Self-Supervised Domain Mismatch Estimation for Autonomous Perception

Jonas Löhdefink¹  Justin Fehrling¹  Marvin Klingner¹  Fabian Hüger²  Peter Schlicht²
Nico M. Schmidt²  Tim Fingscheidt¹

¹Technische Universität Braunschweig  ²Volkswagen Group Automation

Abstract

Autonomous driving requires self awareness of its perception functions. Technically spoken, this can be realized by observers, which monitor the performance indicators of various perception modules. In this work we choose, exemplarily, a semantic segmentation to be monitored, and propose an autoencoder, trained in a self-supervised fashion on the very same training data as the semantic segmentation to be monitored. While the autoencoder’s image reconstruction performance (PSNR) during online inference shows already a good predictive power w.r.t. semantic segmentation performance, we propose a novel domain mismatch metric DM as the earth mover’s distance between a pre-stored PSNR distribution on training (source) data, and an online-acquired PSNR distribution on any inference (target) data. We are able to show by experiments that the DM metric has a strong rank order correlation with the semantic segmentation within its functional scope. We also propose a training domain-dependent threshold for the DM metric to define this functional scope.

1. Introduction

Semantic segmentation is an essential function concerning camera-based perception for autonomous driving. Because of its highly safety-critical nature, it is crucial to observe the performance during inference. Domain shifts in the input space of images are one of the various issues that come into play, being part of everyday scenarios and must be handled. These domain shifts could be, e.g., changing lighting or weather conditions such as rain or fog. The first step towards a better assessment of the input domain is to detect and measure an occurring domain shift.

The commonly used quality measure for object detection and semantic segmentation is the mean intersection over union (mIoU). Unfortunately, an mIoU can only be computed with ground truth semantic segmentation labels, which are not available online during driving, of course. Besides semantic segmentation networks, we assume that other learned functions (even for different tasks) also perform worse when it comes to a performance degradation of the segmentation caused by a domain shift, assuming they were trained on the same data distribution. Hence, we propose the use of a (self-supervised) autoencoder, which allows to monitor domain shifts by computation of a peak-signal-to-noise ratio (PSNR) between input and output images without requiring labels, see Figure 1. Clearly, it is difficult to determine the domain shift on single images as there may always be unusual images, so we focus on investigating batches of images. In fact, we train and evaluate the framework on various datasets simulating domain shifts. A first simple approach to estimate the domain shift is to evaluate the resulting autoencoder’s mean PSNR scores. We also compute PSNR performance histograms both for the training data and for different inference data domains and compare them by the earth mover’s
distance (EMD) [59], obtaining a domain mismatch (DM) metric between two datasets. In our experimental evaluation, we evaluate the PSNR and our novel DM metric with the absolute segmentation performance difference in mIoU, showing a strong correlation for both.

The rest of this paper is structured as follows: Section 2 presents an overview of the state of the art for related fields of research. In Section 3, we explain the details of our domain mismatch estimation. Section 4 then discusses and interprets the results of the conducted experiments. Finally, we conclude our findings in Section 5.

2. Related Work

In this section we provide an overview of the most relevant state-of-the-art approaches of semantic segmentation, autoencoders, and domain shift.

Semantic Segmentation can be considered as pixel-wise classification of images. Some areas of applications for semantic segmentation are medical image analysis, perception of autonomous driving [5, 6], video surveillance, and augmented reality [38].

The architectural concepts for semantic segmentation can be categorized into fully convolutional networks (FCNs) [33], graphical models [11], encoder-decoder based models [40], multi-scale architectures [30], region CNNs (R-CNNs) [24], networks based on dilated convolutions [11, 12], recurrent neural networks (RNNs) [53], attention-based models [13], generative adversarial networks (GANs) [20, 34], and active contour models [26], as comprehensively investigated in [38]. Furthermore, there is also a variety of image segmentation datasets in 2D, 2.5D (including depth), and 3D. Often used 2D datasets are PASCAL VOC [16], PASCAL VOC12 [15], MS COCO [31], Cityscapes [14], KITTI [19], SYNTHIA [45], Berkeley DeepDrive [60], and CamVid [9]. For the evaluation of semantic segmentation models, several quality measures are frequently used, e.g., pixel accuracy (PA), mean pixel accuracy (MPA), mean intersection over union (mIoU), precision, F1-score, and dice coefficient [38].

Due to the efficient implementation and therefore also training and inference time savings, we use the encoder-decoder-based ERFNet [44], which adopts its architecture from [42] and [4]. For our experiments, we use the Cityscapes dataset [14], the KITTI dataset [19] and the Berkeley DeepDrive dataset [60] and report the mIoU since it is the most wide-spread segmentation metric.

Autoencoders are a special case of encoder-decoder architectures, trained to have the same input and output in a self-supervised fashion. Variations of autoencoders can be found in their respective architectures, loss functions, learning principles, and strategies.

Due to the bottleneck in the autoencoder, it is inherently closely related to image compression [2, 32, 49], which often adds quantization, and also to image (and video) super-resolution (SR) methods [22, 37], focusing on reconstructing the original high-resolution image from a low-resolution representation. Furthermore, also texture synthesis [29, 51], image inpainting [58, 61], and style transfer [18, 25] incorporate autoencoder structures. In many cases, decoders make use of transposed convolutions [62, 63] and multi-task learning [10, 24]. Besides this, many architectures use generative adversarial networks (GANs) [20] or extensions such as the conditional GAN (cGAN) [39], or the least squares GAN (LSGAN) [36]. The Wasserstein GAN (WGAN) [3] is another famous representative of GANs, using the Wasserstein-1 distance, also known as the earth mover’s distance (EMD) [59], which we will use as domain mismatch metric. Commonly used quality measures for image compression systems, super resolution approaches, and autoencoders in general are peak-signal-to-noise ratio (PSNR) [32, 47], structural similarity (SSIM) [55], and multi-scale SSIM (MS-SSIM) [56], as well as the mean opinion score (MOS), which is the human-evaluated perceptual quality. Besides, there are numerous other image quality assessment methods, trying to simulate the human perception system [35, 48].

We use the autoencoder architecture for learned image compression from [2], with the difference that we omit the quantization block, since we do not aim at compression.

Domain Shift deals with variations between data domains or distributions, while domains can be considered as environments of different technical or natural data characteristics and different data distributions. Examples for such domain shifts are differing sensor setups in capture devices, or traffic signs in different countries.

Learning models on data distributions differing from the application distributions is referred to as transfer learning [41, 52], since the goal is to transfer the learned knowledge. Specifically, domain adaptation approaches [7, 17] aim at adjusting models to perform well in two (or more) domains in a (semi-)supervised or unsupervised fashion. Moreover, time-variant domains often lead to conceptual shifts [50, 57], posing a particularly difficult problem, since the direction of the drift is unknown. This makes the drift even more important to detect. The maximum mean discrepancy (MMD) [8, 21] is another task-independent method to measure a domain shift between a source and a target domain. In this technique, a function in a reproducing kernel Hilbert space (RKHS) is to be found, being large for samples from the first distribution p and small for samples from the second distribution q. The MMD then is computed by subtracting the mean of function outputs with inputs from q from the mean of function outputs with inputs from p. This method can be thought of comparing not only the means of two distributions but also their higher order moments such as the variance.
The main differences between the MMD and our method is that, first, the MMD maximizes the sample expectation differences from two distributions in a reproducing kernel Hilbert space over a set of functions for each domain pair to be evaluated, while our proposed method is trained only once on the training (source) domain. Second, the MMD uses the difference of mean values to obtain the final metric, while we evaluate the outputs by the EMD. And third, we use neural networks both for semantic segmentation and for domain mismatch estimation, while the function optimized in the typical MMD is not related to neural networks.

3. Domain Mismatch Estimation

A detailed block diagram of the proposed domain mismatch estimation can be seen in Figure 2. It consists of an autoencoder along with a loss function and computational steps to obtain a domain mismatch metric $DM$. The image $x = (x_i)$ with height $H$ and width $W$, consisting of normalized (color) pixels $x_i \in [-1,1]^C$, with $C = 3$ color channels and pixel index $i \in I = \{1,2,\ldots,H \cdot W\}$, is the input to both, an undisplayed but to be observed semantic segmentation, and to our proposed domain mismatch estimator. Its autoencoder receives the normalized image $x$ and produces an image reconstruction $\hat{x} = (\hat{x}_i)$ with $\hat{x}_i \in [-1,1]^C$. An advantage of all autoencoder settings is the fact that no explicit labels are needed because of its self-supervised training. So in addition to the image reconstruction, the loss and quality measure also use the input image $x$. Different domains result in different self-supervised quality measure distributions, which can then be compared by the earth mover’s distance [39], providing our proposed domain mismatch metric.

3.1. Network Architectures and Losses

We use the ERFNet [44] for the task of semantic segmentation to be observed. The network is optimized to run in real-time, while still achieving accurate results. It has an encoder/decoder structure and makes use of factorized residual layers consisting of a combination of two 1D filters instead one 2D filter. Since the semantic segmentation architecture and loss function are identical to that used in [44], we refer the interested reader to this reference.

Concerning our autoencoder, we use an adversarial architecture adopted from [2], [54], and [23]. Speaking in terms of a generative adversarial network, the generator combines the encoder and decoder networks of the autoencoder and the discriminator evaluates its reconstructions in a simultaneous training. In the encoder, decoder, and discriminator, each convolutional operation is zero-padded, always preserving the image dimensions, and followed by an instance normalization layer as well as a ReLU activation function if not stated otherwise.

First in the encoder, there is a convolutional layer with kernel size $7 \times 7$, stride of 1, and 60 feature maps. Afterwards, 4 downsampling blocks follow, each consisting of a convolutional layer with kernel size $3 \times 3$, and a stride of two for spatial reduction of the $(120, 240, 480, 960)$ feature maps. The last convolutional layer has a kernel size $3 \times 3$, stride of one, and 8 feature maps, shaping the bottleneck. The final encoder layer has a tanh activation to yield outputs in the range $[-1, 1]$.

The decoder architecture first has a convolutional layer with kernel size $3 \times 3$, stride of one, and 960 feature maps. Afterwards, there are 9 residual blocks, each consisting of two convolutional layers, bypassed by an identity function, where the second convolutional layer omits the ReLU activation function. The initial image resolution is restored by 4 transposed convolutional layers with kernel size $4 \times 4$, stride of two, and $(960, 480, 240, 120)$ feature maps. The architecture is finalized by a convolutional layer with kernel size $7 \times 7$, stride of 1, three feature maps, and a tanh activation function.
In the discriminator, instead of the ReLU activation function, the leakyReLU function is used. The discriminator consists of 4 convolutional layers with kernel size $4 \times 4$, stride of 2, and $(64, 128, 256, 512)$ feature maps. A final convolutional layer with kernel size $4 \times 4$, stride of one, one feature map, and ReLU activation delivers the discriminator outputs.

The autoencoder loss

$$J^{AE} = \alpha_1 J^{dist} + \alpha_2 J^{FM} + (1 - \alpha_1 - \alpha_2) J^{G, \text{adv}},$$

(1)

with the weighting factors $\alpha_1, \alpha_2 \in [0, 1]$, $\alpha_1 + \alpha_2 \leq 1$, consists of an MSE distortion loss $J^{dist}$, the L1 feature map loss $J^{FM}$ between the discriminator’s feature activations fed with the image $x$ and the reconstruction $\hat{x}$, and the generator-specific least-squares (LS) GAN loss $J^{G, \text{adv}}$ [36]. The discriminator is trained with the discriminator-specific LS-GAN loss $J^{D, \text{adv}}$, which pursues the opposed goal of the generator.

### 3.2. Quality Measures

Evaluating the semantic segmentation performance for a set of images, commonly the mean intersection over union

$$mIoU = \frac{1}{|S|} \sum_{s \in S} \frac{TP_s}{TP_s + FP_s + FN_s}$$

(2)

is used, being composed of the numbers of true-positive ($TP_s$) pixels, false-positive ($FP_s$) pixels, and false-negative ($FN_s$) pixels w.r.t. the ground truth, with the class index $s \in S = \{1, 2, ..., S\}$, being summed up over all images before.

For the evaluation of the autoencoder, the image reconstruction quality for input and output color image pixels in the number range $x_i', \hat{x}_i' \in [0, 255]^C$ usually is computed by the peak signal-to-noise ratio (PSNR), performing a direct MSE comparison of pixel values:

$$\text{PSNR} = 10 \log \left( \frac{x_{max}'^2}{\sum_{i \in I} \|x_i' - \hat{x}_i'\|^2} \right) [\text{dB}]$$

(3)

with $x_{max}' = 255$.

The comparison of two discrete probability distributions $P(\mu)$, $\mu \in M = \{1, 2, ..., M\}$ and $Q(\nu)$, $\nu \in N = \{1, 2, ..., N\}$ can be computed by the earth-mover’s distance (EMD) [59]. This metric computes the minimum work $W$ required to convert one distribution into the other by multiplying the distance $d_{\mu\nu} = |\mu - \nu| \in \{0, 1, ..., \max(M, N) - 1\}$ between the bins with index $\mu$ and $\nu$ with the $M \times N$ flow matrix $F = (f_{\mu\nu})$ with $f_{\mu\nu} \in [0, 1]$ being the flow from bin $\mu$ to $\nu$. The optimal flow is found by minimizing the work according to

$$F^* = \arg \min_F W(P, Q, F) = \arg \min_F \sum_{\mu \in M} \sum_{\nu \in N} f_{\mu\nu} d_{\mu\nu}$$

(4)

under consideration of the four (stochastic) constraints

$$f_{\mu\nu} \geq 0, \quad \mu \in M, \nu \in N$$

$$\sum_{\nu \in N} f_{\mu\nu} \leq P(\mu), \quad \mu \in M$$

$$\sum_{\mu \in M} f_{\mu\nu} \leq Q(\nu), \quad \nu \in N$$

$$\sum_{\mu \in M} \sum_{\nu \in N} f_{\mu\nu} = \min(P(\mu), Q(\nu)).$$

We then obtain the earth-mover’s distance as

$$DM(P, Q) = \sum_{\mu \in M} \sum_{\nu \in N} \frac{f^*_{\mu\nu} d_{\mu\nu}}{\sum_{\mu \in M} \sum_{\nu \in N} f^*_{\mu\nu}},$$

(5)

which we will use as our proposed domain mismatch metric by computing the difference of reconstruction qualities for various datasets.

We use Kendall’s rank order coefficient $[1, 27] \tau = \tau_b$, which accounts for ties in one quantity, whereby in the following we will omit the index b. Having $K$ observations $o_k = (a_k, b_k)$ with $k \in \{1, ..., K\}$, the total number of observation pairs

$$(o_k, o_\ell) = ((a_k, b_k), (a_\ell, b_\ell))$$

(6)

with $k < \ell$ is $n_0 = \binom{K}{2} = \frac{1}{2}K(K-1)$. A pair of observations is called concordant if the observation’s components have the same order (both ascending or both descending), otherwise it is discordant. If the values of one component in the pair are equal, it is called a tie in this component (here: a tie in $a$ or a tie in $b$) and is neither concordant nor discordant. The number of concordant pairs $n_c$, discordant pairs $n_d$, ties in $a$, and ties in $b$ is used to calculate Kendall’s rank order coefficient

$$\tau = \frac{n_c - n_d}{\sqrt{(n - n_b)(n - n_b)}} \in [-1, 1],$$

(7)

where $\tau = 1$ means that the observations are perfectly in the same order, $\tau = -1$ means that they are perfectly in reversed order, and $\tau = 0$ means that there is no correlation in rank order.

### 4. Evaluation and Discussion

In this section, we will introduce the training setup and describe the performance of the segmentation and autoencoder networks on different datasets, as well as we will analyze the proposed method for domain mismatch estimation.

#### 4.1. Data Configurations and Training

For experimental evaluation, we use Cityscapes [14], containing images from several German cities, Berkeley DeepDrive [60], containing data from the U.S., and
KITT [19], containing data from a single German city including surroundings. All these datasets provide the same class labeling scheme for segmentation and are therefore compatible. Furthermore, they all provide a training and a validation set with segmentation labels. For our experiments we distinguish between the Cityscapes training set (CS\text{train}), the Cityscapes validation set (CS\text{val}), the Berkeley DeepDrive training set (BDD\text{train}), the Berkeley DeepDrive validation set (BDD\text{val}), and the KITTI set (which consists of all first images in the stereo training set of KITTI2015). CS\text{train} and CS\text{val} consists of 2,975 and 500 images, respectively, and are downsampled to 512 × 1024 pixels. BDD\text{train} and BDD\text{val} have 7,000 and 1,000 images, respectively, with a resolution of 1280 × 720 pixels. Finally, the KITTI training split has 200 images with a resolution of 375 × 1242 pixels. The models for the semantic segmentation and the autoencoder are trained with PyTorch [43] either with CS\text{train} or BDD\text{train} on an NVIDIA GTX 1080 Ti GPU.

The encoder of the segmentation network is pretrained on ImageNet [46]. For data augmentation, the training images are randomly flipped horizontally and cropped to 192 × 640 pixels. After the pretraining, we continue training for 200 epochs with a batch size of 6, an initial learning rate of 0.0005, an Adam optimizer [28] with \( \beta_1 = 0.9 \) and \( \beta_2 = 0.999 \), and a weight decay of 0.0002, while ignoring the background class.

The GAN training procedure first optimizes the generator while fixing the discriminator weights, and vice versa afterwards. We train for 50 epochs with batch size 1, and an initial learning rate of 0.0002, using an Adam optimizer with \( \beta_1 = 0.5 \) and \( \beta_2 = 0.999 \). Concerning the autoencoder loss function (1), we use the weighting factors \( \alpha_1 = \frac{12}{25} \) for the MSE loss and \( \alpha_2 = \frac{10}{25} \) for the feature matching loss. Furthermore, early stopping w.r.t. the PSNR on the validation set is applied.

### 4.2. Domain-Specific Performance

In this section, we first evaluate the performance of semantic segmentation and autoencoder individually with mIoU (2) and PSNR (3), respectively, for the different datasets. The results for the models trained on CS\text{train} and BDD\text{train} can be seen in Table 1. We also report Kendall’s rank order coefficient \( \tau \) (7), evaluating the degree of rank similarity of the PSNR and mIoU series.

For the CS\text{train}-trained autoencoder, the PSNR performance is best on CS\text{train} (obviously because it is the training set) and performs second best on CS\text{val}, which is also plausible since it is the in-domain case. Evaluated on BDD\text{train} and BDD\text{val} the PSNR falls by several dB compared to the source domain to 21.01 dB and 21.26 dB, respectively, due to the domain shift. The lowest performance is achieved on KITTI with 20.13 dB. We observe a similar ranking of performances in the semantic segmentation results of the segmentation trained on CS\text{train}, with the surprising exception that the KITTI dataset this time does not yield the largest drop in mIoU. When comparing rank orders, only the positions of BDD\text{val} and KITTI seem to be swapped. The rank order coefficient \( \tau \in [-1, 1] \) is 0.6, still indicating a positive correlation in the behavior of PSNR and mIoU. Conclusively, we observe a huge domain-shift-induced performance drop for both models trained on the Cityscapes data, and evaluated on BDD and KITTI data.

As before, the autoencoder trained on BDD\text{train} performs best in its own domain with a PSNR of 25.87 dB on the training set and 25.37 dB on the validation set. Evaluation on CS\text{train} and CS\text{val} is ranked third and fourth w.r.t. PSNR, even though the dB difference to the source domain is quite small. The performance on KITTI is again lower than on the other datasets. In the semantic segmentation, the mIoU again is best for the in-domain datasets BDD\text{train} and BDD\text{val}, while CS\text{train}, CS\text{val}, and KITTI achieve similar mIoU, which is a bit in contrast to the autoencoder performance, which indicates that KITTI has a larger domain shift than the others. Kendall’s \( \tau \) is 0.8, underlining the strong correlation of rank orders.

The models trained on CS\text{train} and BDD\text{train} show at least similar trends in both of the investigated tasks (autoencoder and segmentation), which encourages us to assign the autoencoder the role of an observer for the semantic segmentation. The general trend is: Once PSNR drops, also mIoU can be assumed to drop, while the achievable absolute PSNR scores are data-dependent. This makes it a bit tedious to define a threshold for an acceptable domain shift, since it varies for each training dataset. Rank orders are not

<table>
<thead>
<tr>
<th>Trained on</th>
<th>Model</th>
<th>Measure</th>
<th>CS\text{train}</th>
<th>CS\text{val}</th>
<th>BDD\text{train}</th>
<th>BDD\text{val}</th>
<th>KITTI</th>
<th>Kendall ( \tau )</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS\text{train}</td>
<td>Autoencoder</td>
<td>PSNR</td>
<td>29.55 dB</td>
<td>28.24 dB</td>
<td>21.01 dB</td>
<td>21.26 dB</td>
<td>20.13 dB</td>
<td>0.6</td>
</tr>
<tr>
<td>segmentation</td>
<td>mIoU</td>
<td>81.2%</td>
<td>66.7%</td>
<td>23.1%</td>
<td>26.7%</td>
<td>51.1%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BDD\text{train}</td>
<td>Autoencoder</td>
<td>PSNR</td>
<td>25.18 dB</td>
<td>25.13 dB</td>
<td>25.87 dB</td>
<td>25.37 dB</td>
<td>22.10 dB</td>
<td>0.8</td>
</tr>
<tr>
<td>segmentation</td>
<td>mIoU</td>
<td>45.5%</td>
<td>43.9%</td>
<td>53.8%</td>
<td>49.0%</td>
<td>44.1%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Mean PSNR results for the autoencoder and mIoU results for the semantic segmentation trained and evaluated on various datasets.
necessarily kept in the low PSNR regime (BDD\textsubscript{train} \& BDD\textsubscript{val}, KITTI) for models trained on CS\textsubscript{train}, and (CS\textsubscript{val}, KITTI) for models trained on BDD\textsubscript{train}. However, even here we can reliably always assume that mIoU drops as well to unacceptable low values. Already in this preliminary experiment, investigating mean performance scores, we observed that if semantic segmentation performance (mIoU) drops below training or validation set performance, also autoencoder performance (PSNR) drops.

### 4.3. Domain Mismatch

For better visualization of domains, Figure 3 shows PSNR histograms, resulting from the evaluation on the individual datasets. For both source domains CS and BDD, evaluating the training set itself yields smooth distributions of PSNR scores around their mean values as expected (almost Gaussian), see Figures 3a and 3e. The transition to the validation set in the source domain and further on to one of the target domains implies a decrease of the mean PSNR and an increase of the standard deviation in the distribution, as can be seen in the Figures 3b to 3d for the CS-trained autoencoder, and in Figures 3f to 3h for the BDD-trained autoencoder. Noteworthy, the KITTI dataset is only from a single German city, which may be the cause for the small standard deviation in the histograms 3d and 3h.

Table 2 shows the mIoU differences and earth mover’s distance (EMD) scores, namely our proposed domain mismatch scores DM (5), based on the PSNR histograms for the segmentation and the autoencoder, respectively. Also, Kendall’s rank order coefficient τ is provided, here evaluating the rank order similarity of the DM and ∆mIoU series. The segmentation performance drop is simply stated as the mIoU difference between the training domains (CS\textsubscript{train} \& BDD\textsubscript{train}, respectively) and the target domains.

In consideration of the results for the Cityscapes-trained models, the DM metric for the validation set (here: 1.31 dB) indicates what is to be considered as default (or: typical) domain shift for in-domain data. For each of the out-of-domain shifts, regardless whether the target domain is BDD\textsubscript{train}, BDD\textsubscript{val}, or KITTI, the autoencoder reconstruction performance dropped significantly, so our DM metric increased to 8 dB and more. In each of these cases also the drop in the mIoU is large, with ∆mIoU being more than 50 % absolute for both BDD splits and 30.1 % for KITTI. Again, the mIoU drop on KITTI is not the worst (although the DM metric is), but a 30.1 % absolute mIoU drop definitely justifies KITTI to be “out-of-domain”, as it is marked by the high DM = 9.41 dB. The pure rank orders in the DM metric and the ∆mIoU series lead to a rank order coefficient τ of 0.6, which is still indicating a positive rank correlation.

Considering the models trained on BDD\textsubscript{train}, the val-
Table 2: Domain mismatch metric \( DM \) (5), absolute mIoU differences between the references (\( CS_{\text{val}} \) and \( BDD_{\text{val}} \)) and various datasets, and Kendall’s rank order \( \tau \).

<table>
<thead>
<tr>
<th>Trained on</th>
<th>Reference</th>
<th>Model</th>
<th>Measure</th>
<th>( CS_{\text{train}} )</th>
<th>( CS_{\text{val}} )</th>
<th>( BDD_{\text{train}} )</th>
<th>( BDD_{\text{val}} )</th>
<th>( KITTI )</th>
<th>Kendall ( \tau )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( CS_{\text{train}} )</td>
<td>( CS_{\text{train}} )</td>
<td>Autoencoder</td>
<td>DM</td>
<td>0.0 dB</td>
<td>1.31 dB</td>
<td>8.53 dB</td>
<td>8.29 dB</td>
<td>9.41 dB</td>
<td>0.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Segmentation</td>
<td>( \Delta \text{mIoU} )</td>
<td>0.0 %</td>
<td>14.5 %</td>
<td>58.1 %</td>
<td>54.5 %</td>
<td>30.1 %</td>
<td></td>
</tr>
<tr>
<td>( BDD_{\text{train}} )</td>
<td>( BDD_{\text{train}} )</td>
<td>Autoencoder</td>
<td>DM</td>
<td>0.68 dB</td>
<td>0.74 dB</td>
<td>0.0 dB</td>
<td>0.51 dB</td>
<td>3.77 dB</td>
<td>0.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Segmentation</td>
<td>( \Delta \text{mIoU} )</td>
<td>8.3 %</td>
<td>9.9 %</td>
<td>0.0 %</td>
<td>4.8 %</td>
<td>9.7 %</td>
<td></td>
</tr>
</tbody>
</table>

We infer that the autoencoder is even more sensitive to domain shifts than the semantic segmentation, since for both training datasets, the PSNR evaluated on KITTI dropped significantly while the mIoU showed a smaller decrease. Nevertheless, for small values of our DM metric, the experiments show that the rank orders are concordant, as can especially be seen for the BDD-trained models. Therefore, we propose to set a threshold for the DM metric to define its functional scope, in which the rank orders of the DM metric are expected to correspond to those of the \( \Delta \text{mIoU} \). The threshold should be two times the DM score of the in-domain validation set, so it is depending on the specific domain it is trained and validated in. Hence, for the CS-trained autoencoder the threshold lies at \( 2 \times 1.31 \text{ dB} = 2.62 \text{ dB} \), excluding \( BDD_{\text{train}}, BDD_{\text{val}} \), and KITTI from the functional scope (meaning these are clearly out-of-domain datasets!), and for the BDD-trained autoencoder the threshold is \( 2 \times 0.51 \text{ dB} = 1.02 \text{ dB} \), which excludes only the KITTI dataset. Inside its functional scope, the DM metric makes a statement about the semantic segmentation performance with concordant rank ordering. In comparison to the PSNR, we believe that the DM metric is the better generalizing metric, since the proposed threshold is relying on PSNR distributions, and is therefore less sensitive to single unusual images which do not yet necessarily make up a domain shift. As a result, the autoencoder is well-suited as a batch-type observer, since the DM metric exhibits reliable gradual estimations of the domain shift until exceeding the DM threshold, where the PSNR will collapse even before the mIoU of the semantic segmentation. DM results beyond the DM threshold always indicate a critical domain shift.

5. Conclusions

Observing the performance of safety-critical perception functions during autonomous driving is essential, because vehicles are by nature exposed to various environments, implying domain shifts. We proposed a novel framework to monitor the quality of a semantic segmentation. We accomplish this by estimating the domain shift by an autoencoder trained in self-supervised fashion. A first approach is to evaluate mean PSNR scores which already show a strong rank order correlation to the mIoU. However, comparing autoencoder outputs for various datasets by the earth mover’s distance yields a more stable estimation of the domain shift which we propose as domain mismatch DM metric. We found that the task of reconstructing an image is even more sensitive to domain shifts than semantic segmentation, being pixel-wise classification, which ultimately results in a certain functional scope for the autoencoder, beyond which input data can be clearly classified as “out-of-domain”. Within the valid functional scope of the autoencoder rank orders of our DM metric and mIoU differences are strongly rank-correlated. The proposed DM metric is therefore shown to be well-suited as an observer.

Acknowledgment

The research, leading to the results presented above, is funded by the German Federal Ministry for Economic Affairs and Energy within the project “KI Absicherung – Safe AI for automated driving”.

References


