resulting in one network for the HDR training set and *k* networks for the LDR training sets. The weights of the networks are optimized until convergence and afterwards freezed to predict semantic labels for the three different evaluation tasks. For the first task images from the synthetic source domain are considered. For task two and three images from the real-world target domain are considered.

3.2.1 Closed system performance

For the first evaluation the semantic segmentation networks are used to predict semantic labels for images from the same domain as the training set. For example, the network trained on synthetic source HDR images is used to predict images from the synthetic source HDR test set. A network trained on a source LDR training set of a tone mapping operator *k* is used to predict labels for source LDR test images generated by the same tone mapping operator. This allows to determine the semantic performance for a closed system. This method intends to show how well HDR images or specific tone mapping operators are suitable for the task of a semantic segmentation under optimal conditions. This can often not be guaranteed for real-world automotive applications.

3.2.2 LDR generalization capability

It is very desirable to learn from synthetic images, as this approach could help making computer vision systems less expensive and more safe. Therefore, the second evaluation investigates how well the segmentation networks generalize on LDR images from the real-world target domain when trained on the synthetic source training set. This approach determines the generalization capability of the networks on real-world LDR target domain images. It should demonstrate, how well the network is able to transfer the knowledge from synthetic images to the real-world and therefore if it is conceivable to use synthetic images for training computer-vision systems in autonomous cars. This scenario is more useful in practice than the closed system performance as it allows to transfer the knowledge from synthetic images to the real-world.

3.2.3 HDR generalization capability

The third evaluation determines respectively the generalization capability of the segmentation networks on real-world HDR images. This measure intends to show if it is desirable to use HDR images instead of LDR images to train and test computer vision systems for automotive applications. If the HDR generalization capability reveals that it improves the performance of networks, it can be considered whether the benefit of HDR camera sensors outweigh the additional costs.

3.2.4 Semantic segmentation network

The architecture of the semantic segmentation network is chosen from one of the DeepLab models which are already presented in chapter 2.2.2. It takes images, for example, of street scenes as an input and learns to predict the semantic label of every pixel in the input image during training. In order to adapt the network, such that it is able to use the high-precision float HDR images as inputs, the deep learning framework PyTorch is more preferable than TensorFlow. Therefore, the DeepLab-v2 model, which is often used for domain adaptation experiments as seen in table 2.1, is selected as the semantic segmentation method as there is a free accessible PyTorch implementation available. To maximize the semantic segmentation performance there is carried out a hyperparameter study, which is described in section 4.2.2.

3.3 Deep domain adaptation via image-to-image translation

The third approach of this thesis is to improve the generalization capability of semantic segmentation networks. In section 2.3 recent methods to perform a domain adaptation between source and target images are categorized into generative and non-generative methods. In this thesis it is aimed to perform an image-to-image translation between synthetic source and real-world target images to improve the generalization performance of a network trained on synthetic source images. This would allow to a) improve the semantic segmentation performance on real-world images and furthermore to b) generate new appearance variations in the synthetic images. The benefit of generating new appearances from real-world images is that additional features may be learned with a neural network from real-world images without needing to label them. Therefore, the approach of this thesis is to generate a domain adapted dataset from a synthetic dataset. Furthermore, a semantic segmentation network is trained on this new dataset in order to improve the real-world generalization capability.

The approach can be seen in figure 3.5, where labeled synthetic source images are translated into the domain of unlabeled target images from the real-world. Thereby, only the appearance of the synthetic source domain image is modified without changing the structure of objects, which allows to transfer the semantic label images. With this approach a completely new domain adapted dataset is generated, which is used for training a semantic segmentation network. To evaluate the performance of the domain adaptation approach, the generalization capability of the network on real-world scenes is measured before and after the adaptation.

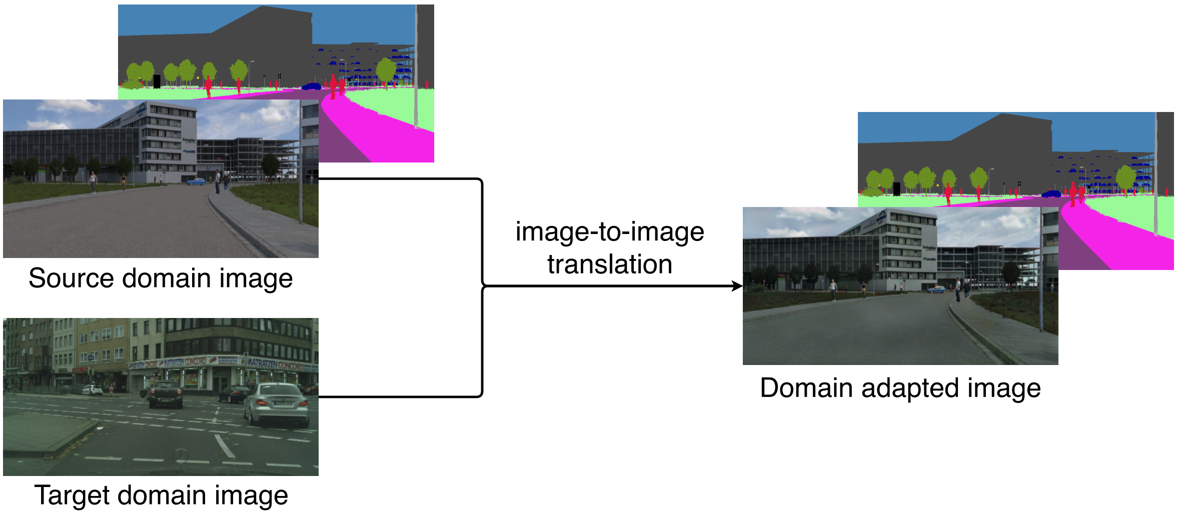


Figure 3.5 Exemplary overview of an image-to-image translation.

3.3.1 Selection of domain adaptation method

The DISE method from Chang et al. [2019] allows to generate realistic variations with seemingly no changes in structure, whereas many other approaches clearly disturb objects in the scene or produce unfavourable artifacts. This is enabled by their complex framework, which allows to split up the texture and structure of an image, which makes it possible to combine the texture of a real-world image with the structure of a synthetic image. This allows to generate a domain adapted dataset where the structure of the synthetic images remains and consequently the respective ground truth stays the same.

3.3.2 Domain adaptation of HDR images

This thesis approaches a domain adaptation from synthetic HDR to real-world HDR, as well as from synthetic LDR to real-world LDR. In addition, it will be investigated how a domain adaptation from synthetic HDR to real-world LDR or from synthetic LDR to real-world HDR behaves. The approach of this work is represented in figure 3.6, where the deep neural network DISE is exemplary shown by some fully connected layers, while the true network is described in subsection 4.3.

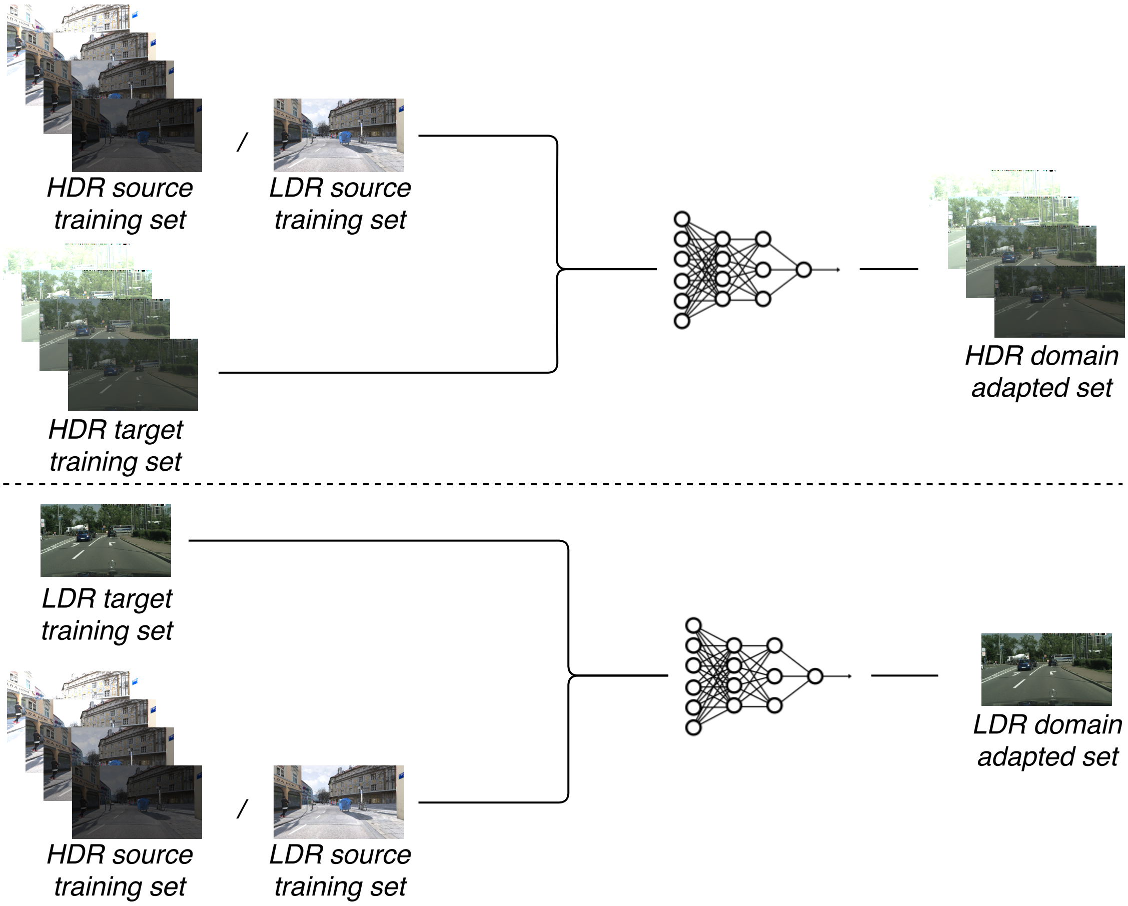


Figure 3.6 Overview of the domain adaptation approach, which is divided into four sub-approaches. For every approach one image set from the source domain and one from the target domain is used to train an image-to-image translation network.

3.3.3 Assessing the highest degree of realism

For this thesis the DISE method is used to perform a domain adaptation from synthetic to real-world. To determine the state of the model which generates the images with highest realism, the loss functions of the network cannot be used as they do not show many influences. For example, the network is able to predict trees in the domain adapted image where it actually should predict sky, which cannot be detected by the loss values. To improve the realism of the image-to-image translation network the convergence of the model is determined manually. This is done by visually comparing generated images from every 5,000 iterations and choosing the state of the model which generates the most preferable images. In this context, the main criteria to be fulfilled are fidelity in generated details (e.g. face contours, small letters in traffic signs), realisticness of road surface (e.g. patterns generated by the convolutional layers) and generated artifacts (e.g. size and number).

4 Experiments

Following, the experiments which are carried out in this thesis are described. In section 4.1 the experiments for tone mapping HDR images are explained. Afterwards, the experiments regarding the semantic segmentation with deep neural networks are presented in section 4.2. Last, in section 4.3 the domain adaptation experiments are described.

4.1 Tone mapping for semantic segmentation

Eight different tone mapping operators are used for experiments. Table 4.1 shows an overview of the selected operators.

Tone mapping operator	**Parameters**	**Default value**
Tumblin (revised) [Tumblin et al., 1999]	· Adaptation display luminance · Maximum display luminance · Maximum monitor contrast	[10,30] cd/m^2 100 cd/m^2 30 - 100
Ward histogram adjustment [Larson et al., 1997]	· Number of bins · Minimum display luminance · Maximum display luminance	100 1 cd/m^2 100 cd/m^2
Ward global [Ward, 1994]	· Maximum display luminance	100 cd/m^2
Drago [Drago et al., 2003]	· Maximum display luminance · Bias parameter	100 cd/m^2 [0,1]
Schlick (nonuniform) [Schlick, 1995]	· Minimum display luminance · Weight of nonuniformity	1 cd/m^2 [0,1]
Reinhard local [Reinhard et al., 2002]	· Sharpening parameter	8.0
Reinhard global [Reinhard et al., 2002]	-	
Linear clip	-	

Table 4.1 Overview of the eight tone mapping operators for experiments and their controllable parameters.

Two tone mapping operators are dependent on 0, 1, 2 and 3 parameters each. Every author provides a default value for the parameters of their operator which is either given as a value or an interval and with or without physical relationship. The Linear clip operator is a simple method which clips all values in the HDR image that are larger than 1.0 and linearly maps the remaining values to the 256 bins of the resulting LDR image. The implementation of the tone mapping operators is used from the MATLAB HDR Toolbox from Banterle et al. [2017].

4.1.1 Parameter optimization

For the first experiment, every tone mapping operator is used to maximize the tone mapped quality index (TMQI) performance for the first image of the investigated HDR dataset. The optimization is performed with a non-linear optimization algorithm, where the default values for each operator are used as an initial solution. The values determined for each parameter after the optimization are used to perform the tone mapping of the full HDR dataset, resulting in one LDR dataset for each tone mapping operator. The optimization for TMOs with more than one parameter is solved using the interior-point algorithm [Byrd et al., 2000] implemented in MATLAB's function *fmincon*. The one-dimensional optimization problems are solved using MATLAB's function *fminbnd*. The Ward histogram adjustment is optimized in two separate steps as it consists of one discrete parameter, which the interior-point algorithm cannot handle. Therefore, at first the two continuous parameters are optimized with the non-linear optimization method and afterwards the best possible discrete value is selected by performing a line search. The MATLAB function of the TMQI-method, which is required for the optimization, was accessed online from `https://ece.uwaterloo.ca/~z70wang/research/tmqi/` on the 12th April 2019.

4.1.2 Tone mapping ranking method for semantic segmentation

For further experiments a ranking score is computed for every tone mapping operator as described in subsection 3.1.3. The scores should help to decide which operators to select or exclude for the training. Nevertheless, all selected operators are used in this work. Consequently, the ranking method is evaluated after the experimental part of this thesis. The results from the ranking method are presented in subsection 5.1.2 and the ranking method is evaluated in section 5.3.

4.2 Semantic segmentation with HDR images

In this chapter the training and evalution of semantic segmentation networks is described. Those are used to measure the semantic segmentation performance of HDR images. Additionally, they are used to compare the performance between synthetic source and real-world target domain for the domain adaptation experiments.

4.2.1 Implementation details

Network

For experiments, a PyTorch implementation of the DeepLab-v2 with ResNet-101 is used. Weights are initialized using a pre-trained model on the Microsoft COCO dataset [Lin et al., 2014]. The weights of the last layer are reinitialized as the number of output classes is different in both tasks. To optimize weights, the cross-entropy loss is used, which is very common for the task of a semantic segmentation [Garcia-Garcia et al., 2018]. The code and pre-trained models were accessed online from `https://github.com/isht7/pytorch-deeplab-resnet` on the 24th April 2019.

Training details

The network is implemented on a single NVIDIA Quadro P6000 with 24 GB memory. Due to limited memory, the training and evaluation cannot be performed on full-size resolution images. During training the images are downsized maintaining their aspect ratio. For data augmentation, square-shaped image patches are cropped randomly out of three different possible positions, either tight-fitting to the left or right boundary of the image or aligned in the center of the image. Additionally, by a chance of 50 percent every patch is horizontally flipped.

During evaluation, the images are downsized while keeping their aspect ratio and cropped into three patches. The predictions of these patches are combined into one single image. This approach has already been used by several authors [e.g. Chen et al., 2018a]. For evaluation, the final prediction gets rescaled back to original image size.

Datasets

For experiments,the real-world Cityscapes (LDR & HDR) is used for evaluation, the synthetic Synscapes (LDR) for validation and the synthetic BIT (HDR) dataset for training. The datasets are explained in subsection 2.2.2. Only 11 out of the 19 Cityscapes classes are used for training. These are road, sidewalk, building, pole, traffic light, traffic sign, vegetation, terrain, sky, person and car.

The synthetic BIT HDR dataset is divided into 11,572 frames for training and 700 frames for testing. Additionally, it is used to generate the following datasets:

- LDR datasets
  Each tone mapping operator from table 4.1 is used to generate a new dataset. Therefore, the following eight different LDR datasets are generated by the TMOs
  - *Drago*
  - *Linear Clip*
  - *Reinhard global*
  - *Reinhard local*
  - *Schlick*
  - *Tumblin*
  - *Ward global*
  - *Ward histogram adjustment*
- Mixed LDR dataset for HDR approximation
  - *HDR*^{3}: Mixed LDR dataset, consisting of equally sized fractions of the LDR datasets from all eight tone mapping operators.
- HDR datasets
  For experiments, the HDR dataset is preprocessed with different kind of operations.
  - *HDR*: The original HDR dataset, with unchanged pixel values.
  - *HDR*^{1}: The HDR dataset with gamma corrected pixel values (γ = 2.2).

- *HDR^2*: The HDR dataset where every image is standardized to zero mean and a standard deviation of 1.
- *HDR^{1,2}*: The HDR dataset with gamma corrected and standardized pixel values. The gamma correction (γ = 2.2) is applied first.

Two visual examples for the tone mapped images can be seen in figure 7.1 and 7.2 in the appendix. As explained in subsection 2.1.1 LDR images contain pixel values that are gamma corrected. Without, monitors would not be able to properly display LDR images on their screen. As the semantic segmentation base network is pretrained on LDR images, a gamma corrected HDR dataset is generated. This should find out whether the semantic segmentation performance is influenced by such a preprocessing operation or not. Additionally, an HDR dataset with standardized images is generated, where each image has a standard deviation of 1 and a mean of 0.

4.2.2 Optimization of the semantic segmentation performance

A fine-tuning of hyperparameters is performed in order to search for the best possible configuration of the network for a semantic segmentation. The fine-tuning is performed using the real-world Cityscapes dataset. Since it can take very long to train a neural network until convergence, it would take an unapplicable amount of time to perform a dense grid-search for every hyperparameter of the network on a single GPU. Accordingly, in this study a coarse sampling of selected hyperparameters is performed.

The best performing hyperparameters of this study are used for further experiments with the synthetic BIT datasets. Repeating the hyperparameter study for every dataset would not be achievable during this thesis because of limited computational power.

4.2.3 Training and evaluation of the networks

To determine the semantic segmentation performance as described in section 3.2, one network is trained for every BIT dataset. This results in 13 semantic segmentation networks. Each network is trained until the validation loss does no longer decrease for 15,000 iterations, which is considered as *early stopping* [Zhang et al., 2016]. The first 500 frames from the Synscapes dataset are used for validation.

To measure the performance of the semantic segmentation the accuracy and the mIoU, as described in subsection 2.2.2, are determined from the predicted labels. The evaluation approach is described in 3.2. The *closed system performance* is determined on the test set of the respective BIT training dataset. The test set does not contain pixels of the classes traffic light and terrain, which decreases the performance for the mIoU measure. The *LDR generalization capability* is determined on the 8 bit and the *HDR generalization capability* on the 16 bit dataset from Cityscapes.

For the semantic segmentation networks, which are trained on the preprocessed BIT HDR datasets, the same preprocessing operations are applied during evaluation. This does not belong to the gamma correction as preprocessing operation, if the input is a LDR image. It is very unusual for machine learning tasks to apply different preprocessing operations during training and testing. However, the knowledge which kind of sensor is used to capture images, may be a reasonable preliminary information. This should allow to determine whether LDR or HDR images are to be expected for the network.

4.3 Domain adaptation for semantic segmentation

4.3.1 Implementation details

The DISE method from Chang et al. [2019] is implemented on a single NVIDIA Quadro P6000 with 24 GB memory. The default values for the hyperparameters are used for training, as described by Chang et al. [2019]. Cropping was performed with an image size of 320x640 pixels. The *Image* module from Pillow [PIL, 2019] was replaced with functions from OpenCV to read and process images in Python, as it is not capable of HDR images. The implementation was accessed online from `https://github.com/a514514772/DISE-Domain-Invariant-Structure-Extraction` on the 16th April 2019.

4.3.2 Domain adaptation with HDR images

To determine the performance on real-world images, multiple semantic segmentation networks are evaluated on the Cityscapes dataset, as described in subsection 4.2.3. To improve the performance on real-world images, a domain adaptation is carried out. For this, one synthetic BIT LDR dataset and one synthetic BIT HDR dataset is used to learn an image-to-image translation with the DISE network. The real-world Cityscapes LDR and HDR dataset is used for the target domain. The following experiments should determine whether an improvement between different domains from different dynamic ranges is possible:

- *Linear clip* $\rightarrow$ *LDR^t*: The synthetic LDR images from the Linear clip operator are used for the source domain. The real-world LDR images from Cityscapes are used for the target domain. The DISE network is used to translate the synthetic LDR images into real-world LDR images.
- *Linear clip* $\rightarrow$ *HDR^t*: The synthetic LDR images from the Linear clip operator are translated into real-world HDR images.
- *HDR^1* $\rightarrow$ *LDR^t*: Gamma corrected synthetic HDR images are translated into real-world LDR images (target domain).
- *HDR^1* $\rightarrow$ *HDR^t*: Gamma corrected synthetic HDR images are translated into gamma corrected real-world HDR images.

The networks are trained until the highest degree of realism is assessed by the approach as described in subsection 3.3.3. Then, the network is used to translate the respective synthetic BIT training sets into the target domain.

One semantic segmentation network is trained for each of the four resulting *domain adapted* datasets. To determine the domain adaptation performance, the *LDR generalization capability* is measured for the datasets which are translated into the LDR target domain. The *HDR generalization capability* determines the performance for the datasets which are translated into the HDR target domain. Comparing the semantic segmentation performances before and after domain adaptation shows the impact of it.

5 Results

Following, the experimental results from the quantitative research of this thesis are presented. Section 5.1 provides the results from the proposed tone mapping ranking method, which estimates the order of the semantic segmentation performance of neural networks trained on LDR datasets from different tone mapping operators. Section 5.2 presents the actual performances of the different semantic segmentation networks. Afterwards, in section 5.3 the proposed ranking method is being evaluated. Finally, section 5.4 shows the results of a domain adaptation from synthetic to real-world with HDR and LDR images.

5.1 Tone mapping of HDR images

The first experiments were to generate multiple LDR datasets from one HDR dataset with several tone mapping operators. Subsection 5.1.1 provides the results of the non-linear optimization to optimize the parameters for each TMO. In subsection 5.1.2 the results of the tone mapping ranking method are provided.

5.1.1 Parameter optimization

The parameter optimization with the TMQI method reduces the computational time from multiple weeks – training many semantic segmentation models with different parameter settings for each TMO – on a high cost GPU, to less an hour – optimizing each operator for the TMQI – using a regular CPU. Figure 5.1 shows exemplary the optimization for the Drago tone mapping operator. The TMQI values are computed and displayed in a regularly spaced grid, while the steps of the interior-region optimization algorithm are displayed as a white line. The starting point is shown

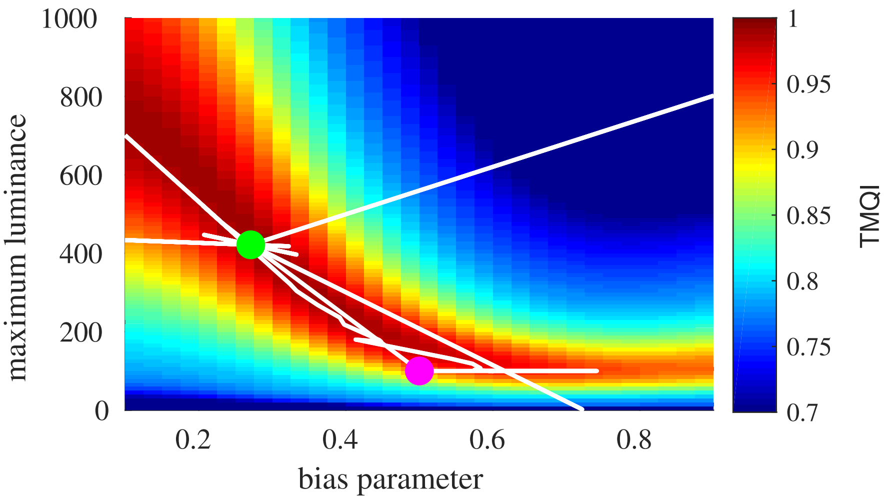


Figure 5.1 The steps of the interior-point algorithm, which is used for the multi-dimensional optimization problem, are visualized as a white line. The grid shows the TMQI values for the corresponding parameter values, the maximum is shown as a green circle and the starting point as a magenta circle.

by a magenta circle and the determined maximum of the TMQI function by a green one.

An overview of the initial values for each tone mapping operator for the nonlinear optimization, the constraints for each parameter, as well as the final determined values for each parameter are given in table 5.1.

Tone mapping operator	Parameter shortcut	Initial value	Determined value	Lower bound	Upper bound
Tumblin (revised)	$\mathrm{L_{da}}$	20.00 cd/m^2	0.0260 cd/m^2	0.00	∞
	$\mathrm{Ld_{max}}$	100.00 cd/m^2	25.1765 cd/m^2	0.00	∞
	$\mathrm{C_{max}}$	65.00	31.7410	0.00	∞
Ward histogram adjustment	$\mathrm{n_{bin}}$	100	80	5	256
	$\mathrm{Ld_{min}}$	1.00 cd/m^2	3.9015 cd/m^2	0.00	10.00
	$\mathrm{Ld_{max}}$	65.00 cd/m^2	100.7479 cd/m^2	10.00	∞
Ward global	$\mathrm{Ld_{max}}$	-	1.5414 cd/m^2	0.00	1000.00
Drago	$\mathrm{Ld_{max}}$	100.00 cd/m^2	110.0935 cd/m^2	0.00	∞
	$\mathrm{d_b}$	0.50	0.0035	0.00	1.00
Schlick (nonuniform)	$\mathrm{Ld_{min}}$	1.00 cd/m^2	0.0619 cd/m^2	0.00	∞
	k	0.50	0.9987	0.00	1.00
Reinhard local	ϕ	-	99.4975	0.00	100.00
Reinhard global	-				
Linear clip	-				

Table 5.1 Results from the parameter optimization.

5.1.2 Evaluation of tone mapping operators with ranking method

To compare the performance between the tone mapping operators, the proposed ranking method from subsection 3.1.3 is applied to the tone mapped datasets. This requires to determine following evaluation metrics

- *Overall image quality (OIQ) score*
- *Image dynamics (ID) score*
- *Tone mapping sensitivity (TMS) score*

whose results are presented in the following.

Overall image quality scores

To determine the OIQ score the TMQI and BRISQUE values are computed for every tone mapped image in the dataset. The result of the TMQI values can be seen exemplary for the Drago operator (left) and the Ward histogram adjustment operator (right) in figure 5.2. While the TMQI values of the tone mapped images from the Drago operator have a much higher variation, the values of the images from the Ward operator remain more constant. The values from the Ward histogram adjustment operator are mostly higher, meaning the images have a better overall image quality in average, as measured by the TMQI evaluation metric.

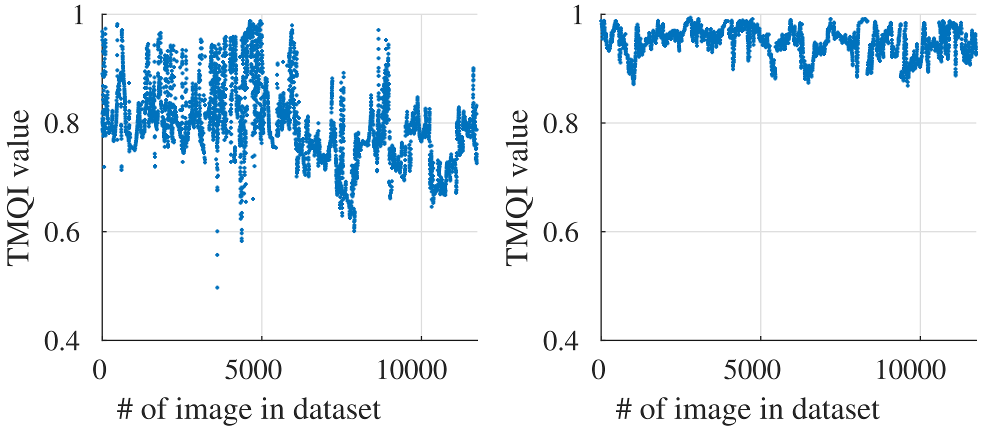


Figure 5.2 The figure shows the TMQI values for every tone mapped image in the LDR dataset from the Drago tone mapping operator (left) and the Ward histogram adjustment operator (right).

Similar, the BRISQUE values are computed for every tone mapped image in the dataset. From these values the respective mean value is computed by equations 3.1 and 3.3. The average of the TMQI values can be seen in figure 5.3 and the one of the BRISQUE values is shown in figure 5.4. It should be noticed that the TMQI values are defined in the interval from 0 to 1, where higher values mean better overall image quality. In contrast, for the BRISQUE evaluation metric smaller values correspond to a better image quality. It can be seen, that the Drago operator clearly has

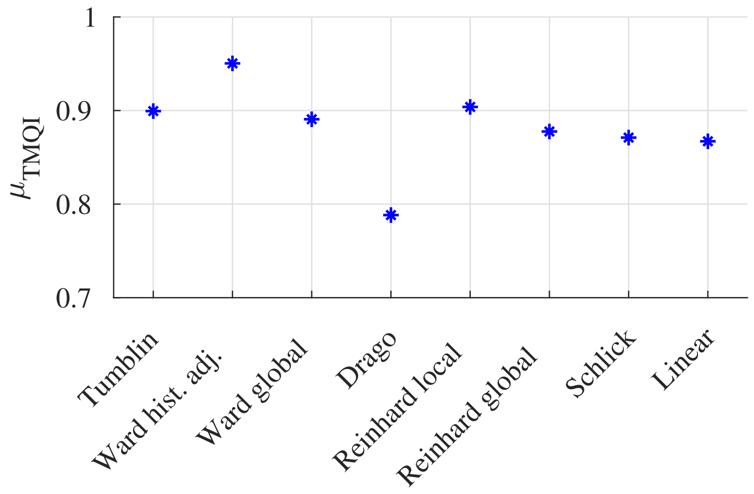


Figure 5.3 Arithmetic mean of the TMQI values for every tone mapped image of the respective tone mapping operator.

the lowest mean TMQI values. This means that the generated LDR images have a lower image quality in comparison to the other operators in average, according to this evaluation metric. A graphical evaluation of the generated images shows, that the Drago operator generates images with a strong variation in exposure, while all of the other operators produce more similar exposed ones.

Scores are computed from the TMQI and BRISQUE mean values by equations 3.2 and 3.4. The scores correspond to a linear interpolation between lowest and highest image quality values. The

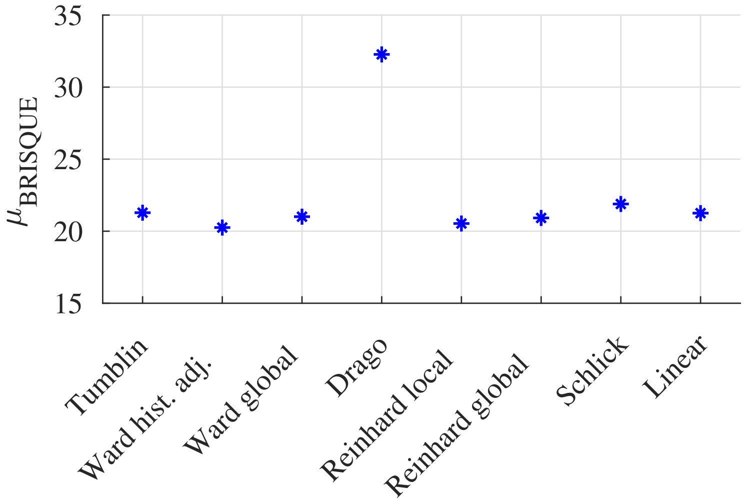


Figure 5.4 Arithmetic mean of the BRISQUE values for every tone mapped image of the respective tone mapping operator.

scores are visualized in figure 5.5, where higher scores correspond to a better image quality. As can be seen, the Ward histogram adjustment operator performs best, while the Drago operator performs worst for those evaluation metrics.

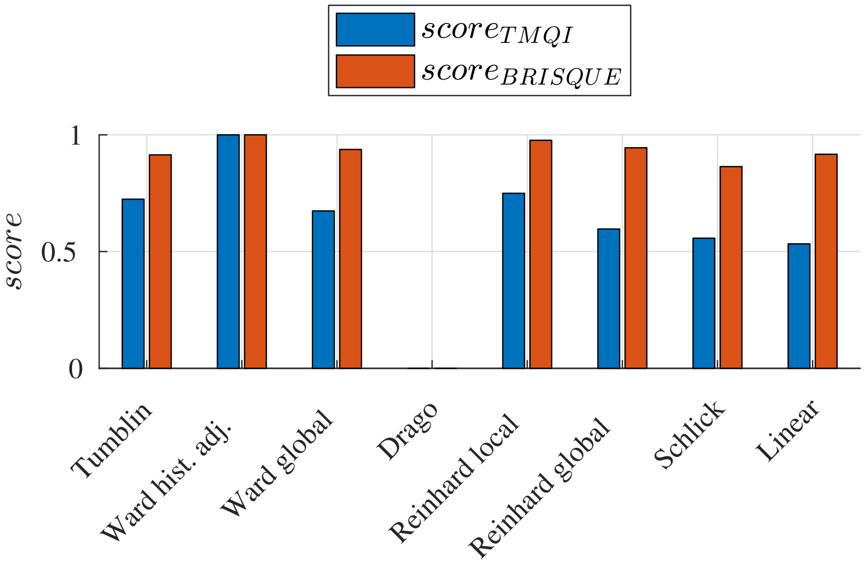


Figure 5.5 Scores for the TMQI and BRISQUE evaluation metrics.

The scores for the overall image quality evaluation metric are determined by equation 3.5. The results are shown in figure 5.6.

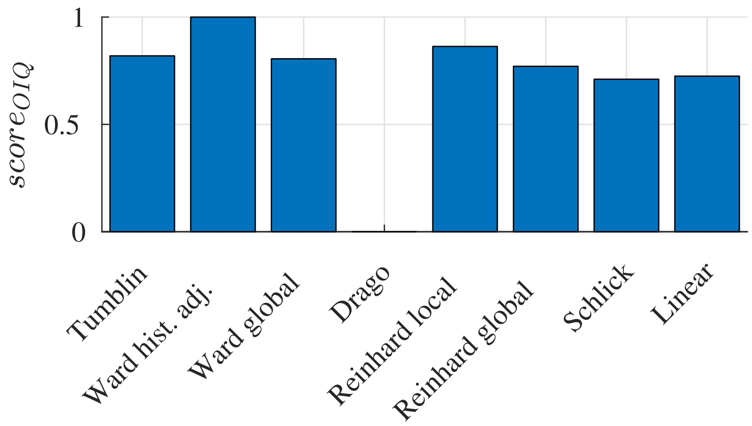


Figure 5.6 Scores for the overall image quality (OIQ) evaluation metric.

Image dynamics scores

For the computation of the image dynamics scores the luminance histogram of every tone mapped image is calculated by equation 3.6. For each histogram the skew and kurtosis value is computed. The respective mean values for every tone mapping operator are calculated from equations 3.7 and 3.9. The mean skew values are shown in figure 5.7 and the mean kurtosis values are displayed in figure 5.8. The Drago operator has clearly the largest skew and kurtosis values meaning that there are large peaks on the right side of the histogram. This means that a large amount of grey values is close to the maximum grey value, which corresponds to very bright pixels. Figure 7.2 in the appendix shows an example, where the Drago operator generates rather overexposed LDR images.

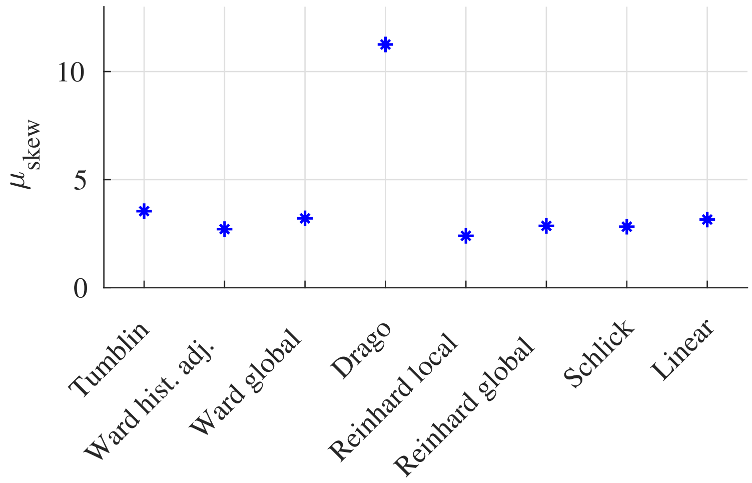


Figure 5.7 Arithmetic mean of the skew values for every tone mapped image of the respective tone mapping operator.

Similarly as for the overall image quality, scores are computed for these values. They are determined by equations 3.8 and 3.10 and can be seen in figure 5.9.

The final score for the image dynamics is computed by equation 3.11 and the results can be seen in figure 5.10. Again the Drago operator performs worst with a score of 0. For this evaluation

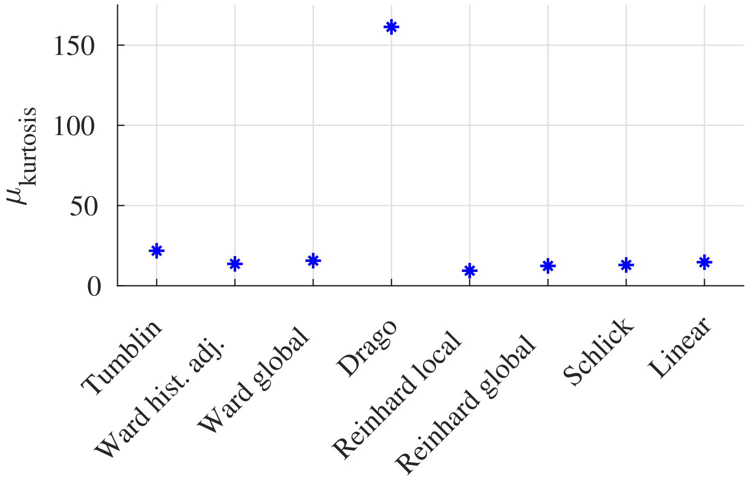


Figure 5.8 Arithmetic mean of the kurtosis values for every tone mapped image of the respective tone mapping operator.

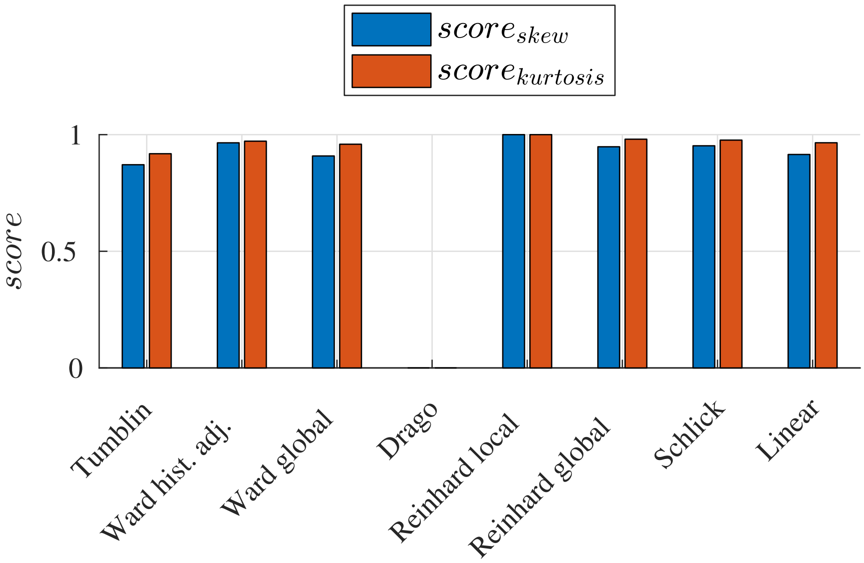


Figure 5.9 Scores for the skew and kurtosis evaluation metrics.

metric the Reinhard local operator performs best. Nevertheless, the differences between the seven best performing operators are only small compared to the difference to the Drago operator.

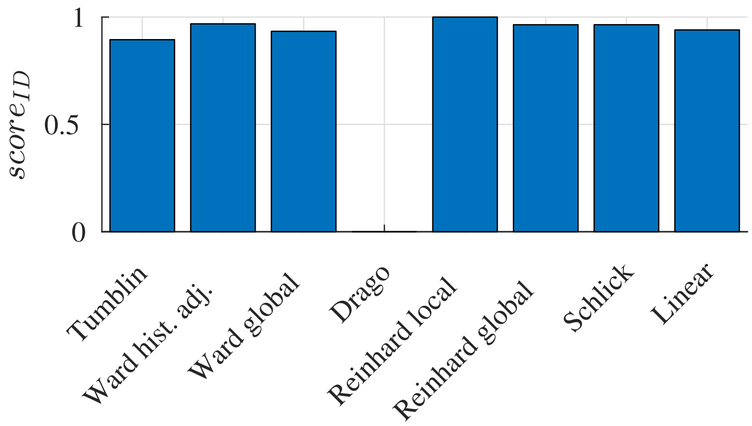


Figure 5.10 Scores for the image dynamics (ID) evaluation metric.

Tone mapping sensitivity scores

The last evaluation metric for the tone mapping ranking method is the sensitivity of the operators with respect to different images. For this, the standard deviations of the TMQI and BRISQUE values are computed by equations 3.12 and 3.14. The result for the TMQI values is shown in figure 5.11 and the one for the BRISQUE values can be seen in figure 5.12.

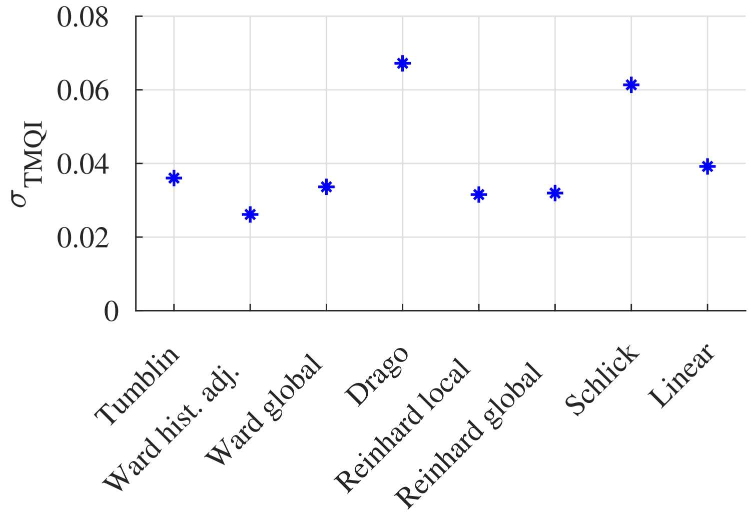


Figure 5.11 Standard deviation of the TMQI values for every tone mapped image of the respective tone mapping operator.

Smaller values correspond to a better sensitivity score. The respective scores are computed by equations 3.13 and 3.15 and can be seen in figure 5.13.

In contrast to the overall image quality and image dynamics metrics, there are larger differences between the results of each TMO. The tone mapping sensitivity scores are determined by equation 3.16 and are shown in figure 5.14. While the Drago operator performs again worst with a score of 0, the Schlick operator shows a reduced performance as well.

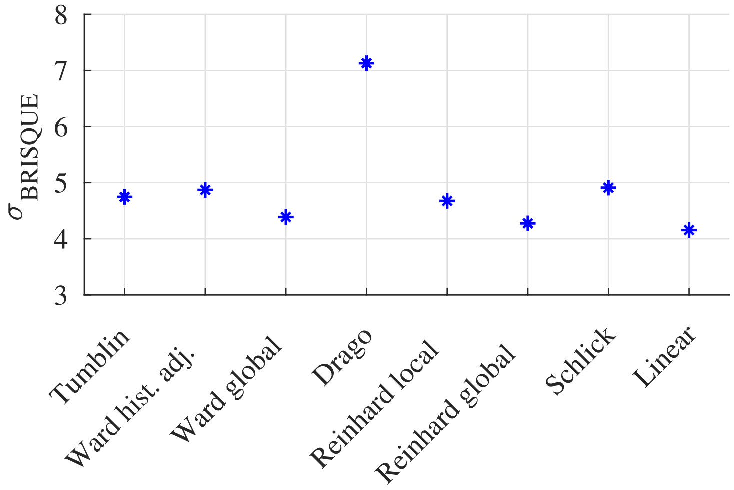


Figure 5.12 Standard deviation of the BRISQUE values for every tone mapped image of the respective tone mapping operator.

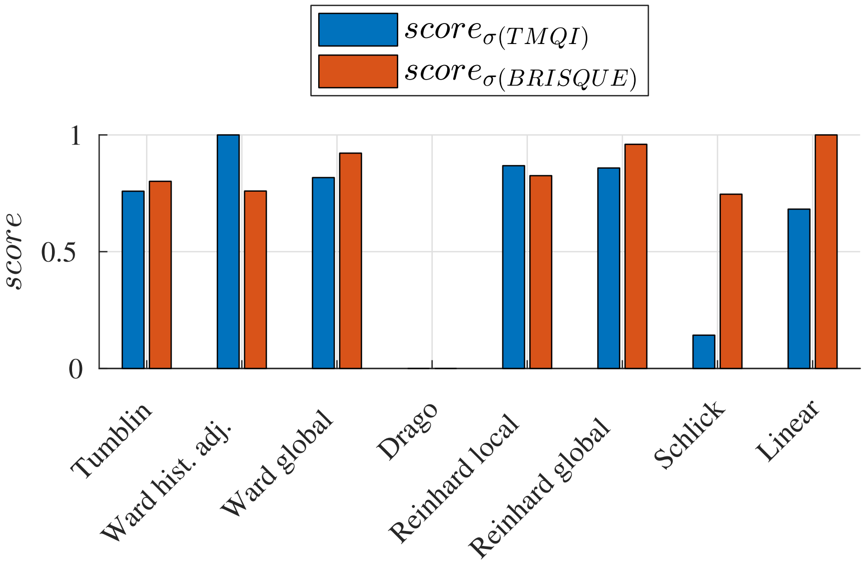


Figure 5.13 Scores for the TMQI and BRISQUE standard deviation evaluation metrics.

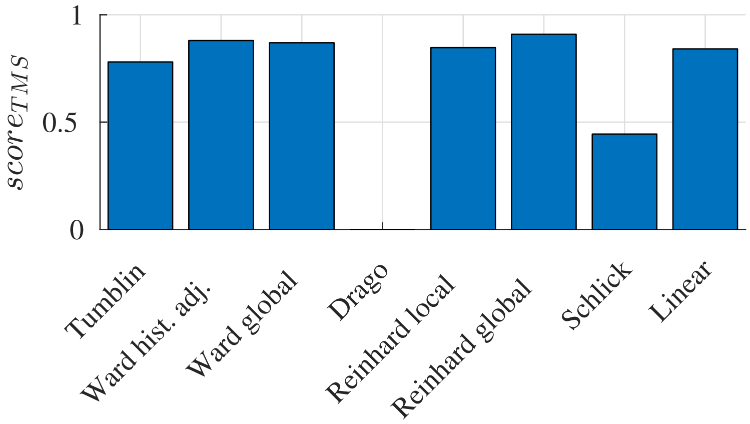


Figure 5.14 Scores for the tone mapping sensitivity (TMS) evaluation metric.

Final ranking scores

The scores from the three different evaluation metrics can be seen in figure 5.15. For every method the Drago operator performs worst with a score of 0. The second worst operator is the Schlick operator as measured by these evaluation approaches. For the other six operators the scores are continuously rather high.

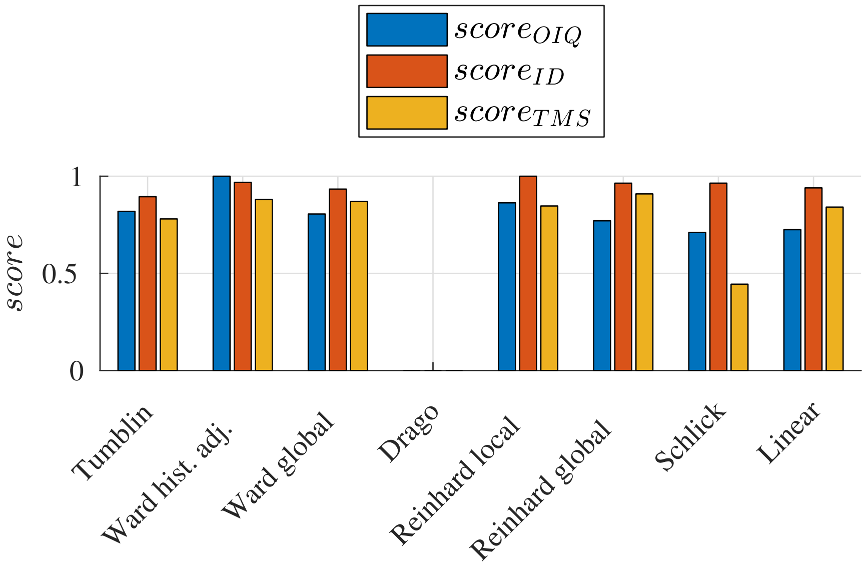


Figure 5.15 Overview of the scores for the overall image quality (OIQ), image dynamics (ID) and tone mapping sensitivity (TMS) scores.

The final score for each tone mapping operator is calculated by equation 3.17. The scores are shown in figure 5.16. Best performing TMO by this tone mapping ranking method is the Ward histogram adjustment operator. The clearly worst performing TMO is the Drago operator. For all operators in between there are only small differences. According to these scores the Drago operator may perform clearly worst for a semantic segmentation while all the other operators may perform similarly good.

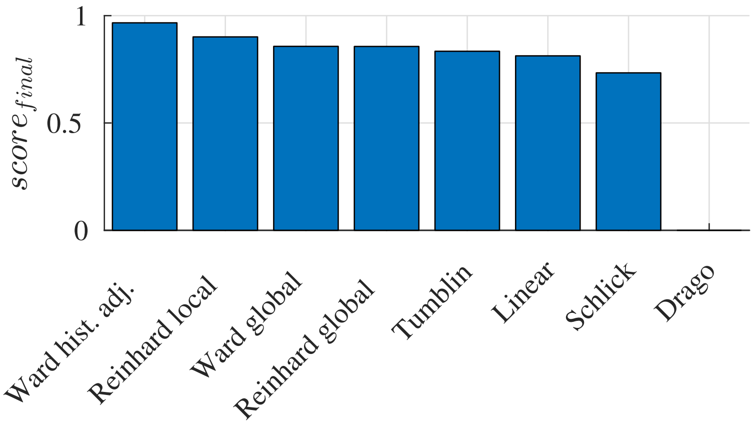


Figure 5.16 Ranking scores for the final tone mapping ranking method.

5.2 Deep semantic segmentation

In this section the experimental results of the semantic segmentation are presented. In subsection 5.2.1 the results from the hyperparameter study are shown and in subsection 5.2.2 the semantic segmentation performances of all the trained networks are contained.

5.2.1 Hyperparameter study results

For experiments, a fine-tuning of hyperparameters is performed in order to search for the best possible configuration of the semantic segmentation network. For this hyperparameter study all loss curves are computed by averaging the validation loss over 2975 iterations, which corresponds to one epoch on the Cityscapes training set. In each iteration the validation loss is computed on a random image from the Cityscapes validation set. Since the ground truth of the Cityscapes *test* set is not available, all of the evaluations are performed on the *val* set. Because of limited memory a batch size of 1 is used during training.

Baseline

The baseline model is trained on the Cityscapes *train* set for 238,000 iterations (80 epochs). It is optimized through the stochastic gradient descent algorithm with momentum and an initial learning rate of 0.0001 with a weight decay of 0.0005. During training the cropped image patches are rescaled randomly by a factor between 0.5 and 1.5 with a base image size of 470. This has been implemented for the DeepLab-v2 model in order to achieve robustness to different object sizes. The learning rate decays linearly from 0.0001 to 0.00002, since the training is terminated before the maximum number of 297,500 iterations (100 epochs), where the learning rate 0.00000, is reached. During evaluation the test images are divided into patches of size 600x600 pixels.

Loss versus accuracy

The first experiment in the hyperparameter study aims to investigate the relationship between loss and accuracy. The baseline model is evaluated after several stages during training. Figure 5.17 shows the loss and the mIoU on the validation set for multiple training stages of the model. A clear correlation can be seen between loss and mIoU showing that both measures are converging similarly during training.

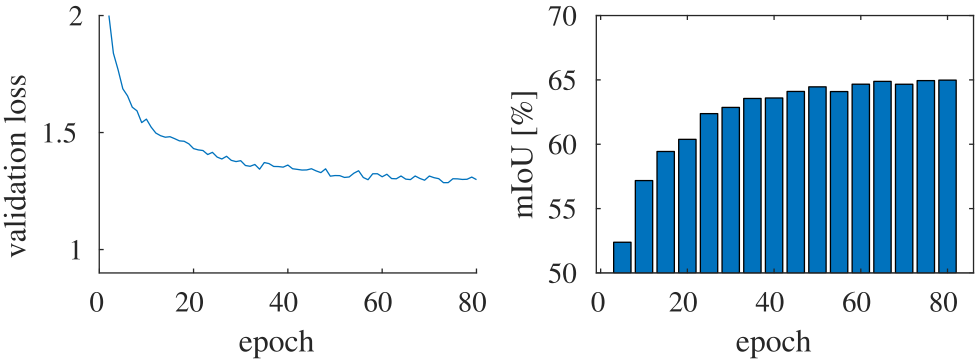


Figure 5.17 Left can be seen the loss on the validation set averaged over 1 epoch, while right shows the respective mIoU on the validation set.

Optimizer and learning rate

For the next experiment during the hyperparameter study different optimizers are used. The stochastic gradient descent (SGD) [Kiefer & Wolfowitz, 1952] and the ADAM [Kingma & Ba, 2014] optimizer are two commonly used techniques. Both optimization methods are investigated with different learning rates. As the amount of GPUs was limited during this work, no full training until convergence could be carried out. Therefore, every model is only trained for 15 epochs. Figure 5.18 shows the results of the investigations. The learning rates are chosen in a way, that the validation loss has its minimum somewhere between highest and lowest investigated learning rate. For the SGD method, the optimum learning rate lies somewhere around 0.001 and for the ADAM optimizer around 0.000005. Both learning rates are used to train a semantic segmentation network until convergence. The loss on the validation set can be seen in figure 5.19. The loss

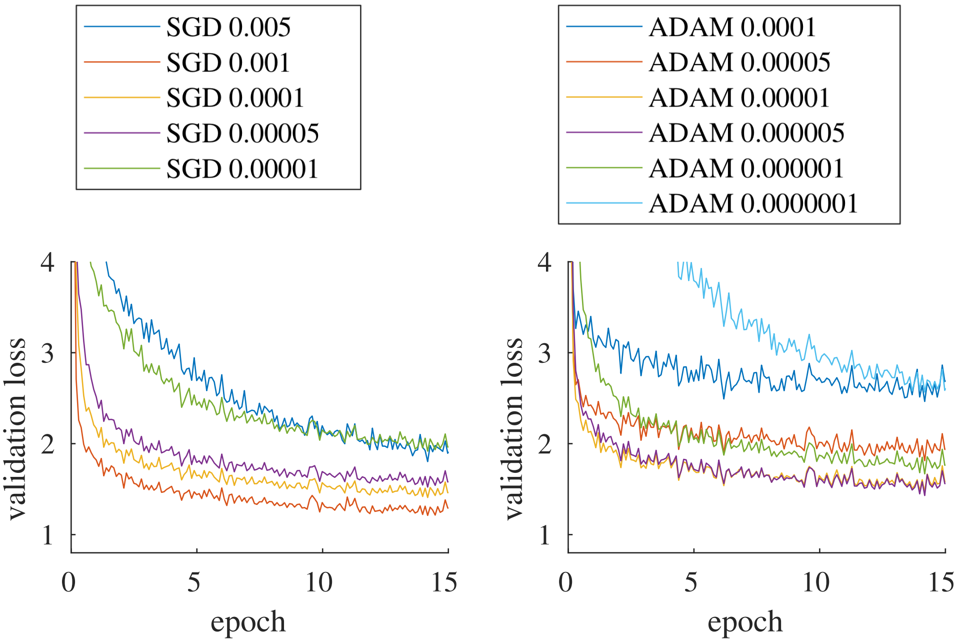


Figure 5.18 Loss on the validation set for the stochastic gradient descent with momentum (left) and the ADAM optimizer (right) with different learning rates.

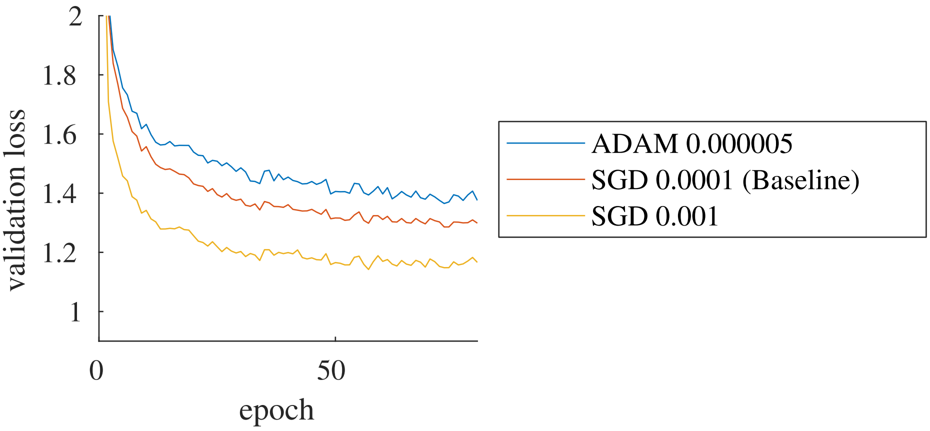


Figure 5.19 Loss on the validation set with the best learning rate for the SGD and the best one for the ADAM optimizer. As a reference the loss from the baseline is visualized.

of the SGD with a learning rate of 0.001 is much smaller than the one of the basemodel and the one of the ADAM optimizer.

Table 5.2 shows the mIoU class scores for the three different models after 80 epochs.

Method	Road	Sidewalk	Building	Wall	Fence	Pole	Traffic Light	Traffic Sign	Vegetation	Terrain
Baseline	**0.97**	0.77	0.89	**0.44**	0.41	0.40	0.49	0.62	0.89	0.56
Adam	**0.97**	0.76	0.88	0.32	0.33	0.40	0.45	0.63	0.89	0.52
SGD	**0.97**	**0.80**	**0.90**	**0.44**	**0.47**	**0.46**	**0.57**	**0.68**	**0.90**	**0.59**

Method	Sky	Person	Rider	Car	Truck	Bus	Train	Motorcycle	Bicycle	mIoU
Baseline	**0.92**	0.70	0.45	0.91	**0.63**	**0.70**	**0.49**	**0.45**	0.66	0.65
Adam	0.91	0.70	0.41	0.90	0.33	0.34	0.00	0.26	0.66	0.56
SGD	**0.92**	**0.74**	**0.51**	**0.92**	0.58	**0.70**	0.32	**0.45**	**0.68**	**0.66**

Table 5.2 Baseline: SGD with learning rate of 0.0001; Adam with lr of 0.000005; SGD with lr of 0.001; Each model is evaluated after 80 epochs

The SGD with a learning rate of 0.001 has the highest mIoU scores for most of the evaluated classes. The mIoU averaged over all classes is 66%, while it is 1% lower for the baseline and 10% lower for the ADAM optimizer. Therefore, the SGD with a learning rate of 0.001 is used for the following experiments.

Switching from Adam to SGD

In the study from Keskar & Socher [2017] they show that changing the optimization method during training improves the generalization capability. Figure 5.20 shows the loss on the validation set for changing from ADAM with a learning rate of 0.000005 (red) to SGD with a learning rate of 0.0001 (yellow) during training. As reference, the loss of the current model is shown in blue. This experiment has been repeated, switching from ADAM with a learning rate of 0.000005 to SGD

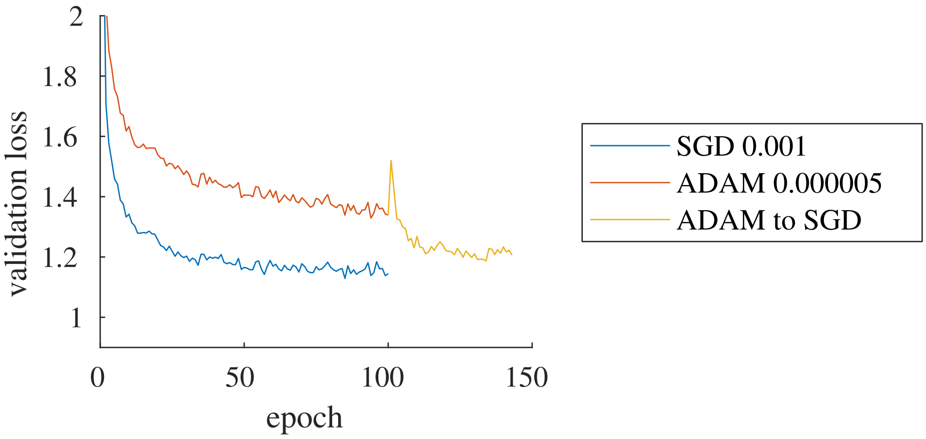


Figure 5.20 Results for switching from ADAM to SGD.

with a learning rate of 0.001 after 80 epochs. However, the validation loss directly exploded after the change of the optimizer and it took nearly 70 epochs until the loss has converged again. The two models are also evaluated by the class-wise mIoU score.
Switching after 100 epochs from ADAM to SGD with a learning rate of 0.0001 takes 34 more epochs until convergence and results in a mIoU of 0.65%. Switching after 80 epochs from ADAM to SGD with a learning rate of 0.001 takes 66 more epochs and results in a mIoU of 0.64%. For both experiments the outcome shows no improvement compared to the SGD with 0.001.

Learning rate scheduler

For the next experiments different schedulers for the learning rate are investigated. The learning rate scheduler determines how the value of the learning rate changes during training. For all previous experiments the learning rate is decayed linearly. For following experiments the learning rate is selected constant and decayed exponentially. Additionally, a cyclic learning rate is performed, which has shown to improve the capabilities of a neural network [Smith, 2017]. The cyclic learning rate depends on three parameters. One for the cycle length and two more for the maximum and minimum value of the learning rate. During training the learning rate linearly changes from the minimum to the maximum learning rate and back for one cycle. After the first cycle, the same process repeats until no more reduction of the loss can be obtained. Figure 5.21 shows the result of the investigations. The parameters for the cyclic 1 learning rate are determined by a preliminary experiment, as explained by the authors from Zhang et al. [2016]. The parameters of the cyclic 2 learning rate are selected from figure 5.18. Nevertheless, for this experiment the linear learning rate results in the lowest validation loss values for most of the time during training.

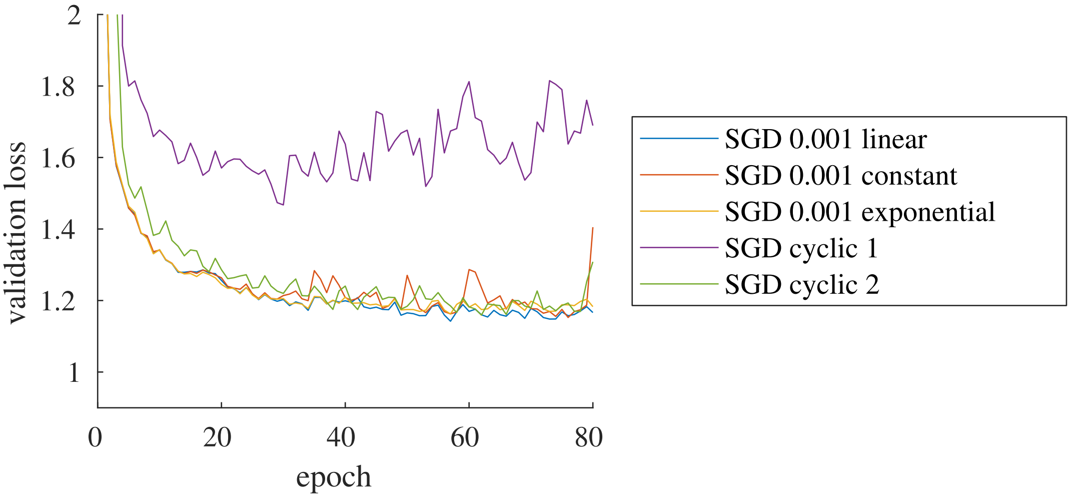


Figure 5.21 Results with different learning rate schedulers.

Input image size

For the next hyperparameter investigation the input image size is changed. In the baseline configuration a scale augmentation is performed, with a base input image size of 470x470 pixels and a random rescaling with a factor between 0.5 and 1.5. For this experiment no random rescaling is performed and a base input size of 700x700 pixels is selected. The performance can thereby not be improved, but the mIoU of 66% is already achieved after 21 instead of 80 epochs, which allows to reduce the training time by a large factor.

Evaluation image size

The same model is evaluated with different image patch sizes, which are fused into the final image for the mIoU assessment. This brings +1% in performance, by using 700x700 patches instead of 600x600 for the prediction. The final performance is 67% on the Cityscapes validation set.

Class-weights for cross-entropy loss function

For the last experiment, only eleven out of the nineteen Cityscapes classes are used for evaluation. The BIT datasets, which are used for further experiments, contain only pixels from these classes. Therefore, this investigation aims to improve the class-wise predictions of these 11 classes by weighting the cross-entropy loss values during training according to the occurance of their pixels.

The class weights are computed by

$$w_i = 1 - \frac{n_i}{n} \tag{5.1}$$

with n_i being the number of pixels of class *i* and *n* being the total amount of pixels from all classes. The computed class weights are shown in table 5.3.

	Road	Sidewalk	Building	Pole	Traffic Light	Traffic Sign	Vegetation	Terrain	Sky	Person	Car
Weight	0.6104	0.9441	0.7732	0.9847	0.9980	0.9931	0.8207	0.9914	0.9653	0.9866	0.9326

Table 5.3 Class weights for cross-entropy loss.

Table 5.4 shows the performance of the semantic segmentation network trained with an unweighted and a weighted cross-entropy loss function. The convergence was terminated with the early stopping method, because the loss significantly increased for this experiment, which may be reasoned by the smaller amount of classes. As can be seen the results are almost identical.

Method	Road	Sidewalk	Building	Pole	Traffic Light	Traffic Sign	Vegetation	Terrain	Sky	Person	Car	mIoU
Unweighted	**0.98**	**0.82**	**0.92**	0.49	**0.58**	**0.72**	**0.91**	**0.60**	**0.93**	**0.79**	**0.94**	**0.79**
Weighted	**0.98**	0.81	**0.92**	**0.51**	**0.58**	**0.72**	**0.91**	**0.60**	**0.93**	**0.79**	**0.94**	**0.79**

Table 5.4 Semantic segmentation performance with the weighted and unweighted cross-entropy loss function.

Final hyperparameter setting

The final hyperparameters are determined as follows. The stochastic gradient descent with momentum and a learning rate of 0.001 with a weight decay of 0.0005 is used for optimization.

Convergence of the network is determined by the early stopping method. Images for training are rescaled into 700x700 pixels with no scale augmentation. Images for testing are divided into three patches with 700x700 pixels, which are combined for the final evaluation. Gradients are computed by an unweighted cross-entropy loss function.

5.2.2 Semantic segmentation performances

For the following experiments one semantic segmentation network is trained for each of the 13 datasets, which are described in section 4.2. The validation loss is computed during training after every 500 iterations on the validation set until the convergence of the network is determined by the early stopping method. Every network is trained two individual times. The semantic segmentation performance is then averaged to achieve a more robust estimation of the performance. The individual values can be seen in the appendix, in table 7.1 for the closed system performance, in table 7.2 for the LDR generalization capability and in table 7.3 for the HDR generalization capability.

Closed system performance

Figure 5.22 shows the semantic segmentation performances of the networks as measured by the closed system performance. The names on the x-axis show, on which dataset the networks were trained. The evaluation was performed on the respective test set of the same dataset. The performances are determined by the accuracy (blue, left axis) and by the mIoU (orange, right axis). The first eight networks were trained on the tone mapped datasets, which are LDR ones.

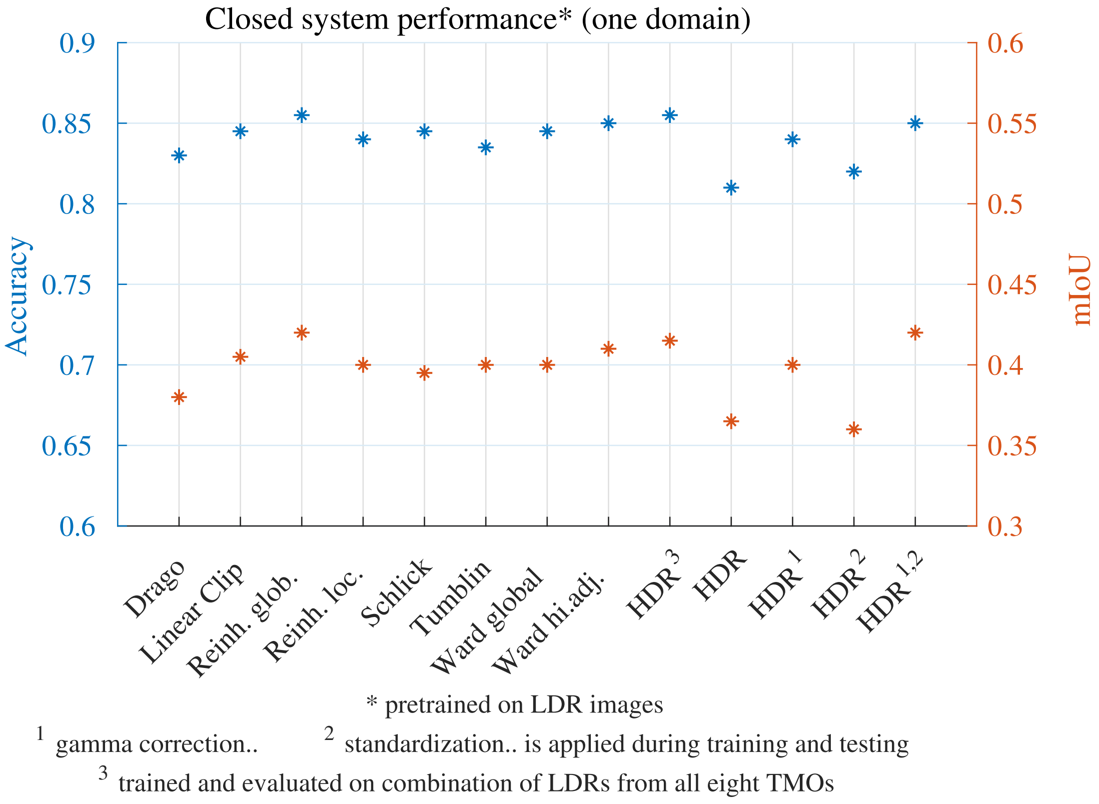


Figure 5.22 Evaluation of the semantic segmentation networks listed on the x-axis by the closed system performance.

The HDR^3 corresponds to the network trained on the mixed LDR dataset for the HDR approximation and the last four networks were trained on the HDR datasets with different preprocessing operations. In the following, each network is named according to the dataset it was trained on.

The best performing network is the Reinhard global, followed by the HDR^3 and the HDR^{1,2} networks. The Drago, HDR and HDR^2 are the three worst performing ones. For the closed system performance, where the preprocessing operation is the same for the training and test set, the gamma correction improves the accuracy by 3% and the mIoU by 3.5%. The results of this experiment show, that there are quite large differences between the semantic segmentation performances, although every network was trained on images of the exact same scenes.

LDR generalization capability

In figure 5.23 the results of the LDR generalization capability are shown, measuring the performance for real-world LDR images. The difference between best and worst performing network is much larger than for the closed system performance. Again, the Drago, HDR and HDR^2 are the three worst performing networks. For this experiment the HDR^3 network shows the highest performance in total. The best performing tone mapping operators are the Tumblin, Ward global, Ward histogram adjustment and once again the Reinhard global network. The model trained on the raw HDR pixel values performs even worse, than the worst performing from the networks trained on LDR images. Contrarily, the HDR^1 network with gamma corrected pixel values outperforms four of the eight LDR networks.

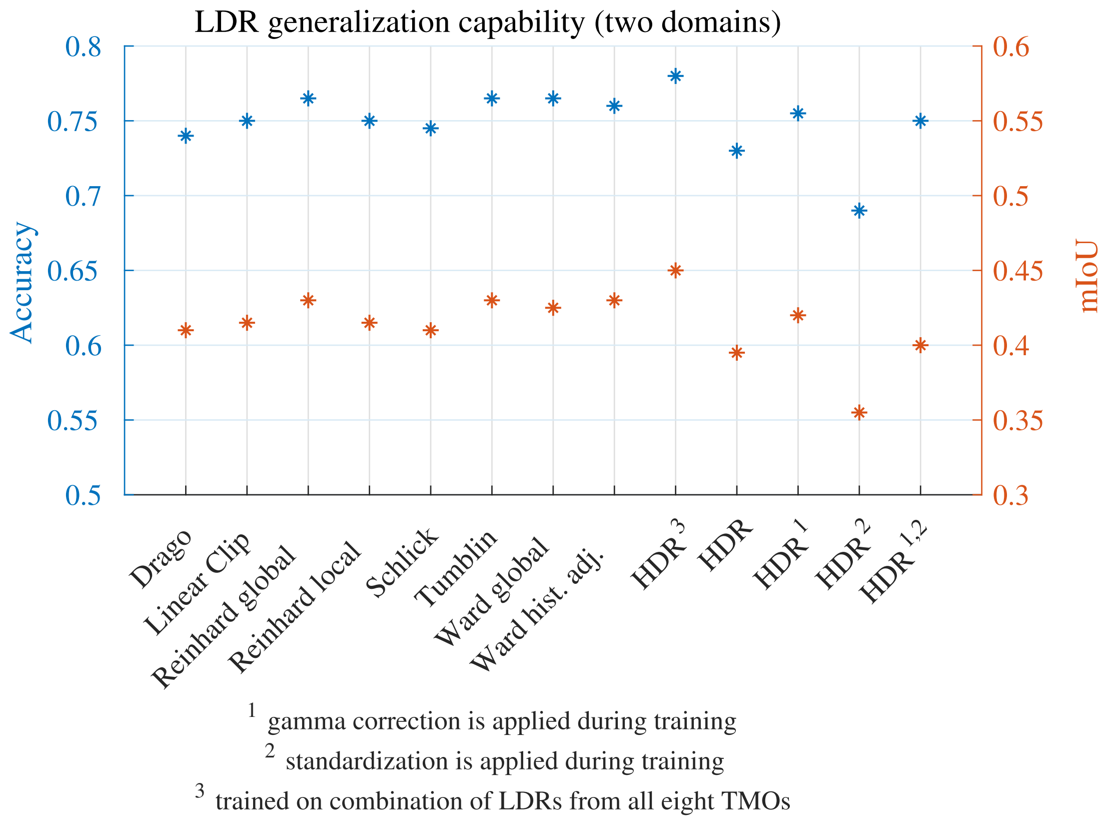


Figure 5.23 Evaluation of the semantic segmentation networks listed on the x-axis by the LDR generalization capability.

HDR generalization capability

The HDR generalization capability determines the semantic segmentation performance for real-world HDR images. The results are shown in figure 5.24. The performances are very different to the ones before. The best performing networks are clearly the HDR[3], HDR[1] and HDR[1,2] networks, followed by the HDR network. Once again the gamma correction improves the semantic segmentation performance for HDR images. The HDR[3] network was intended to approximate the high dynamic range but consists of only LDR images. Nevertheless, it outperforms the HDR network trained on raw pixel values. The eight LDR networks clearly perform worse on the real-world HDR images than the other networks. Worst performance is shown by HDR[2]. The best tone mapping operator is for this scenario the Schlick operator, which has shown a rather low performance for the LDR evaluations.

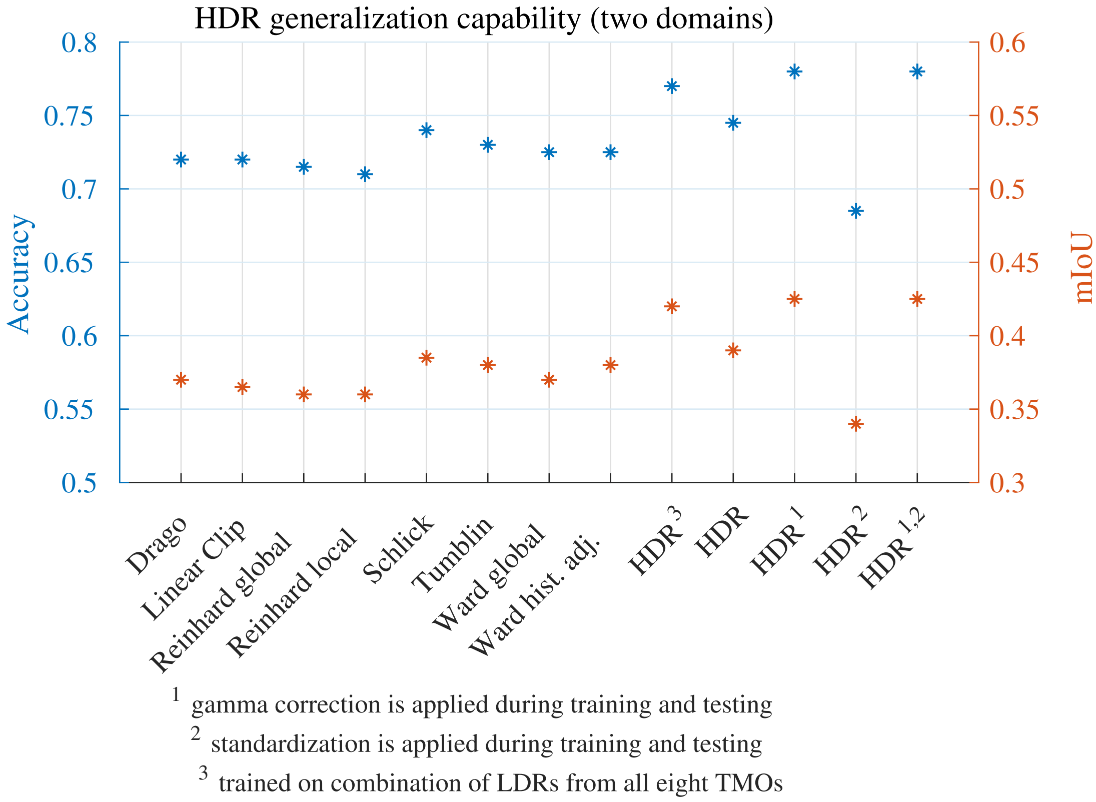


Figure 5.24 Evaluation of the semantic segmentation networks listed on the x-axis by the HDR generalization capability.

Best performing semantic segmentation networks

The highest overall semantic segmentation performance might be measured by the accuracy, as two out of the eleven classes for evaluation are not present in the BIT test set. Therefore, the best semantic segmentation performance is shown by the Reinhard global network with 85.5% accuracy and 42% mIoU for the closed system performance.

The highest performance for LDR real-world images is shown by the HDR[3] network with 78% and 45% mIoU accuracy.

The best performance for HDR real-world images is shown by the HDR[1] network with 78% and 42.5% mIoU accuracy.

5.3 Evaluation of tone mapping ranking method

Pearson correlations are calculated to evaluate the proposed tone mapping ranking method from subsection 3.1.3. Figure 5.25 shows the relationship between semantic segmentation performance, as measured by the accuracy and mIoU, and final ranking score of the eight tone mapping operators. The marker of the Drago operator is shown in red. A moderate to high correlation can be seen, with a value between 0.58 and 0.79, for the ranking method with the closed system performance and the LDR generalization capability. There is no correlation between the ranking

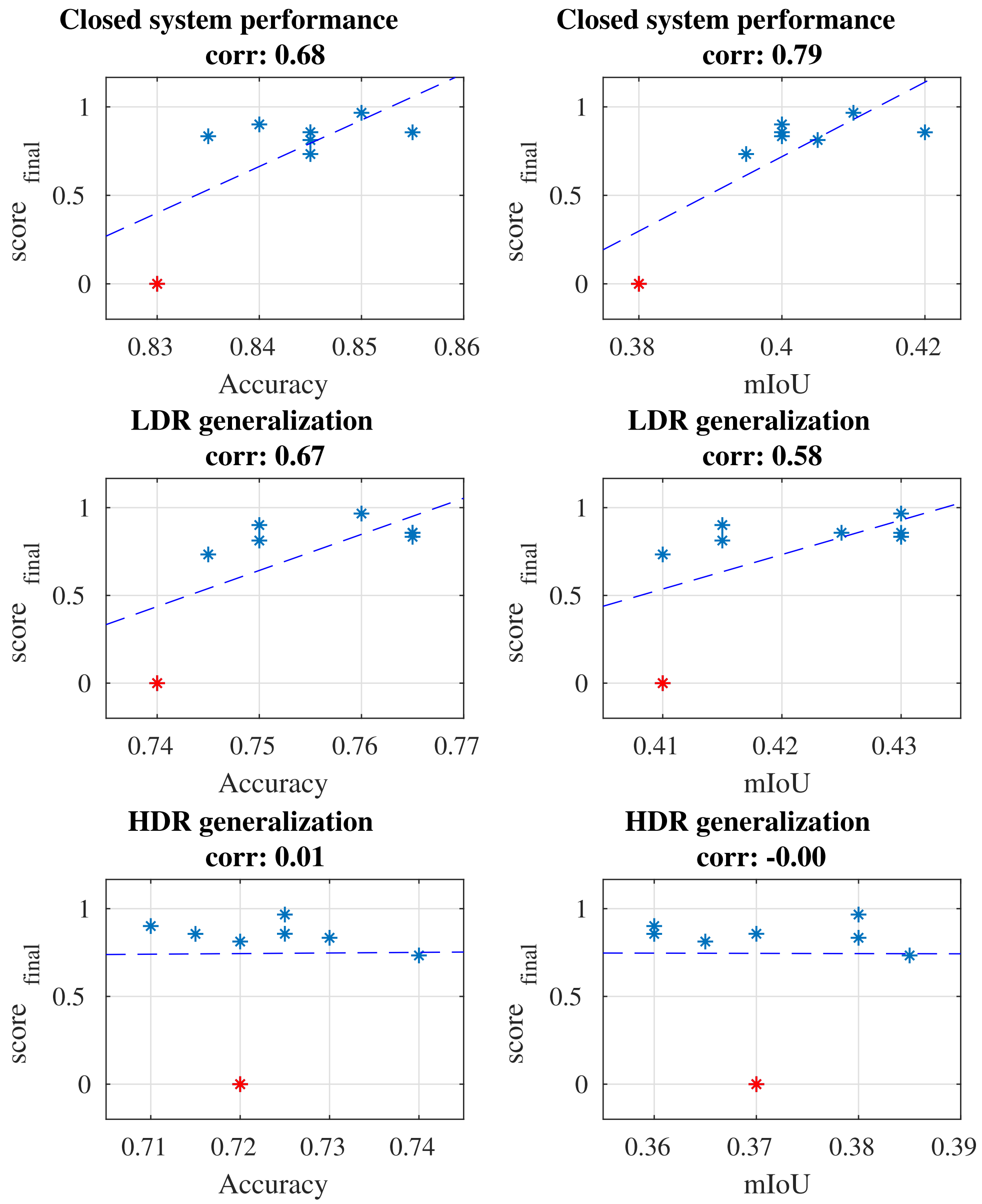


Figure 5.25 The red marker classifies the Drago operator, while the blue markers show the values for the other tone mapping operators.

method and the HDR generalization capability.

The Pearson correlations between final ranking scores and semantic segmentation performance are determined again for the individual components of the tone mapping ranking method. In addition, it is investigated if the order of the ranking scores performs better than the actual values. Furthermore, the correlations are determined again by computing the ranking scores with exclusion of the Drago operator. The results are shown in table 5.5.

	Closed system performance (Accuracy)	Closed system performance (mIoU)	LDR generalization (Accuracy)	LDR generalization (mIoU)	HDR generalization (Accuracy)	HDR generalization (mIoU)
With Drago operator						
a) OIQ scores	0.64	0.75	0.66	0.59	0.06	0.09
b) ID scores	**0.69**	0.75	0.57	0.44	0.10	0.02
c) TMS scores	0.67	**0.85**	**0.77**	**0.68**	-0.24	**-0.26**
d) final scores	0.68	0.79	0.67	0.58	0.01	0.00
e) final, only ranking	0.49	0.60	0.61	0.63	**-0.36**	-0.18
Without Drago operator						
a) OIQ scores	0.12	0.32	0.35	**0.53**	-0.41	-0.06
b) ID scores	**0.43**	0.17	-0.50	-0.41	-0.45	-0.39
c) TMS scores	0.31	**0.63**	**0.51**	0.44	**-0.77**	**-0.79**
d) final scores	0.35	0.47	0.17	0.32	-0.68	-0.41
e) final, only ranking	0.42	0.55	0.27	0.40	-0.70	-0.45

Table 5.5 Validation of the tone mapping ranking method through correlation with semantic segmentation performance of several experiments. a) to c) shows an ablation study, where only single components of the method are used, while d) shows the correlation coefficients between the semantic performance and the final scores of the method. e) shows the correlation coefficients when only considering the order of the final scores but not the actual values.

The correlation between final ranking scores and semantic segmentation performances is for any evaluation metric lower than the correlation with just components of it. For the HDR generalization there is no correlation at all when considering the Drago operator.

It can be seen for the HDR generalization, that there is usually a negative correlation for the different semantic segmentation evaluations. This means that the actual scores should have been inverted to achieve better approximations for the HDR generalization measure. For most of the semantic segmentation evaluation methods the TMS measure seems to be the best metric to approximate the semantic segmentation performance amongst the investigated metrics. With a correlation of up to 85% for the closed system performance, as measured by the mIoU, there can be seen a high correlation between expected and actual semantic segmentation performance.

5.4 Deep domain adaptation

Following the experimental results of the domain adaptation are presented. In subsection 5.4.1 the domain gap between synthetic source and real-world target domain is visualized and in subsection 5.4.2 the semantic segmentation performances after the adaptation are presented.

5.4.1 Domain gap between synthetic and real-world images

Figure 5.26 shows the performance of the semantic segmentation networks on the synthetic source domain (red) and the real-world LDR target domain (blue). These correspond to the closed system performance (red) and the LDR generalization capability (blue). The filled area between the markers shows the performance drop. This indicates the domain gap reasoned by the different domains of the test sets. It can be seen that the domain gap causes a performance drop of about 10% accuracy between synthetic and real-world LDR images. This is similar between synthetic LDR and HDR images, except for the HDR^{1,2} network.

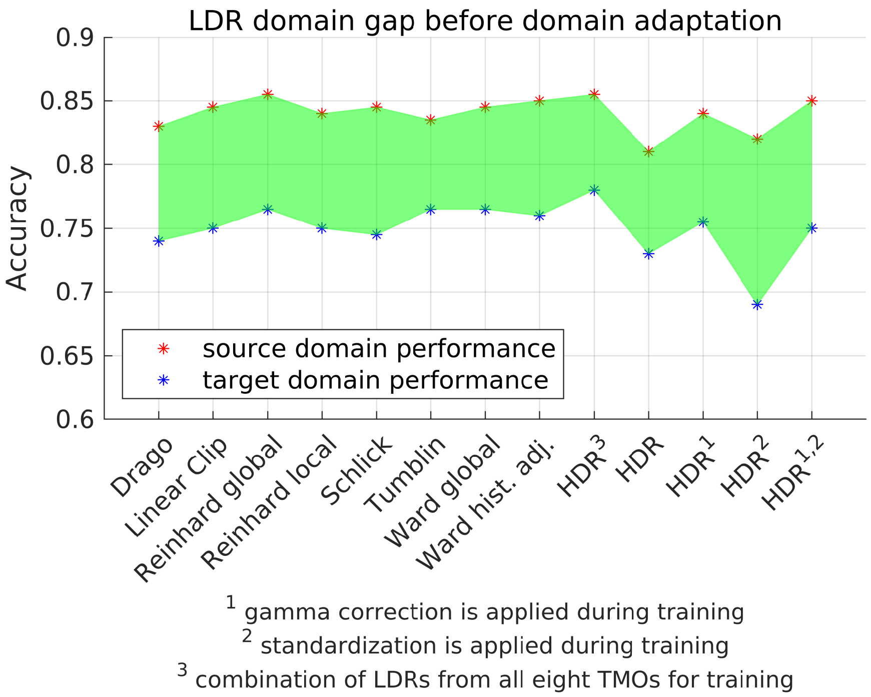


Figure 5.26 Domain gap between synthetic source and real-world LDR target domain performance.

Evaluating the networks on real-world HDR images shows a different result. Figure 5.27 presents the domain gap between synthetic source domain performance (red) and real-world HDR target domain performance (blue). The domain gap of the eight LDR networks from the different TMOs is much larger than before. For the Reinhard global operator the performance drop is 14%. For the HDR and HDR^{1} network, the performance drop is only 4-5%.

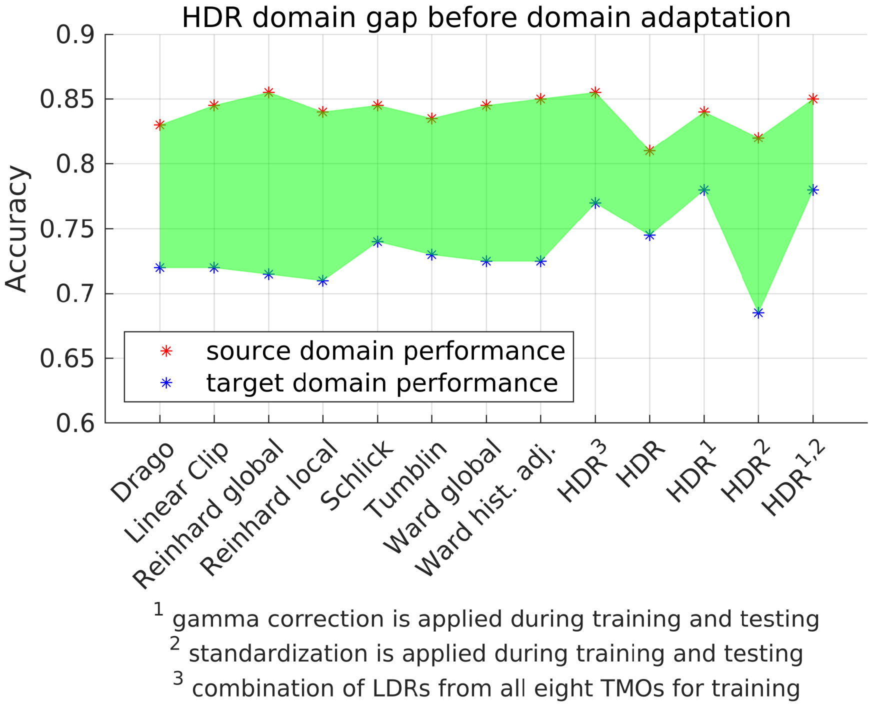


Figure 5.27 Domain gap between synthetic source and real-world HDR target domain performance.

5.4.2 Domain adaptation for semantic segmentation

To reduce the domain gap and improve the semantic segmentation performance the LDR dataset from the Linear clip operator and the HDR^1 dataset is image-to-image translated into the real-world domain. This results in two domain adapted datasets for the real-world LDR domain and two for the real-world HDR domain.

The networks trained on the real-world LDR domain adapted datasets are evaluated by the LDR generalization capability. Again, the semantic segmentation performances from two individual trainings and evaluations with the same hyperparameter settings are averaged to achieve a more robust estimation of the performance. The individual results are given in table 7.4 in the appendix. The final performances are shown in figure 5.28. The domain gap could be reduced clearly by the domain adaptation, as the performance has improved by 3-6% on the real-world LDR target domain. The image-to-image translation from synthetic LDR to real-world LDR improved the performance more than the one from synthetic HDR to real-world LDR. The remaining performance drop is 3.5% for the Linear clip operator.

Similarly, the networks trained on the real-world HDR domain adapted datasets are evaluated by the HDR generalization capability. Figure 5.28 shows the semantic segmentation performances. The domain adaptation from synthetic LDR to real-world HDR has improved the performance by 7%. In contrast, the domain adaptation from synthetic HDR to real-world HDR could not show a clear improvement. The semantic segmentation performance is increased by only 0.5%.

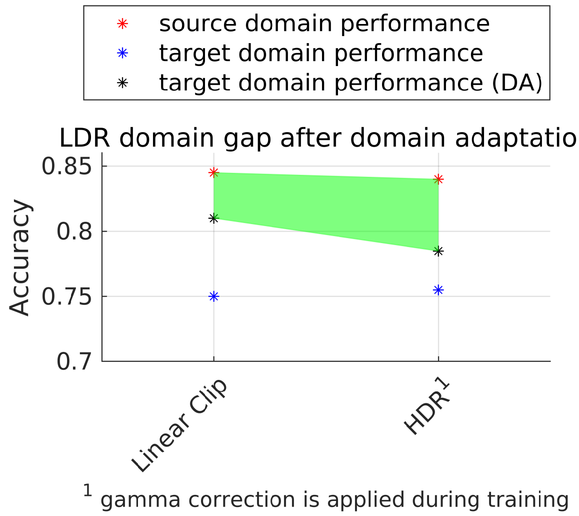


Figure 5.28 Semantic segmentation performance on real-world LDR images before (blue) and after (black) domain adaptation (DA).

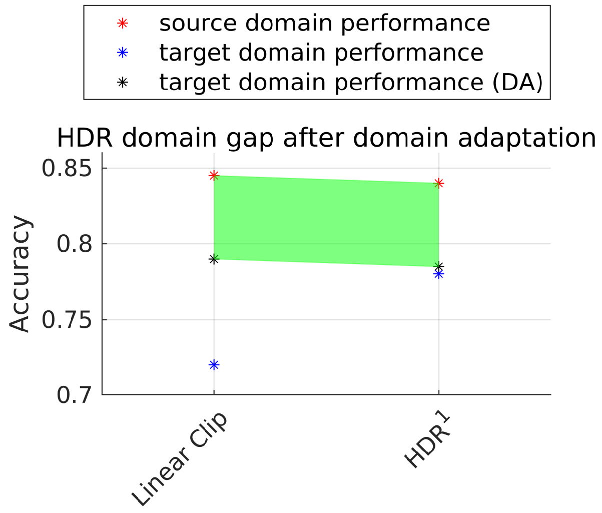


Figure 5.29 Semantic segmentation performance on real-world HDR images before (blue) and after (black) domain adaptation (DA).

6 Discussion and Conclusions

In this thesis it was analysed how well synthetic high dynamic range images are suited for computer vision tasks in automotive applications. In specific, the semantic segmentation performance of neural networks trained on synthetic HDR images was investigated. For this purpose three approaches were presented in chapter 3. Goal of the first approach was to find out if there is an advantage of high dynamic range images by training semantic segmentation networks on tone mapped LDR datasets. This method is discussed in section 6.1. The second approach aimed to evaluate the semantic segmentation networks and is discussed in section 6.2. The third approach investigated if the performance on real-world scenes could be improved, as the training was carried out on synthetic images. This approach is discussed in section 6.3.

6.1 High dynamic range imaging

The first approach of this thesis was to apply several tone mapping operators to the HDR dataset and train a network on every resulting LDR dataset. Additionally, the proposed tone mapping ranking method was carried out to approximate the semantic segmentation performances.

Tone mapping of HDR images
For experiments, the HDR dataset was tone mapped with one single parameter setting for each TMO, which was determined by a non-linear optimization with the TMQI evaluation metric. Intention of the first approach was not to investigate the best absolute possible semantic segmentation performance of a tone mapping operator. Instead, it was aimed to investigate which tone mapping operator performs best with one single parameter setting. This should find out how well they can be applied to unseen images. This might be useful for automotive applications, where the non-linear optimization may take too long for every image. However, the parameter optimization may also be performed for every image. This would very likely change the semantic segmentation performances for some of the operators and may be a possibility to further improve the results.

Tone mapping ranking method
The scores of the proposed tone mapping ranking method turned out to show only a moderate correlation with the semantic segmentation performances. The highest correlation occurred between the scores of the *tone mapping sensitivity* evaluation metric and the semantic segmentation results. As the training and evaluation of a neural network is computationally very expensive and requires costly GPUs, it may be nevertheless a good alternative to estimate the semantic segmentation performance for tone mapped LDR datasets. One limitation of the ranking method evaluation is, that the computed Pearson correlations measure only linear relationships. Non-linear relations are thereby not considered.

Limitations and problems of HDR imaging

While high dynamic range imaging allows to access the full luminance range of a scene, it brings also some disadvantages, which have been shown during experiments:

- *Generation*: Maybe one of the most influencing factors, why mainly low dynamic range images are used for computer vision applications, is the high effort to capture HDR images from real scenes. While professional equipment is very costly, generating them with usual cameras brings several disadvantages, as explained in subsection 2.1.2.
- *Usability*: Working with high dynamic images is often non-trivial, which is another disadvantage of HDR images. It requires special software or hardware in order to display the images. Conventional PC systems are designed to work with low dynamic range images, as the standard image viewers and displays are not capable of viewing HDR content. Additionally, the amount of available HDR images on the internet is much less, which makes it way more difficult to collect large-scale datasets like ImageNet.
- *Representation*: Another very challenging factor is the representation of the HDR images. While standard low dynamic range pixel values are typically defined by integers between 0 and 255 for each color channel, there are numerous ways to store HDR images. They can be similarly discretized as integers, only by a larger amount of grey values. But they can also be stored as float variables, with any kind of representation. It can therefore be very difficult to compare high dynamic range images from different sources, as it may not be clear, what the pixel values in the HDR image should represent.
- *Accurateness*: While LDR images are rather designed to give an impression of the content of a scene, HDR images are intended to be accurate to the physical properties of a scene. Anyway, the accuracy is strongly limited to the precision during their generation, whether using real cameras or computer graphics. Accessing the accuracy of HDR images is difficult and for most situations there is no statement, which makes comparing HDR images from different sources even more challenging.

6.2 Semantic segmentation performance of synthetic HDR images

In section 3.2 an approach was presented to determine the performance of semantic segmentation networks, which was divided into three subtasks. The *closed system performance* investigated the near-optimal semantic segmentation performance. The *LDR generalization capability* determined the performance for real-world images from a regular LDR camera sensor, while the *HDR generalization capability* determined the real-world performance for an HDR camera sensor. These evaluation methods were used to determine the semantic segmentation performance for the different tone mapped datasets and the HDR dataset.

For the closed system performance it was assumed, that the HDR dataset would perform better than the LDR datasets. The idea was, that neural networks should benefit from the additional information of the high dynamic range images. Nevertheless, the LDR images from some tone mapping operators turned out to perform better. This may be reasoned to the pretraining of all networks on LDR images. Therefore, features were already learned from LDR images, which might not be well transferable to the features of HDR images. Performing a gamma correction, during training and testing, improved the results for HDR images. This showed, that the features are better suited for LDR images, as the gamma correction makes the representation of HDR images more similar to the one of LDR images. For a fair comparison, the experiments might be repeated with untrained networks. However, this requires a much larger dataset for training, which was not available for this thesis.

Furthermore, it should be mentioned that the results may change for different configurations. The experiments were carried out with one specific network architecture. The performances would be very likely different for other architectures. Also the HDR dataset, which is used to generate the LDR datasets, plays an important role for the results. Other HDR datasets may perform rather different for the semantic segmentation evaluations. Last, the hyperparameter setting for the networks was determined on the real-world Cityscapes dataset. This setting was used to train the networks on the synthetic HDR and LDR datasets. It was not investigated, if there were better hyperparameter settings for training on those datasets. These factors limited the investigations, which aimed to determine the maximum possible semantic segmentation performance of HDR and LDR images.

6.3 Domain adaptation of synthetic HDR images for semantic segmentation

The domain adaptation improved the results for all three experiments with LDR images. Only the domain adaptation between synthetic HDR and real-world HDR images showed no improvement. However, the performance was already much closer without the adaptation. Based on this results, it is assumed, that the domain specific appearance – often considered as the style of an image [Gatys et al., 2016] – is mainly reasoned due to LDR specific operations like tone mapping. Therefore, no improvement between HDR images could be achieved with the image-to-image translation network, which aimed to modify the appearance of source images to the target domain. This is only an assumption and might be further investigated with additional domain adaptation methods.

7 Conclusion

7.1 Summary

This work hypothesized that high dynamic range images are beneficial for the performance of semantic segmentation networks in autonomous driving vehicles. For validating this hypothesis three approaches were presented.

First, multiple LDR datasets were generated from a synthetic HDR dataset with eight different tone mapping operators. A ranking method was proposed to estimate the semantic segmentation performance for each tone mapping operator. This method consists of three different evaluation metrics, the *overall image quality*, the *image dynamics* and the *tone mapping sensitivity*. The *Drago* [Drago et al., 2003] operator performed worst for all evaluations, while the seven other TMOs performed similarly well.
An evaluation of the proposed ranking method showed, that the *tone mapping sensitivity* metric seemed to be the best approach, out of the investigated methods, to estimate the semantic segmentation performance.

To determine the performance of semantic segmentation networks, three evaluation methods were proposed to consider the following aspects. The *closed system performance* allowed to determine the near-optimal performance of a segmentation network and was determined on synthetic images. The *LDR generalization capability* and the *HDR generalization capability* were used to investigate the effectiveness of a network on real-world images.
The best overall semantic segmentation performance is shown by the *Reinhard global* [Reinhard et al., 2002] operator with 85.5% accuracy for the closed system performance. With 78% the highest accuracy for LDR real-world images is shown by a network trained on a combination of synthetic LDR images from the eight different TMOs. The best performance for HDR real-world images is shown by a network trained on an synthetic HDR images with again 78% accuracy. The *Drago* [Drago et al., 2003] operator was the worst performing TMO for all three evaluations. This demonstrates, that it is essential to use HDR images for achieving the highest performance on real-world scenes. Both, training directly on HDR images and training on a combination of LDR images from different TMOs, requires high dynamic range images. Furthermore, it was shown that a gamma correction improved the semantic segmentation performance of networks trained on HDR images when using a network that is pretrained on LDR images.

Additionally, experiments demonstrated that neural networks can gain a comprehensive understanding on real-world target domain images from synthetic source domain images. The semantic segmentation performance could even be improved with the third approach of this thesis, which was demonstrated with a domain adaptation. The accuracy on real-world images was increased from 75% to 81% for LDR images. No improvement could be achieved from synthetic HDR to real-world HDR. Nevertheless, the performance drop from synthetic HDR to real-world HDR turned out to be much smaller than the one from synthetic LDR to real-world LDR. This reveals another benefit of HDR images, as they cause a much smaller performance drop for images from different domains.

7.2 Outlook

In addition to the investigations carried out in this thesis, there are several possibilities to extend the research.

One of them is to discover the actual difference between the performance of computer vision systems trained on LDR and HDR images. As all networks in this thesis are pre-trained on LDR images, this limitation should be eliminated for the next experiments. This could be solved by creating a large enough HDR dataset with inverse tone mapping or high dynamic range imaging techniques. Another possibility would be to train the model from scratch. However, this may affect the absolute performance, if there are not enough training images available. Furthermore, the performance on critical situations should be investigated, where the dynamic range of LDR images is clearly not high enough. For example, when a self-driving car is leaving a tunnel, which can cause a very large scene contrast.

Additionally, it may be further investigated how synthetic images can improve the safety of automotive applications like autonomous driving systems. This could be achieved by customizing existing domain adaptation methods or by generating more realistic synthetic images. The latter may be realized through improved image-to-image translation methods. These should allow to generate synthetic images with a very high degree of realism from unlabeled real-world images by deep learning approaches.

Extended research possibilities to demonstrate the effectiveness of high dynamic range images may be shown with other computer vision tasks where the limited dynamic range of LDR images could affect the performance. This might also be the case for object detection tasks where objects in very contrasty regions may not be detected by an artificial intelligence system.

References

Astey Highways (2013) CH / A2 Driveway to the Gotthard Tunnel (Göschenen Portal). https://www.youtube.com/watch?v=Qt-T9TAwrLw. Online; accessed on 09 August 2019

Banterle F, Artusi A, Debattista K, Chalmers A (2017) Advanced high dynamic range imaging. 2nd ed. Natick, MA, USA: AK Peters (CRC Press). ISBN: 9781498706940

BIT (2019) Ground truth training data for autonomous driving. Dataset, BIT Technology Solutions GmbH, Gewerbering 3, 83539 Pfaffing OT Forsting, Germany

Bolte J-A, Bär A, Lipinski D, Fingscheidt T (2019) Towards corner case detection for autonomous driving. In: *Proceedings of the IEEE Intelligent Vehicles Symposium*, preprint

Bosch A, Zisserman A, Muñoz X (2008) Scene classification using a hybrid generative/discriminative approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(4): 712–727

Brostow GJ, Fauqueur J, Cipolla R (2009) Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2): 88–97

Byrd RH, Gilbert JC, Nocedal J (2000) A trust region method based on interior point techniques for nonlinear programming. *Mathematical programming*, 89(1): 149–185

Čadík M, Wimmer M, Neumann L, Artusi A (2006) Image attributes and quality for evaluation of tone mapping operators. In: *Proceedings of Pacific Graphics*, Taipei, Taiwan: National Taiwan University Press, 35–44

Chang W-L, Wang H-P, Peng W-H, Chiu W-C (2019) All about structure: Adapting structural information across domains for boosting semantic segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1900–1909

Chen L-C, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2018a) DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. In: *Proceedings of the IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4): 834–848

Chen Y-H, Chen W-Y, Chen Y-T, Tsai B-C, Frank Wang Y-C, Sun M (2017) No more discrimination: Cross city adaptation of road scene segmenters. In: *Proceedings of the IEEE International Conference on Computer Vision*, 1992–2001

Chen Y, Li W, Sakaridis C, Dai D, Van Gool L (2018b) Domain adaptive faster R-CNN for object detection in the wild. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3339–3348

Chen Y, Li W, Van Gool L (2018c) ROAD: Reality oriented adaptation for semantic segmentation of urban scenes. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7892–7901

Chen Y, Li W, Chen X, Van Gool L (2019) Learning semantic segmentation from synthetic data: A geometrically guided input-output adaptation approach. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1841–1850

Cityscapes Benchmark Suite (2019) Pixel-level semantic labeling task. `https://www.cityscapes-dataset.com/benchmarks/#scene-labeling-task`. Online; accessed on 19 June 2019

Cordts M, Omran M, Ramos S, Rehfeld T, Enzweiler M, Benenson R, Franke U, Roth S, Schiele B (2016) The Cityscapes dataset for semantic urban scene understanding. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3213–3223

Csurka G (2017) Domain adaptation in computer vision applications. In: G Csurka (ed) Advances in Computer Vision and Pattern Recognition. Cham, Germany: Springer. Chap. A Comprehensive Survey on Domain Adaptation for Visual Applications, 1–35. ISBN: 978-3-319-58347-1

Dai J, He K, Sun J (2016) Instance-aware semantic segmentation via multi-task network cascades. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3150–3158

De Neve S, Goossens B, Philips W, et al. (2009) An improved HDR image synthesis algorithm. In: *Proceedings of the IEEE International Conference on Image Processing*, 1545–1548

Debevec PE, Malik J (2008) Recovering high dynamic range radiance maps from photographs. In: *ACM SIGGRAPH 2008 classes*, 31

Devlin K (2002) A review of tone reproduction techniques. *Department of Computer Science, University of Bristol, Technical Report CSTR-02-005*

Drago F, Myszkowski K, Annen T, Chiba N (2003) Adaptive logarithmic mapping for displaying high contrast scenes. In: *Computer Graphics Forum*, 22(3): 419–426

Eilertsen G, Unger J, Mantiuk RK (2016) Evaluation of tone mapping operators for HDR video. In: *High Dynamic Range Video*, Elsevier, 185–207

Endo Y, Kanamori Y, Mitani J (2017) Deep reverse tone mapping. *ACM Transactions on Graphics*, 36(6): 177–1

Gaidon A, Wang Q, Cabon Y, Vig E (2016) Virtual worlds as proxy for multi-object tracking analysis. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4340–4349

Garcia-Garcia A, Orts-Escolano S, Oprea S, Villena-Martinez V, Garcia-Rodriguez J (2018) A review on deep learning techniques applied to semantic segmentation. *International Journal of Multimedia Information Retrieval*, 7(2): 87–93

Gatys LA, Ecker AS, Bethge M (2016) Image style transfer using convolutional neural networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2414–2423

Geiger A, Lenz P, Stiller C, Urtasun R (2013) Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11): 1231–1237

Gong R, Li W, Chen Y, Van Gool L (2019) DLOW: Domain flow for adaptation and generalization. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2477–2486

Granados M, Ajdin B, Wand M, Theobalt C, Seidel H-P, Lensch HP (2010) Optimal HDR reconstruction with linear digital cameras. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 215–222

Grossberg MD, Nayar SK (2003) Determining the camera response from images: What is knowable? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(11): 1455–1467

He K, Zhang X, Ren S, Sun J (2015) Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: *Proceedings of the IEEE International Conference on Computer Vision*, 1026–1034

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

Hoffman J, Wang D, Yu F, Darrell T (2016) FCNs in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv:1612.02649*

Hoffman J, Tzeng E, Park T, Zhu J-Y, Isola P, Saenko K, Efros AA, Darrell T (2018) CyCADA: Cycle-consistent adversarial domain adaptation. In: *Proceedings of the 35th International Conference on Machine Learning*, 80: 1989–1998

Hong W, Wang Z, Yang M, Yuan J (2018) Conditional generative adversarial network for structured domain adaptation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1335–1344

Huang X, Cheng X, Geng Q, Cao B, Zhou D, Wang P, Lin Y, Yang R (2018) The ApolloScape Dataset for Autonomous Driving. *arXiv:1803.06184*

Isola P, Zhu J-Y, Zhou T, Efros AA (2017) Image-to-image translation with conditional adversarial networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1125–1134

ITU (2015) Recommendation ITU-R BT.709-6: Parameter values for the HDTV standards for production and international programme exchange. `https://www.itu.int/dms_pubrec/itu-r/rec/bt/R-REC-BT.709-6-201506-I!!PDF-E.pdf`. Online; accessed on 27 September 2019

Ivakhnenko AG (1971) Polynomial theory of complex systems. *IEEE Transactions on Systems, Man and Cybernetics*, (4): 364–378

Jain A, Zamir AR, Savarese S, Saxena A (2016) Structural-RNN: Deep learning on spatio-temporal graphs. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5308–5317

Kang B, Nguyen TQ (2019) Random Forest With Learned Representations for Semantic Segmentation. *IEEE Transactions on Image Processing*, 28(7): 3542–3555

Keskar NS, Socher R (2017) Improving generalization performance by switching from adam to sgd. *arXiv:1712.07628*

Kiefer J, Wolfowitz J, et al. (1952) Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, 23(3): 462–466

Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. *arXiv:1412.6980*

Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*, 1097–1105

Larson GW, Rushmeier H, Piatko C (1997) A visibility matching tone reproduction operator for high dynamic range scenes. *IEEE Transactions on Visualization and Computer Graphics*, 3(4): 291–306

Ledda P, Chalmers A, Troscianko T, Seetzen H (2005) Evaluation of tone mapping operators using a high dynamic range display. In: *Proceedings of the ACM Transactions on Graphics*, 24(3): 640–648

Lee K-H, Ros G, Li J, Gaidon A (2019) SPIGAN: Privileged adversarial learning from simulation. In: *Proceedings of the International Conference on Learning Representations*, preprint

Li J, Wang JZ (2003) Automatic linguistic indexing of pictures by a statistical modeling approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9): 1075–1088

Li L-J, Socher R, Fei-Fei L (2009) Towards total scene understanding: Classification, annotation and segmentation in an automatic framework. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2036–2043

Li X, Zhao H, Han L, Tong Y, Yang K (2019) GFF: Gated fully fusion for semantic segmentation. *arXiv:1904.01803*

Li Y, Qi H, Dai J, Ji X, Wei Y (2017) Fully convolutional instance-aware semantic segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2359–2367

Lin G, Milan A, Shen C, Reid ID (2016) RefineNet: Multi-path refinement networks for high-resolution semantic segmentation. *arXiv:1611.06612*

Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft coco: Common objects in context. In: *Proceedings of the European Conference on Computer Vision*, Springer, 740–755

Liu Z, Li X, Luo P, Loy C-C, Tang X (2015) Semantic image segmentation via deep parsing network. In: *Proceedings of the IEEE International Conference on Computer Vision*, 1377–1385

Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3431–3440

Luo Y, Liu P, Guan T, Yu J, Yang Y (2019a) Significance-aware information bottleneck for domain adaptive semantic segmentation. *arXiv:1904.00876*

Luo Y, Zheng L, Guan T, Yu J, Yang Y (2019b) Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2507–2516

Mantiuk RK, Myszkowski K, Seidel H-P (2015) High dynamic range imaging. *Wiley Encyclopedia of Electricaland Electronics Engineering*

Mantiuk R (2013) Tone mapping. `https://www.cl.cam.ac.uk/~rkm38/pdfs/tone_mapping.pdf`. Online; accessed on 01 April 2019

Mittal A, Moorthy AK, Bovik AC (2011) Blind/referenceless image spatial quality evaluator. In: *Proceedings of the Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers*, IEEE, 723–727

Mordvintsec A, Olah C, Tyka M (2015) Inceptionism: Going deeper into neural networks. `https://ai.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html`. Online; accessed on 29 July 2019

Naverlabs (2019) Virtual KITTI dataset. `https://europe.naverlabs.com/research/computer-vision/proxy-virtual-worlds/`. Online; accessed on 30 July 2019

Neuhold G, Ollmann T, Rota Bulo S, Kontschieder P (2017) The mapillary vistas dataset for semantic understanding of street scenes. In: *Proceedings of the IEEE International Conference on Computer Vision*, 4990–4999

Ng T-T, Chang S-F, Tsui M-P (2007) Using geometry invariants for camera response function estimation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1–8

Patel VM, Gopalan R, Li R, Chellappa R (2015) Visual domain adaptation: A survey of recent advances. *IEEE Signal Processing Magazine*, 32(3): 53–69

Pharr M, Jakob W, Humphreys G (2016) Physically based rendering: From theory to implementation. 3rd ed. Cambridge, MA, USA: Morgan Kaufmann. ISBN: 978-0-12-800645-0

PIL (2019) Image module. `https://pillow.readthedocs.io/en/3.1.x/reference/Image.html`. Online; accessed on 29 September 2019

Pinggera P, Ramos S, Gehrig S, Franke U, Rother C, Mester R (2016) Lost and found: Detecting small road hazards for self-driving vehicles. *arXiv:1609.04653*

Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: Unified, real-time object detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 779–788

Reinhard E, Stark M, Shirley P, Ferwerda J (2002) Photographic tone reproduction for digital images. In: *Proceedings of the ACM Transactions on Graphics*, 21(3): 267–276

Reinhard E, Heidrich W, Debevec P, Pattanaik S, Ward G, Myszkowski K (2010) High dynamic range imaging: Acquisition, display, and image-based lighting. 2nd ed. Burlington, MA, USA: Morgan Kaufmann. ISBN: 978-0-12-374914-7

Rempel AG, Trentacoste M, Seetzen H, Young HD, Heidrich W, Whitehead L, Ward G (2007) Ldr2hdr: On-the-fly reverse tone mapping of legacy video and photographs. In: *ACM transactions on graphics*, *26*. (3). ACM, 39

Richter SR, Vineet V, Roth S, Koltun V (2016) Playing for data: Ground truth from computer games. In: *Proceedings of the European Conference on Computer Vision*, Springer, 102–118

Ronneberger O, Fischer P, Brox T (2015) U-net: Convolutional networks for biomedical image segmentation. In: *International Conference on Medical Image Computing and Computer Assisted Intervention*, Springer, 234–241

Ros G, Sellart L, Materzynska J, Vazquez D, Lopez AM (2016) The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3234–3243

Sankaranarayanan S, Balaji Y, Jain A, Lim SN, Chellappa R (2017) Unsupervised domain adaptation for semantic segmentation with GANs. *arXiv:1711.06969*

Schlick C (1995) Quantization techniques for visualization of high dynamic range pictures. In: *Photorealistic Rendering Techniques*, Springer, 7–20

Schmidhuber J (2015) Deep learning in neural networks: An overview. *Neural Networks*, 61: 85–117

Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*

Smith LN (2017) Cyclical learning rates for training neural networks. In: *Proceedingss of the IEEE Conference on Applications of Computer Vision*, 464–472

Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1–9

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

Tumblin J, Rushmeier H (1993) Tone reproduction for realistic images. *IEEE Computer Graphics and Applications*, 13(6): 42–48

Tumblin J, Hodgins JK, Guenter BK (1999) Two methods for display of high contrast images. *ACM Transactions on Graphics*, 18(1): 56–94

Vu T-H, Jain H, Bucher M, Cord M, Pérez P (2019a) ADVENT: Adversarial entropy minimization for domain adaptation in semantic segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2517–2526

Vu T-H, Jain H, Bucher M, Cord M, Pérez P (2019b) DADA: Depth-aware domain adaptation in semantic segmentation. *arXiv:1904.01886*

Wang M, Deng W (2018) Deep visual domain adaptation: A survey. *Neurocomputing*, 312: 135–153

Wang Z, Bovik AC, Sheikh HR, Simoncelli EP, et al. (2004) Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612

Ward G (1994) A contrast-based scalefactor for luminance display. *Graphics Gems IV*: 415–421

WHO (2019) World health organization: Road traffic injuries. `https://www.who.int/violence_injury_prevention/road_traffic/en/`. Online; accessed on 07 August 2019

Wrenninge M, Unger J (2018) Synscapes: A photorealistic synthetic dataset for street scene parsing. *arXiv:1810.08705*

Wu Z, Han X, Lin Y-L, Gokhan Uzunbas M, Goldstein T, Nam Lim S, Davis LS (2018) DCAN: Dual channel-wise alignment networks for unsupervised scene adaptation. In: *Proceedings of the European Conference on Computer Vision*, 518–534

Yeganeh H, Wang Z (2012) Objective quality assessment of tone-mapped images. *IEEE Transactions on Image Processing*, 22(2): 657–667

Yosinski J, Clune J, Bengio Y, Lipson H (2014) How transferable are features in deep neural networks? In: *Advances in Neural Information Processing Systems*, 3320–3328

Yu F, Xian W, Chen Y, Liu F, Liao M, Madhavan V, Darrell T (2018) BDD100K: A diverse driving video database with scalable annotation tooling. *arXiv:1805.04687*

Zhang C, Bengio S, Hardt M, Recht B, Vinyals O (2016) Understanding deep learning requires rethinking generalization. *arXiv:1611.03530*

Zhang Y, David P, Gong B (2017) Curriculum domain adaptation for semantic segmentation of urban scenes. In: *Proceedings of the IEEE International Conference on Computer Vision*, 2020–2030

Zhu X, Zhou H, Yang C, Shi J, Lin D (2018) Penalizing top performers: Conservative loss for semantic segmentation adaptation. In: *Proceedings of the European Conference on Computer Vision*, 568–583

Zhu Y, Sapra K, Reda FA, Shih KJ, Newsam S, Tao A, Catanzaro B (2019) Improving semantic segmentation via video propagation and label relaxation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8856–8865

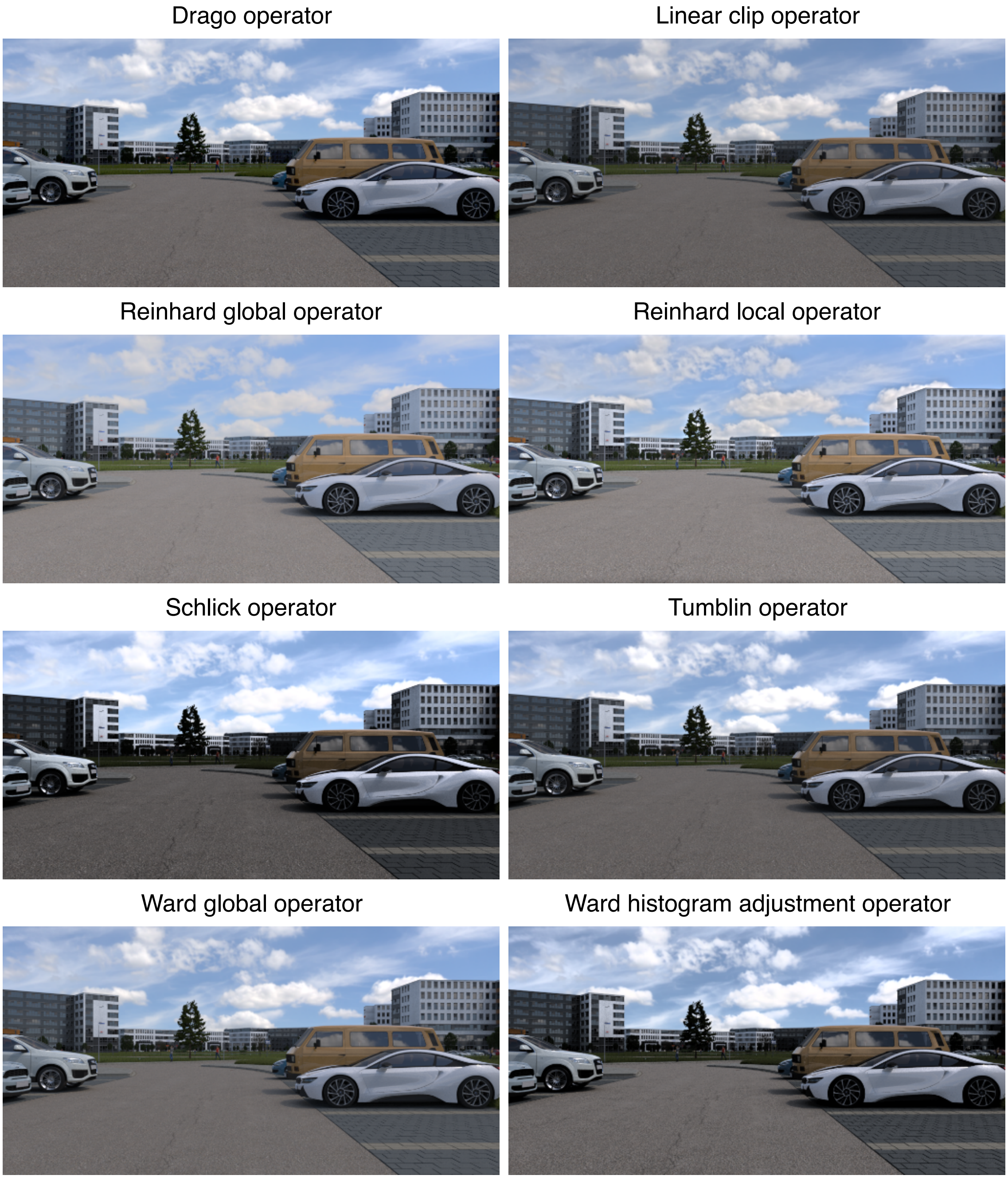
Zhuang Y, Yang F, Tao L, Ma C, Zhang Z, Li Y, Jia H, Xie X, Gao W (2018) Dense relation network: Learning consistent and context-aware representation for semantic image segmentation. In: *Proceedings of the IEEE International Conference on Image Processing*, 3698–3702

Zou Y, Yu Z, Vijaya Kumar B, Wang J (2018) Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In: *Proceedings of the European Conference on Computer Vision*, 289–305

Appendix

Appendix I: LDR images from tone mapping operators

Figure 7.1 shows LDR images from every tone mapping operator which is used in this thesis. The images were generated from the first image of the HDR BIT training dataset, which was used to determine the parameter settings for each TMO. The images in figure 7.2 were generated from a



Drago operator

Linear clip operator

Reinhard global operator

Reinhard local operator

Schlick operator

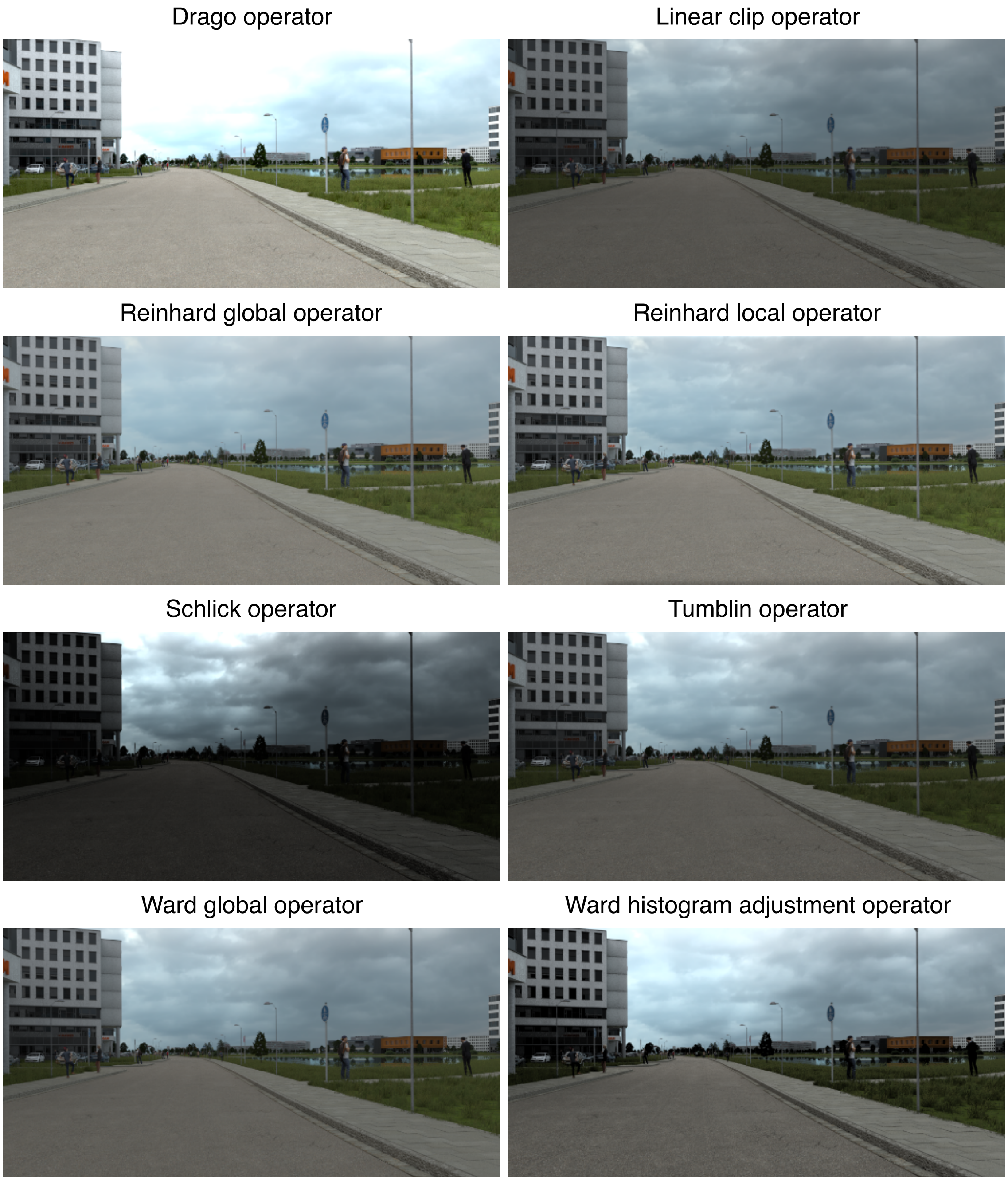
Tumblin operator

Ward global operator

Ward histogram adjustment operator

Figure 7.1 1st image in the BIT training dataset tone mapped by the eight different TMOs.

different HDR image in the training set. The differences between the tone mapped LDR images from the randomly selected HDR image are more concise than the one from the first image. This is best visible for the sky of the images.



Drago operator

Linear clip operator

Reinhard global operator

Reinhard local operator

Schlick operator

Tumblin operator

Ward global operator

Ward histogram adjustment operator

Figure 7.2 Randomly selected image in the BIT training dataset tone mapped by the eight different TMOs.

Appendix II: Semantic segmentation performances

Table 7.1 shows the semantic segmentation performances for the *closed system performance*. Table 7.2 presents the *LDR generalization capability* and table 7.3 the *HDR generalization capability*. The methods are described in section 3.2 and the experiments are presented in subsection 4.2.3.

Method	Road	Sidewalk	Building	Pole	Traffic Light	Traffic Sign	Vegetation	Terrain	Sky	Person	Car	mIoU	Accuracy
Tone mapped datasets													
Drago -1	0.87	0.57	0.78	0.11	0.00	0.11	0.50	0.00	0.61	0.39	0.17	0.37	0.82
Drago -2	0.90	0.64	0.80	0.10	0.00	0.13	0.49	0.00	0.67	0.39	0.19	0.39	0.84
Linear clip -1	0.90	0.64	0.79	0.13	0.00	0.13	0.51	0.00	0.65	0.44	0.19	0.40	0.84
Linear clip -2	**0.91**	0.66	0.81	**0.16**	0.00	0.16	0.51	0.00	0.66	0.44	0.22	0.41	0.85
Rein. glob. -1	**0.91**	0.70	0.80	0.12	0.00	0.20	0.52	0.00	0.65	0.43	0.24	**0.42**	0.85
Rein. glob. -2	**0.91**	0.68	0.81	0.11	0.00	0.20	**0.54**	0.00	0.68	0.41	0.26	**0.42**	**0.86**
Rein. loc. -1	0.90	0.56	0.80	0.14	0.00	0.15	0.52	0.00	0.63	0.45	0.20	0.40	0.84
Rein. loc. -2	0.90	0.62	0.79	0.13	0.00	0.15	0.50	0.00	0.64	0.45	0.20	0.40	0.84
Schlick -1	0.87	0.62	0.80	0.12	0.00	0.12	0.46	0.00	0.66	0.41	0.26	0.39	0.84
Schlick -2	0.88	0.60	**0.83**	0.12	0.00	0.11	0.49	0.00	**0.69**	0.39	0.27	0.40	0.85
Tumblin -1	0.89	0.65	0.79	0.10	0.00	0.17	0.48	0.00	0.64	0.45	0.25	0.40	0.84
Tumblin -2	0.89	0.59	0.77	0.11	0.00	0.13	0.52	0.00	0.65	0.46	0.22	0.40	0.83
Ward glob. -1	0.89	0.63	0.80	0.13	0.00	0.14	0.48	0.00	0.66	**0.47**	0.22	0.40	0.85
Ward glob. -2	0.88	0.57	0.80	0.15	0.00	0.13	0.51	0.00	0.66	0.45	0.21	0.40	0.84
Ward hi.adj. -1	0.89	0.68	0.80	0.12	0.00	0.14	0.53	0.00	0.68	0.42	0.29	0.41	0.85
Ward hi.adj. -2	0.90	0.68	0.81	0.12	0.00	0.10	0.51	0.00	0.67	0.43	0.32	0.41	0.85
Mixed LDR dataset													
HDR^3 -1	**0.91**	0.70	0.81	0.12	0.00	0.21	0.52	0.00	0.66	0.41	0.27	**0.42**	**0.86**
HDR^3 -2	**0.91**	0.71	0.80	0.10	0.00	0.21	0.51	0.00	0.65	0.36	0.27	0.41	0.85
HDR datasets													
HDR -1	0.86	0.57	0.75	0.15	0.00	0.13	0.41	0.00	0.55	0.44	0.33	0.38	0.81
HDR -2	0.85	0.48	0.75	0.13	0.00	0.10	0.46	0.00	0.64	0.35	0.21	0.36	0.80
HDR^1 -1	0.87	0.60	0.79	0.12	0.00	0.13	0.50	0.00	0.66	0.46	0.20	0.39	0.83
HDR^1 -2	0.90	0.67	0.80	**0.16**	0.00	0.13	0.52	0.00	0.66	0.44	0.27	0.41	0.85
HDR^2 -1	0.87	0.63	0.74	0.04	0.00	0.11	0.31	0.00	0.60	0.36	0.08	0.34	0.80
HDR^2 -2	0.89	0.72	0.78	0.08	0.00	0.17	0.38	0.00	0.67	0.40	0.13	0.38	0.84
HDR^{1,2} -1	0.90	**0.74**	0.80	0.11	0.00	0.12	0.50	0.00	0.65	**0.47**	**0.36**	**0.42**	0.85
HDR^{1,2} -2	**0.91**	0.73	0.79	0.10	0.00	**0.22**	0.51	0.00	0.64	0.46	0.31	**0.42**	0.85

Table 7.1 Every method in the table corresponds to a DeepLab-v2 model trained until convergence. The methods name indicates how the training images were generated (e.g. by the Drago tone mapping operator). Each training was repeated with identical settings (-1 and -2). The values in the table correspond to the class-wise mIoU scores determined from the predictions of the test dataset. For this scenario, the test set belongs to the same domain as the training dataset, meaning it was generated in the same way (e.g. also by the Drago tone mapping operator). This approach corresponds to the *closed system performance*. A more detailed description of the experiment is given in subsection 4.2.3.

Method	Road	Sidewalk	Building	Pole	Traffic Light	Traffic Sign	Vegetation	Terrain	Sky	Person	Car	mIoU	Accuracy
Tone mapped datasets													
Drago -1	**0.79**	0.24	0.58	0.19	0.05	0.33	0.55	0.14	0.37	0.45	0.67	0.40	0.74
Drago -2	0.77	0.26	0.59	0.20	0.06	0.37	0.54	0.17	0.45	0.51	0.66	0.42	0.74
Linear clip -1	0.75	0.24	0.64	0.19	0.05	0.35	**0.63**	0.18	0.40	0.50	0.69	0.42	0.75
Linear clip -2	0.77	0.24	0.60	0.19	0.04	0.35	**0.63**	**0.19**	0.40	0.44	0.69	0.41	0.75
Rein. glob. -1	**0.79**	0.28	0.62	0.20	0.07	0.34	0.57	**0.19**	0.50	0.46	0.69	0.43	0.76
Rein. glob. -2	0.78	0.27	0.64	0.19	0.06	0.35	0.61	0.18	0.48	0.48	0.69	0.43	0.77
Rein. loc. -1	0.78	0.24	0.64	0.19	0.07	0.34	0.60	0.15	0.45	0.47	0.65	0.42	0.76
Rein. loc. -2	0.76	0.26	0.61	0.20	0.06	0.34	0.55	0.14	0.41	0.49	0.66	0.41	0.74
Schlick -1	**0.79**	0.27	0.62	0.20	0.03	0.35	0.49	0.12	0.50	0.46	0.70	0.41	0.75
Schlick -2	0.76	0.25	0.65	0.21	0.03	0.36	0.52	0.12	0.45	0.47	0.66	0.41	0.74
Tumblin -1	**0.79**	0.28	0.63	0.21	0.06	0.36	0.61	0.17	0.48	0.48	0.67	0.43	0.77
Tumblin -2	0.77	0.27	0.63	0.20	0.08	0.37	0.60	0.16	0.40	0.51	0.70	0.43	0.76
Ward glob. -1	**0.79**	0.27	0.62	0.21	0.07	0.34	0.58	**0.19**	0.43	0.47	0.70	0.42	0.76
Ward glob. -2	**0.79**	0.26	0.64	0.22	0.06	0.36	0.58	0.17	0.48	0.47	0.68	0.43	0.77
Ward hi.adj. -1	0.75	0.27	0.64	0.22	**0.09**	0.37	0.59	0.15	0.48	0.47	0.69	0.43	0.75
Ward hi.adj. -2	0.78	0.28	0.65	**0.23**	0.07	0.37	0.57	0.17	0.50	0.47	0.71	0.43	0.77
Mixed LDR dataset													
HDR^3 -1	**0.79**	0.29	0.67	0.22	0.07	0.37	**0.63**	0.17	0.51	0.48	0.69	**0.45**	**0.78**
HDR^3 -2	**0.79**	0.30	**0.69**	0.21	0.07	**0.38**	**0.63**	0.16	0.45	**0.52**	**0.73**	**0.45**	**0.78**
HDR datasets													
HDR -1	0.74	0.23	0.59	0.18	0.04	0.34	0.50	0.11	0.52	0.43	0.69	0.40	0.73
HDR -2	0.73	0.23	0.61	0.20	0.05	0.36	0.56	0.12	0.36	0.44	0.65	0.39	0.73
HDR^1 -1	0.77	0.25	0.60	0.20	0.05	0.34	0.58	0.16	0.43	0.48	0.69	0.41	0.75
HDR^1 -2	0.76	0.26	0.63	0.20	0.05	0.37	0.58	0.15	**0.56**	0.48	0.70	0.43	0.76
HDR^2 -1	0.76	0.28	0.57	0.17	0.03	0.30	0.39	0.08	0.39	0.41	0.59	0.36	0.70
HDR^2 -2	0.76	**0.31**	0.62	0.16	0.02	0.33	0.45	0.09	0.49	0.45	0.66	0.39	0.73
HDR^{1,2} -1	0.74	0.26	0.63	0.18	0.02	0.33	0.58	0.14	0.51	0.48	0.66	0.41	0.75
HDR^{1,2} -2	0.78	0.28	0.60	0.19	0.03	0.34	0.53	0.13	0.48	0.45	0.66	0.41	0.75

Table 7.2 Every method in the table corresponds to a DeepLab-v2 model trained until convergence. The methods name indicates how the training images were generated (e.g. by the Drago tone mapping operator). Each training was repeated with identical settings (-1 and -2). The values in the table correspond to the class-wise mIoU scores determined from the predictions of the test dataset. For this scenario, the test set belongs to the 8 bit LDR Cityscapes dataset. This approach corresponds to the *LDR generalization capability*. A more detailed description of the experiment is given in subsection 4.2.3.

Method	Road	Sidewalk	Building	Pole	Traffic Light	Traffic Sign	Vegetation	Terrain	Sky	Person	Car	mIoU	Accuracy
Tone mapped datasets													
Drago -1	**0.81**	0.21	0.55	0.18	0.03	0.24	0.43	0.14	0.35	0.38	0.62	0.36	0.72
Drago -2	0.79	0.24	0.55	0.17	0.04	0.29	0.43	0.17	0.44	0.46	0.61	0.38	0.72
Linear clip -1	0.76	0.22	0.55	0.17	0.03	0.26	0.49	0.14	0.41	0.43	0.62	0.37	0.72
Linear clip -2	0.77	0.21	0.53	0.16	0.02	0.26	0.50	0.14	0.41	0.37	0.61	0.36	0.72
Rein. glob. -1	0.77	0.24	0.54	0.16	0.03	0.24	0.41	0.15	0.49	0.36	0.58	0.36	0.71
Rein. glob. -2	0.77	0.23	0.56	0.15	0.03	0.25	0.44	0.12	0.49	0.38	0.59	0.36	0.72
Rein. loc. -1	0.78	0.20	0.56	0.16	0.04	0.25	0.48	0.11	0.44	0.40	0.61	0.37	0.72
Rein. loc. -2	0.77	0.22	0.53	0.17	0.04	0.24	0.42	0.10	0.38	0.41	0.60	0.35	0.70
Schlick -1	0.79	0.25	0.58	0.18	0.03	0.31	0.45	0.13	0.50	0.41	0.66	0.39	0.74
Schlick -2	0.79	0.23	0.60	0.19	0.02	0.29	0.50	0.12	0.48	0.41	0.60	0.38	0.74
Tumblin -1	0.79	0.24	0.57	0.17	0.03	0.28	0.50	0.15	0.46	0.41	0.63	0.38	0.74
Tumblin -2	0.78	0.23	0.56	0.16	0.05	0.28	0.49	0.13	0.37	0.43	0.64	0.38	0.72
Ward glob. -1	0.79	0.24	0.55	0.19	0.05	0.24	0.42	0.15	0.44	0.37	0.62	0.37	0.72
Ward glob. -2	0.79	0.23	0.57	0.19	0.04	0.26	0.42	0.13	0.51	0.37	0.60	0.37	0.73
Ward hi.adj. -1	0.75	**0.27**	0.64	**0.22**	**0.09**	**0.37**	0.59	0.15	0.48	**0.47**	0.69	**0.43**	0.75
Ward hi.adj. -2	0.76	0.23	0.57	0.18	0.06	0.28	0.48	0.13	0.47	0.39	0.60	0.38	0.72
Mixed LDR dataset													
HDR^3 -1	0.80	0.26	0.64	0.20	0.04	0.32	0.58	0.18	0.50	0.42	0.66	0.42	0.77
HDR^3 -2	0.80	**0.27**	0.64	0.19	0.04	0.32	0.59	0.14	0.44	**0.47**	0.69	0.42	0.77
HDR datasets													
HDR -1	0.79	0.23	0.58	0.16	0.04	0.31	0.51	0.14	**0.59**	0.37	0.65	0.40	0.75
HDR -2	0.77	0.24	0.59	0.18	0.04	0.32	0.55	0.14	0.38	0.39	0.62	0.38	0.74
HDR^1 -1	**0.81**	0.25	0.64	0.18	0.06	0.30	**0.62**	**0.19**	0.46	0.45	**0.70**	0.42	**0.78**
HDR^1 -2	0.80	0.25	**0.65**	0.17	0.03	0.32	0.61	**0.19**	0.57	0.42	0.69	**0.43**	**0.78**
HDR^2 -1	0.74	0.25	0.52	0.13	0.05	0.25	0.34	0.07	0.32	0.35	0.52	0.32	0.66
HDR^2 -2	0.76	**0.27**	0.57	0.13	0.02	0.29	0.41	0.08	0.44	0.38	0.60	0.36	0.71
HDR^{1,2} -1	0.78	0.25	0.62	0.16	0.02	0.32	0.55	0.11	0.51	0.45	0.64	0.40	0.75
HDR^{1,2} -2	0.79	**0.27**	0.61	0.18	0.02	0.33	0.53	0.10	0.47	0.41	0.64	0.40	0.75

Table 7.3 Every method in the table corresponds to a DeepLab-v2 model trained until convergence. The methods name indicates how the training images were generated (e.g. by the Drago tone mapping operator). Each training was repeated with identical settings (-1 and -2). The values in the table correspond to the class-wise mIoU scores determined from the predictions of the test dataset. For this scenario, the test set belongs to the 16 bit HDR Cityscapes dataset. This approach corresponds to the *HDR generalization capability*. A more detailed description of the experiment is given in subsection 4.2.3.

Appendix III: Domain adaptation results

Table 7.4 shows the results from the domain adaptation experiments, which are described in subsection 4.3.2.

Method	Road	Sidewalk	Building	Pole	Traffic Light	Traffic Sign	Vegetation	Terrain	Sky	Person	Car	mIoU	Accuracy
Performance on target domain (LDR^t)													
Linear clip -1	0.75	0.24	0.64	0.19	0.05	0.35	**0.63**	0.18	0.40	0.50	0.69	0.42	0.75
Linear clip -2	0.77	0.24	0.60	0.19	0.04	0.35	**0.63**	**0.19**	0.40	0.44	0.69	0.41	0.75
HDR^1 -1	0.77	0.25	0.60	0.20	0.05	0.34	0.58	0.16	0.43	0.48	0.69	0.41	0.75
HDR^1 -2	0.76	0.26	0.63	0.20	0.05	**0.37**	0.58	0.15	0.56	0.48	0.70	0.43	0.76
Lin. → LDR^t -1	**0.87**	**0.35**	**0.68**	0.20	0.02	0.36	0.59	0.15	0.64	0.50	0.73	0.46	**0.81**
Lin. → LDR^t -2	**0.87**	**0.35**	0.67	**0.21**	0.05	**0.37**	0.58	0.17	0.63	**0.51**	**0.75**	**0.47**	**0.81**
HDR^1 → LDR^t -1	0.85	0.33	0.64	**0.21**	0.04	0.35	0.49	0.15	0.68	0.48	0.72	0.45	0.79
HDR^1 → LDR^t -2	0.84	0.32	0.61	0.18	0.02	0.34	0.48	0.17	**0.71**	0.43	0.69	0.43	0.78
Performance on target domain (HDR^t)													
Linear clip -1	0.76	0.22	0.55	0.17	0.03	0.26	0.49	0.14	0.41	0.43	0.62	0.37	0.72
Linear clip -2	0.77	0.21	0.53	0.16	0.02	0.26	0.50	0.14	0.41	0.37	0.61	0.36	0.72
HDR^1 -1	0.81	0.25	0.64	0.18	**0.06**	0.30	0.62	**0.19**	0.46	0.45	0.70	0.42	0.78
HDR^1 -2	0.80	0.25	0.65	0.17	0.03	0.32	0.61	**0.19**	0.57	0.42	0.69	0.43	0.78
Lin. → HDR^t -1	0.86	0.29	0.66	0.16	0.04	0.30	0.56	0.13	0.56	0.43	0.71	0.43	0.79
Lin. → HDR^t -2	0.86	0.29	0.66	0.15	0.03	0.30	0.55	0.12	0.57	0.42	0.70	0.42	0.79
HDR^1 → HDR^t -1	0.82	0.27	**0.68**	0.20	0.03	0.31	0.56	0.13	0.63	0.43	0.70	0.43	0.79
HDR^1 → HDR^t -2	0.83	0.30	0.65	0.18	0.03	0.32	0.53	0.11	0.61	0.44	0.71	0.43	0.78

Table 7.4 Results of the domain adaptation.

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

....

date   (signature)