Self-Supervised Domain Mismatch Estimation for Autonomous Perception

Jonas Löhdefink[1] Justin Fehrling[1] Marvin Klingner[1] Fabian Hüger[2] Peter Schlicht[2]
Nico M. Schmidt[2] Tim Fingscheidt[1]

{j.loehdefink, j.fehrling, m.klingner, t.fingscheidt}@tu-bs.de
{fabian.hueger, peter.schlicht, nico.maurice.schmidt}@volkswagen.de

[1]Technische Universität Braunschweig [2]Volkswagen 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

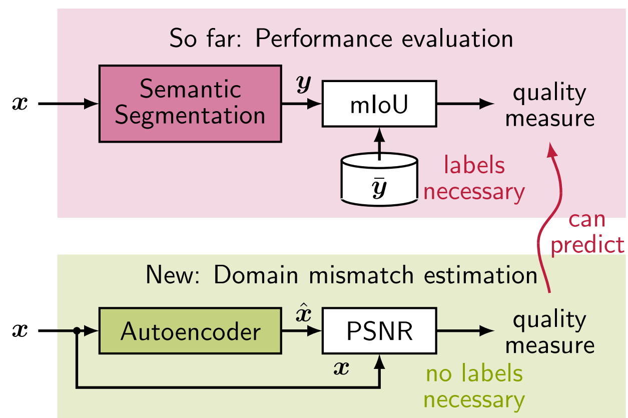


Figure 1: **Performance evaluation of semantic segmentation** (simplified sketch). Evaluation of the mean intersection over union (mIoU) requires ground truth segmentation labels $\bar{y}$, while the proposed domain mismatch estimation is performed on the basis of the PSNR of an autoencoder, trained and evaluated without labels.

at hand, 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], focussing 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 drifts [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.

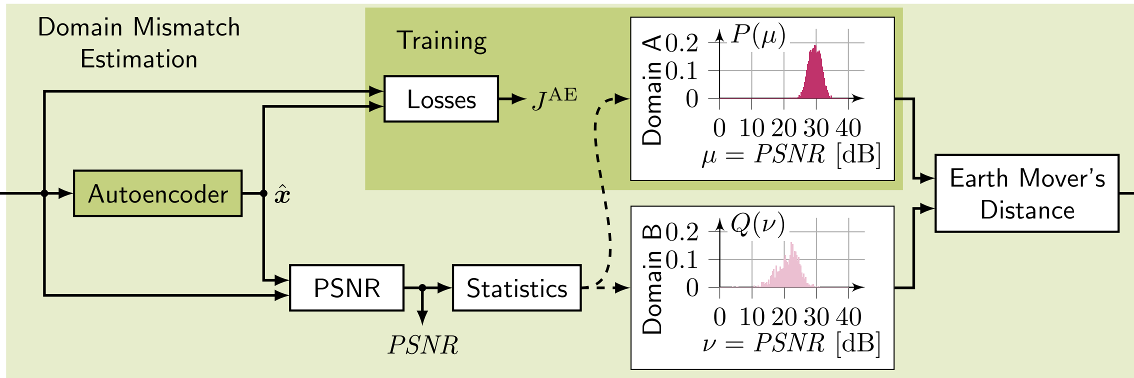


Figure 2: **Our proposed domain mismatch estimation**. The loss function for the autoencoder is only used during self-supervised training and is not needed during inference. The histogram of $PSNR$ values in the training data domain (A) is compared to an acquired histogram during inference (domain B), using the earth mover's distance (EMD), yielding the proposed domain mismatch metric DM.

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 $\boldsymbol{x} = (\boldsymbol{x}_i)$ with height H and width W, consisting of normalized (color) pixels $\boldsymbol{x}_i \in [-1, 1]^C$, with $C = 3$ color channels and pixel index $i \in \mathcal{I} = \{1, 2, ..., H \cdot W\}$, 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 $\boldsymbol{x}$ and produces an image reconstruction $\hat{\boldsymbol{x}} = (\hat{\boldsymbol{x}}_i)$ with $\hat{\boldsymbol{x}}_i \in [-1, 1]^C$. 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 $\boldsymbol{x}$. Different domains result in different self-supervised quality measure distributions, which can then be compared by the earth mover's distance [59], 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 7×7, stride of 1, and 60 feature maps. Afterwards, 4 downsampling blocks follow, each consisting of a convolutional layer with kernel size 3×3, and a stride of two for spatial reduction of the $(120, 240, 480, 960)$ feature maps. The last convolutional layer has a kernel size 3×3, 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 3×3, 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 4×4, stride of two, and $(960, 480, 240, 120)$ feature maps. The architecture is finalized by a convolutional layer with kernel size 7×7, 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 4×4, stride of 2, and $(64, 128, 256, 512)$ feature maps. A final convolutional layer with kernel size 4×4, stride of one, one feature map, and ReLU activation delivers the discriminator outputs.

The autoencoder loss

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

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 $\boldsymbol{x}$ and the reconstruction $\hat{\boldsymbol{x}}$, and the generator-specific least-squares (LS) GAN loss $J^{\mathrm{G,adv}}$ [36]. The discriminator is trained with the discriminator-specific LS-GAN loss $J^{\mathrm{D,adv}}$, 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

$$mIoU = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \frac{TP_s}{TP_s + FP_s + FN_s} \quad (2)$$

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 $s \in \mathcal{S} = \{1, 2, ..., S\}$, 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 $\boldsymbol{x}'_i, \hat{\boldsymbol{x}}'_i \in [0, 255]^C$ usually is computed by the peak signal-to-noise ratio (PSNR), performing a direct MSE comparison of pixel values:

$$PSNR = 10 \log \left(\frac{(x'_{\text{max}})^2}{\frac{1}{C \cdot H \cdot W} \sum_{i \in \mathcal{I}} \left\| \boldsymbol{x}'_i - \hat{\boldsymbol{x}}'_i \right\|^2} \right) [\text{dB}] \quad (3)$$

with $x'_{\text{max}} = 255$.

The comparison of two discrete probability distributions $P(\mu)$, $\mu \in \mathcal{M} = \{1, 2, ..., M\}$ and $Q(\nu)$, $\nu \in \mathcal{N} = \{1, 2, ..., N\}$ 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 μ and ν with the $M \times N$ flow matrix $\boldsymbol{F} = (f_{\mu\nu})$ with $f_{\mu\nu} \in [0, 1]$ being the *flow* from bin μ to ν. The optimal flow is found by minimizing the work according to

$$\boldsymbol{F}^* = \arg\min_{\boldsymbol{F}} W(P, Q, \boldsymbol{F}) = \arg\min_{\boldsymbol{F}} \sum_{\mu \in \mathcal{M}} \sum_{\nu \in \mathcal{N}} f_{\mu\nu} d_{\mu\nu} \quad (4)$$

under consideration of the four (stochastic) constraints

$$\begin{aligned} f_{\mu\nu} &\geq 0, & \mu \in \mathcal{M}, \nu \in \mathcal{N} \\ \sum_{\nu \in \mathcal{N}} f_{\mu\nu} &\leq P(\mu), & \mu \in \mathcal{M} \\ \sum_{\mu \in \mathcal{M}} f_{\mu\nu} &\leq Q(\nu), & \nu \in \mathcal{N} \\ \sum_{\mu \in \mathcal{M}} \sum_{\nu \in \mathcal{N}} f_{\mu\nu} &= \min(P(\mu), Q(\nu)). \end{aligned}$$

We then obtain the earth-mover's distance as

$$DM(P, Q) = \frac{\sum\limits_{\mu \in \mathcal{M}} \sum\limits_{\nu \in \mathcal{N}} f^*_{\mu\nu} d_{\mu\nu}}{\sum\limits_{\mu \in \mathcal{M}} \sum\limits_{\nu \in \mathcal{N}} f^*_{\mu\nu}}, \quad (5)$$

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] $\tau = \tau_{\mathrm{b}}$, which accounts for ties in one quantity, whereby in the following we will omit the index b. Having K observations $\boldsymbol{o}_k = (a_k, b_k)$ with $k \in \{1, ..., K\}$, the total number of observation pairs

$$(\boldsymbol{o}_k, \boldsymbol{o}_\ell) = \big((a_k, b_k), (a_\ell, b_\ell)\big) \quad (6)$$

with $k < \ell$ is $n_{\mathrm{p}} = \binom{K}{2} = \frac{1}{2} K(K-1)$. 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 n_{a}, and ties in b n_{b} is used to calculate Kendall's rank order coefficient

$$\tau = \frac{n_{\mathrm{c}} - n_{\mathrm{d}}}{\sqrt{(n - n_{\mathrm{a}})(n - n_{\mathrm{b}})}} \in [-1, 1], \quad (7)$$

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

Trained on	Model	Measure	Evaluated on					Kendall τ
			CS_{train}	CS_{val}	BDD_{train}	BDD_{val}	KITTI	
CS_{train}	Autoencoder	PSNR	29.55 dB	28.24 dB	21.01 dB	21.26 dB	20.13 dB	0.6
	Segmentation	mIoU	81.2 %	66.7 %	23.1 %	26.7 %	51.1 %	
BDD_{train}	Autoencoder	PSNR	25.18 dB	25.13 dB	25.87 dB	25.37 dB	22.10 dB	0.8
	Segmentation	mIoU	45.5 %	43.9 %	53.8 %	49.0 %	44.1 %	

Table 1: Mean PSNR results for the autoencoder and mIoU results for the semantic segmentation trained and evaluated on various datasets.

KITTI [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_{train}), the Cityscapes validation set (CS_{val}), the Berkeley DeepDrive training set (BDD_{train}), the Berkeley DeepDrive validation set (BDD_{val}), and the KITTI set (which consists of all first images in the stereo training set of KITTI2015). CS_{train} and CS_{val} consists of 2,975 and 500 images, respectively, and are downsampled to 512×1024 pixels. BDD_{train} and BDD_{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_{train} or BDD_{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 $\alpha_1 = \frac{12}{23}$ for the MSE loss and $\alpha_2 = \frac{10}{23}$ for the feature matching loss. Furthermore, early stopping w.r.t. the PSNR on the validation set is applied.

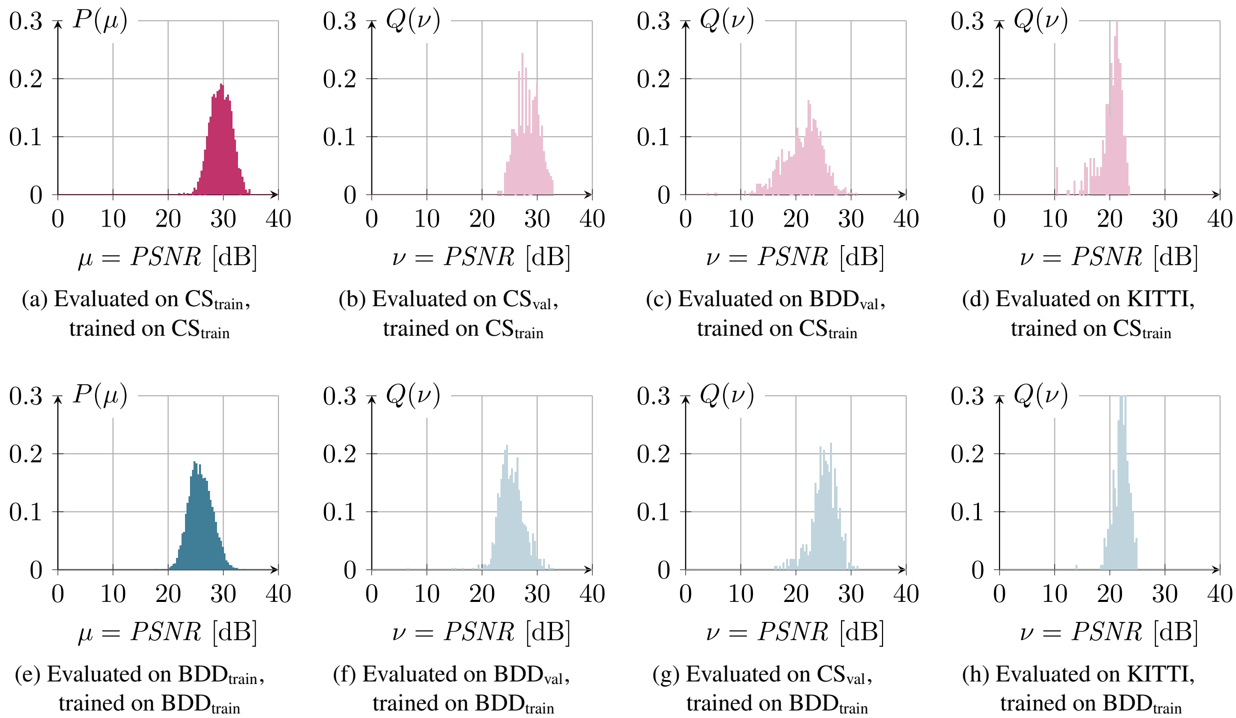
4.2. Domain-Specific Performance

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_{train} and BDD_{train} can be seen in Table 1. We also report Kendall's rank order coefficient τ (7), evaluating the degree of rank similarity of the PSNR and mIoU series.

For the CS_{train}-trained autoencoder, the PSNR performance is best on CS_{train} (obviously because it is the training set) and performs second best on CS_{val}, which is also plausible since it is the in-domain case. Evaluated on BDD_{train} and BDD_{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_{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_{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_{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_{train} and CS_{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_{train} and BDD_{val}, while CS_{train}, CS_{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 τ is 0.8, underlining the strong correlation of rank orders.

The models trained on CS_{train} and BDD_{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



(a) Evaluated on CS_{train}, trained on CS_{train}

(b) Evaluated on CS_{val}, trained on CS_{train}

(c) Evaluated on BDD_{val}, trained on CS_{train}

(d) Evaluated on KITTI, trained on CS_{train}

(e) Evaluated on BDD_{train}, trained on BDD_{train}

(f) Evaluated on BDD_{val}, trained on BDD_{train}

(g) Evaluated on CS_{val}, trained on BDD_{train}

(h) Evaluated on KITTI, trained on BDD_{train}

Figure 3: Histograms $P(\mu)$ (source domain) and $Q(\nu)$ (target domain), with μ, ν, representing autoencoder performance PSNRs with models trained and evaluated on different datasets. The upper histograms (red, 3a to 3d) stem from the autoencoder trained on CS_{train}, while the lower ones (blue, 3e to 3h) are trained on BDD_{train}.

necessarily kept in the low PSNR regime (BDD_{train}, BDD_{val}, KITTI) for models trained on CS_{train}, and (CS_{val}, KITTI) for models trained on BDD_{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 ΔmIoU series. The segmentation performance drop is simply stated as the mIoU difference between the training domains (CS_{train} and BDD_{train}, respectively) and the target domains.

In consideration of the results for the Cityscapes-trained models, the DM metric for the validation set (here: $1.31\,\mathrm{dB}$) 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_{train}, BDD_{val}, or KITTI, the autoencoder reconstruction performance dropped significantly, so our DM metric increased to $8\,\mathrm{dB}$ and more. In each of these cases also the drop in the mIoU is large, with ΔmIoU 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 $DM = 9.41\,\mathrm{dB}$. The pure rank orders in the DM metric and the ΔmIoU 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_{train}, the val-

Trained on	Reference	Model	Measure	Evaluated on					Kendall τ
				CS_{train}	CS_{val}	BDD_{train}	BDD_{val}	KITTI	
CS_{train}	CS_{train}	Autoencoder	DM	$0.0\,\mathrm{dB}$	$1.31\,\mathrm{dB}$	$8.53\,\mathrm{dB}$	$8.29\,\mathrm{dB}$	$9.41\,\mathrm{dB}$	0.6
		Segmentation	ΔmIoU	$0.0\,\%$	$14.5\,\%$	$58.1\,\%$	$54.5\,\%$	$30.1\,\%$	
BDD_{train}	BDD_{train}	Autoencoder	DM	$0.68\,\mathrm{dB}$	$0.74\,\mathrm{dB}$	$0.0\,\mathrm{dB}$	$0.51\,\mathrm{dB}$	$3.77\,\mathrm{dB}$	0.8
		Segmentation	ΔmIoU	$8.3\,\%$	$9.9\,\%$	$0.0\,\%$	$4.8\,\%$	$9.7\,\%$	

Table 2: Domain mismatch metric DM (5), absolute mIoU differences between the references (CS_{val} and BDD_{val}) and various datasets, and Kendall's rank order τ.

idation set domain shift of $0.51\,\mathrm{dB}$ is smaller than for Cityscapes, corresponding to an mIoU difference of $4.8\,\%$ to the training set. The domain mismatch estimate DM for both CS datasets is a bit higher as with BDD_{val}, so we assume that DM and ΔmIoU perform proportionally. And indeed, as the DM metric increases from $0.51\,\mathrm{dB}$ for BDD_{val} over $0.68\,\mathrm{dB}$ for CS_{train} to $0.74\,\mathrm{dB}$ for CS_{val}, also the ΔmIoU increases following the same rank order of datasets. Interestingly again, the DM metric for KITTI is highest (here: by far highest), which is appropriate for ΔmIoU being more than doubled w.r.t. the source validation set BDD_{val}. Due to the concordant rank order of the DM metric and the ΔmIoU in all but one cases, Kendall's rank order coefficient for the BDD-trained models is 0.8.

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 ΔmIoU. 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 $2 \times 1.31\,\mathrm{dB} = 2.62\,\mathrm{dB}$, excluding BDD_{train}, BDD_{val}, and KITTI from the functional scope (meaning these are clearly out-of-domain datasets!), and for the BDD-trained autoencoder the threshold is $2 \times 0.51\,\mathrm{dB} = 1.02\,\mathrm{dB}$, 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

[1] Johannes Abel, Magdalena Kaniewska, Cyril Guillaume, Wouter Tirry, and Tim Fingscheidt. Objective Assessment of Artificial Speech Bandwidth Extension Approaches. In *Proc. of 12. ITG Symposium Speech Communication*, pages 190–194, Paderborn, Germany, Oct. 2016. 4

[2] Eirikur Agustsson, Michael Tschannen, Fabian Mentzer, Radu Timofte, and Luc Van Gool. Generative Adversarial

Networks for Extreme Learned Image Compression. In *Proc. of ICCV*, pages 221–231, Seoul, Korea, Oct. 2019. 2, 3

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. In *Proc. of ICML*, pages 214–223, Sydney, Australia, Aug. 2017. 2

[4] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. In *Proc. of PAMI*, pages 2481–2495, Kharagpur, India, Oct. 2016. 2

[5] Andreas Bär, Fabian Hüger, Peter Schlicht, and Tim Fingscheidt. On the Robustness of Teacher-Student Frameworks for Semantic Segmentation. In *Proc. of CVPR - Workshops*, pages 1–9, Long Beach, CA, USA, June 2019. 2

[6] Jan-Aike Bolte, Andreas Bär, Daniel Lipinski, and Tim Fingscheidt. Towards Corner Case Detection for Autonomous Driving. In *Proc. of IV*, pages 366–373, Paris, France, June 2019. 2

[7] Jan-Aike Bolte, Markus Kamp, Antonia Breuer, Silviu Homoceanu, Peter Schlicht, Fabian Huger, Daniel Lipinski, and Tim Fingscheidt. Unsupervised Domain Adaptation to Improve Image Segmentation Quality Both in the Source and Target Domain. In *Proc. of CVPR - Workshops*, pages 1–10, Long Beach, CA, USA, June 2019. 2

[8] Karsten M. Borgwardt, Arthur Gretton, Malte J. Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J. Smola. Integrating Structured Biological Data by Kernel Maximum Mean Discrepancy. *Bioinformatics*, 22(14):e49–e57, July 2006. 2

[9] Gabriel J. Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic Object Classes in Video: A High-Definition Ground Truth Database. *Pattern Recognition Letters*, 30(2):88–97, Jan. 2009. 2

[10] Rich Caruana. Multitask Learning. *Machine Learning*, 28(1):41–75, July 1997. 2

[11] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Semantic Image Segmentation With Deep Convolutional Nets and Fully Connected CRFs. In *Proc. of ICLR*, pages 1–14, San Diego, CA, USA, May 2015. 2

[12] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking Atrous Convolution for Semantic Image Segmentation. *arXiv*, June 2017. (arXiv:1706.05587). 2

[13] Liang-Chieh Chen, Yi Yang, Jiang Wang, Wei Xu, and Alan L. Yuille. Attention to Scale: Scale-Aware Semantic Image Segmentation . In *Proc. of CVPR*, pages 1063–6919, Las Vegas, NV, USA, June 2016. 2

[14] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *Proc. of CVPR*, pages 3213–3223, Las Vegas, NV, USA, June 2016. 2, 4

[15] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2):303–338, Sept. 2010. 2

[16] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes Challenge: A Retrospective. *International Journal of Computer Vision*, 111(1):98–136, Jan. 2015. 2

[17] Yaroslav Ganin and Victor Lempitsky. Unsupervised Domain Adaptation by Backpropagation. In *Proc. of ICML*, pages 1180–1189, Lille, France, July 2015. 2

[18] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image Style Transfer Using Convolutional Neural Networks. In *Proc. of CVPR*, pages 2414–2423, Las Vegas, NV, USA, June 2016. 2

[19] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision Meets Robotics: The KITTI Dataset. *International Journal of Robotics Research (IJRR)*, 32(11):1231–1237, Aug. 2013. 2, 5

[20] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *Proc. of NIPS*, pages 2672–2680, Montréal, Canada, Dec. 2014. 2

[21] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773, Mar. 2012. 2

[22] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Deep Back-Projection Networks for Super-Resolution. In *Proc. of CVPR*, pages 1664–1673, Salt Lake City, UT, USA, June 2018. 2

[23] Justin Johnson and Alexandre Alahi and Li Fei-Fei. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In *Proc. of ECCV*, pages 694–711, Amsterdam, Netherlands, Oct. 2016. 3

[24] Kaiming He and Georgia Gkioxari and Piotr Dollár and Ross Girshick. Mask R-CNN. In *Proc. of ICCV*, pages 2980–2988, Venice, Italy, Oct. 2017. 2

[25] Tero Karras, Samuli Laine, and Timo Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. In *Proc. of CVPR*, pages 4401–4410, Long Beach, CA, USA, June 2019. 2

[26] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active Contour Models. *International Journal of Computer Vision*, 1(4):321–331, Jan. 1988. 2

[27] Maurice G. Kendall. The Treatment of Ties in Ranking Problems. *Biometrika*, 33(3):239–251, Nov. 1945. 4

[28] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *Proc. of ICLR*, pages 1–15, San Diego, CA, USA, May 2015. 5

[29] Chuan Li and Michael Wand. Precomputed Real-Time Texture Synthesis With Markovian Generative Adversarial Networks. In *Proc. of ECCV*, pages 702–716, Amsterdam, Netherlands, Oct. 2016. 2

[30] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature Pyramid Networks for Object Detection. In *Proc. of CVPR*, pages 2117–2125, Honolulu, HI, USA, July 2017. 2

[31] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence

Zitnick. Microsoft COCO: Common Objects in Context. In *Proc. of ECCV*, pages 740–755, Zurich, Switzerland, Sept. 2014. 2

[32] Jonas Löhdefink, Andreas Bär, Nico M. Schmidt, Fabian Hüger, Peter Schlicht, and Tim Fingscheidt. On Low-Bitrate Image Compression for Distributed Automotive Perception: Higher Peak SNR Does Not Mean Better Semantic Segmentation. In *Proc. of IV*, pages 352–359, Paris, France, June 2019. 2

[33] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation. In *Proc. of CVPR*, pages 3431–3440, Boston, MA, USA, June 2015. 2

[34] Pauline Luc, Camille Couprie, Soumith Chintala, and Jakob Verbeek. Semantic Segmentation using Adversarial Networks. In *NIPS Workshop on Adversarial Training*, pages 1–12, Barcelona, Spain, Dec. 2016. 2

[35] Chao Ma, Chih-Yuan Yang, Xiaokang Yang, and Ming-Hsuan Yang. Learning a No-Reference Quality Metric for Single-Image Super-Resolution. *Computer Vision and Image Understanding*, 158:1–16, May 2017. 2

[36] Xudong Mao, Qing Li, Haoran Xie, Raymond Yiu Keung Lau, Zhen Wang, and Stephen Paul Smolley. Least Squares Generative Adversarial Networks. In *Proc. of ICCV*, pages 2794–2802, Venice, Italy, Oct. 2017. 2, 4

[37] Xiaojiao Mao, Chunhua Shen, and Yu-Bin Yang. Image Restoration Using Very Deep Convolutional Encoder-Decoder Networks With Symmetric Skip Connections. In *Proc. of NIPS*, pages 2802–2810, Barcelona, Spain, Dec. 2016. 2

[38] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image Segmentation Using Deep Learning: A Survey. *arXiv*, Jan. 2020. (arXiv:2001.05566). 2

[39] Mehdi Mirza and Simon Osindero. Conditional Generative Adversarial Nets. *arXiv*, Nov. 2014. (arXiv:1411.1784). 2

[40] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning Deconvolution Network for Semantic Segmentation. In *Proc. of ICCV*, pages 1520–1528, Las Condes, Chile, Dec. 2015. 2

[41] Sinno Jialin Pan and Qiang Yang. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, Oct. 2010. 2

[42] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation. *arXiv*, June 2016. (arXiv:1606.02147). 2

[43] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic Differentiation in PyTorch. In *Proc. of NIPS - Workshops*, pages 1–4, Long Beach, CA, USA, Dec. 2017. 5

[44] Eduardo Romera, José M. Álvarez, Luis M. Bergasa, and Roberto Arroyo. ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation. *IEEE Transactions on Intelligent Transportation Systems*, 19(1):263–272, Jan. 2018. 2, 3

[45] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. The Synthia Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes. In *Proc. of CVPR*, pages 3234–3243, Las Vegas, NV, USA, June 2016. 2

[46] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, Dec. 2015. 5

[47] David Salomon. *Data Compression: The Complete Reference*. Springer Science & Business Media, 2004. 2

[48] Hossein Talebi and Peyman Milanfar. NIMA: Neural Image Assessment. *IEEE Transactions on Image Processing*, 27(8):3998–4011, Sept. 2018. 2

[49] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. Lossy Image Compression With Compressive Autoencoders. In *Proc. of ICLR*, pages 1–19, Toulon, France, Apr. 2017. 2

[50] Alexey Tsymbal. The Problem of Concept Drift: Definitions and Related Work. *Computer Science Department, Trinity College Dublin*, 106(2):58, Apr. 2004. 2

[51] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Improved Texture Networks: Maximizing Quality and Diversity in Feed-Forward Stylization and Texture Synthesis. In *Proc. of CVPR*, pages 6924–6932, Honulu, HI, USA, July 2017. 2

[52] Hemanth Demakethepalli Venkateswara, Shayok Chakraborty, and Sethuraman Panchanathan. Deep-Learning Systems for Domain Adaptation in Computer Vision: Learning Transferable Feature Representations. *IEEE Signal Processing Magazine*, 34(6):117–129, Nov. 2017. 2

[53] Francesco Visin, Marco Ciccone, Adriana Romero, Kyle Kastner, Kyunghyun Cho, Yoshua Bengio, Matteo Matteucci, and Aaron Courville. Reseg: A Recurrent Neural Network-Based Model for Semantic Segmentation. In *Proc. of CVPR - Workshops*, pages 41–48, Las Vegas, NV, USA, June 2016. 2

[54] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-Resolution Image Synthesis and Semantic Manipulation With Conditional GANs. In *Proc. of CVPR*, pages 8798–8807, Salt Lake City, UT, USA, June 2018. 3

[55] Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, and Eero P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, Apr. 2004. 2

[56] Zhou Wang, Eero Simoncelli, and Alan Bovik. Multi-Scale Structural Similarity for Image Quality Assessment. In *Proc. of ACSSC*, pages 1398–1402, Pacific Grove, CA, USA, Nov. 2003. 2

[57] Gerhard Widmer and Miroslav Kubat. Learning in the Presence of Concept Drift and Hidden Contexts. *Machine Learning*, 23(1):69–101, Apr. 1996. 2

[58] Raymond A. Yeh, Chen Chen, Teck Yian Lim, Alexander G. Schwing, Mark Hasegawa-Johnson, and Minh N. Do. Semantic Image Inpainting With Deep Generative Models. In *Proc. of CVPR*, pages 5485–5493, Honolulu, HI, USA, July 2017. 2

[59] Yossi Rubner and Carlo Tomasi and Leonidas J. Guibas. The Earth Mover's Distance as a Metric for Image Retrieval. *International Journal of Computer Vision*, 40(2):99–121, Nov. 2000. 1, 2, 3, 4

[60] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. BDD100K: A Diverse Driving Video Database With Scalable Annotation Tooling. *arXiv*, Aug. 2018. (arXiv:1805.04687). 2, 4

[61] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. Generative Image Inpainting With Contextual Attention. In *Proc. of CVPR*, pages 5505–5514, Salt Lake City, UT, USA, June 2018. 2

[62] Matthew D. Zeiler and Rob Fergus. Visualizing and Understanding Convolutional Networks. In *Proc. of ECCV*, pages 818–833, Zurich, Switzerland, Sept. 2014. 2

[63] Matthew D. Zeiler, Dilip Krishnan, Graham W. Taylor, and Rob Fergus. Deconvolutional Networks. In *Proc. of CVPR*, pages 2528–2535, San Francisco, CA, USA, June 2010. 2