An Unsupervised Temporal Consistency (TC) Loss to Improve the Performance of Semantic Segmentation Networks

Serin Varghese[1,3] Sharat Gujamagadi[1] Marvin Klingner[3] Nikhil Kapoor[1] Andreas Bär[3]
Jan David Schneider[1] Kira Maag[2] Peter Schlicht[1] Fabian Hüger[1] Tim Fingscheidt[3]
{john.serin.varghese, sharat.gujamagadi, nikhil.kapoor,
jan.david.schneider, peter.schlicht, fabian.hueger}@volkswagen.de
{kmaag}@uni-wuppertal.de {s.varghese, m.klingner, andreas.baer, t.fingscheidt}@tu-bs.de

[1]Volkswagen Group Automation [2]University of Wuppertal [3]Technische Universität Braunschweig

Abstract

Deep neural networks (DNNs) for highly automated driving are often trained on a large and diverse dataset, and evaluation metrics are reported usually on a per-frame basis. However, when evaluated on video sequences, the predictions are often unstable between consecutive frames. As such unstable predictions over time can lead to severe safety consequences, there is a growing need to understand, evaluate, and improve the temporal consistency of DNNs. In this paper, we explore such a temporal characteristic and propose a novel unsupervised temporal consistency (TC) loss that penalizes unstable semantic segmentation predictions. This loss function is used in a two-stage training scheme to jointly optimize for both, accuracy of semantic segmentation predictions, and its temporal consistency based on video sequences. We demonstrate that our training strategy helps in improving the temporal consistency of two state-of-the-art semantic segmentation networks on two different road-scenes datasets. We report an absolute 4.25% improvement in the mean temporal consistency (mTC) of the HRNetV2 network and an absolute 2.78% improvement on the DeepLabv3+ network, both evaluated on the Cityscapes dataset, with only a slight decrease in accuracy. When evaluating on the same video sequences using a synthetic dataset Sim KI-A, we show absolute improvements in both, accuracy (2.19% mIoU) and temporal consistency (0.21% mTC) for the DeepLabv3+ network. We confirm similar improvements for the HRNetV2 network.

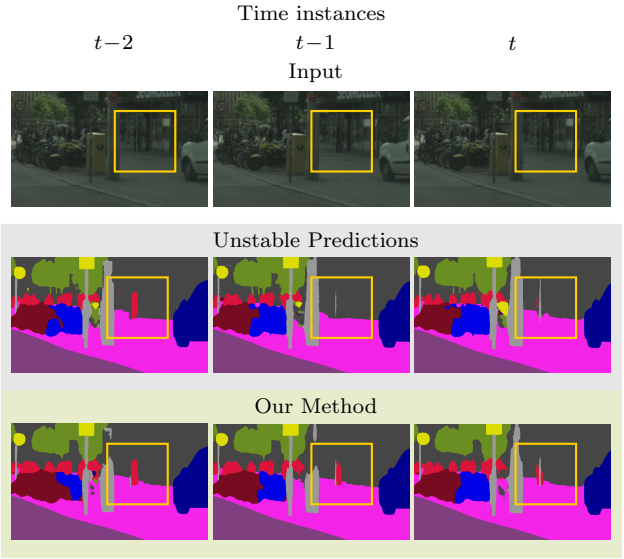


Figure 1: **Examples of stable and unstable predictions** of semantic segmentation networks on image sequences. The yellow boxes highlight the segmentation area of interest in the frames. *Top*: Left to right are the frames at discrete time instances $t-2$, $t-1$, and t of a video that is fed into a semantic segmentation network. *Center*: The predictions of the pedestrian and the pavement are unstable, i.e., they are not consistent across time (both $t-2 \rightarrow t-1$, and $t-1 \rightarrow t$). *Bottom*: Our approach focuses on improving temporal consistency of predictions of semantic segmentation networks over time. With our approach we observe that the pedestrian is clearly detected in each time instance.

1. Introduction

Deep neural networks (DNNs) have shown great advances in the fields of bio-imaging, pose detection, and

other computer vision tasks, expanding their application in use cases such as highly automated driving.

Approaches using convolutional neural networks (CNNs) achieve state-of-the-art performance in tasks, such as, object detection, semantic segmentation, and maneuver detection.

These DNNs are usually trained on a large diverse dataset. This diversity in the dataset is to ensure their generalization to varying objects, scenarios and environmental conditions [23]. The variation in the training set may include variation in weather such as rain, fog, and clear weather conditions, variations in types to street scenes such as inner city and highway scenes, etc. The DNNs are trained on such a dataset by minimizing some given loss function on the training set. They perform well if the test domain is similar to the training domain and when evaluation is performed on a *frame* basis. When using DNNs in real-world applications, however, the inputs are usually image *sequences* (video) from a camera, and there are, therefore, additional characteristics of the network that have to be evaluated to ensure robust predictions. Not only is it important for the predictions of the neural network to be accurate, but also these predictions should be *stable* over time. We refer the term *stability* to denote network predictions that do not fluctuate over time. A typical example of an unstable prediction is shown in Figure 1, where a pedestrian seems to disappear in subsequent frames, even though the input frames remain mostly similar. We argue that temporal stability of such DNNs could be an important safety-relevant criteria, that needs to be investigated, especially for environment perception networks used in highly automated driving.

In this paper, we explore the aforementioned property of network prediction stability, specifically for semantic segmentation networks. We attempt to solve the problem of temporally consistent predictions by a learning task with an additional fine-tuning step. Firstly, we introduce a novel temporal consistency (TC) loss function that penalizes unstable semantic segmentation predictions and we evaluate a two-stage training strategy to jointly optimize for both, accuracy of semantic segmentation predictions (frame-based evaluation), and its temporal consistency (sequence-based evaluation). Second, we compare our novel object-based stability approach against a prior art pixel-based stability approach [16] and show on the `Cityscapes` dataset [8] that our method outperforms the prior art. Finally, we extend our approach to a synthetic dataset, `Sim KI-A`, to evaluate accuracy and stability on labelled sequences, which, to the best of our knowledge, has not been done in any previous temporal consistency investigation. We show improvements in *both*, mean intersection-over-union (mIoU) and mean temporal consistency (mTC) [22], when evaluated on the same test sequences.

The paper is structured as follows: Section 2 reviews the relevant related work. In Section 3, we describe our intuition and explain our proposed loss to enforce the temporal consistency of semantic segmentation predictions. In Section 4, we describe the datasets, semantic segmentation networks, and optical flow algorithms that we have used in our experiments. In Section 5, we present results and provide discussions. Finally, we conclude the work in Section 6.

2. Related Work

In this section, we introduce the related work in two fields, namely evaluation metrics for semantic segmentation, and methods to improve stability of the predictions of semantic segmentation networks.

Evaluation metrics for semantic segmentation give a quantitative measurement for comparing and tracking the performance of the networks. These are classified broadly into two categories based on two evaluation criteria [21] namely on the one side the accuracy, or in other words, the success of the network, and on the other side the computational complexity in terms of speed and memory requirements. For measuring the accuracy of the segmentation, pixel accuracy [1] and variants of the Jaccard index [9] are the most popular metrics for evaluating semantic segmentation networks. The Jaccard index, also known as mean intersection-over-union (mIoU), is

$$\text{mIoU} = \frac{1}{S} \sum_{s \in \mathcal{S}} \frac{\text{TP}_s}{\text{TP}_s + \text{FP}_s + \text{FN}_s}, \tag{1}$$

where TP_s, FP_s and FN_s are the class-specific true positives, false positives, and false negatives, respectively, and $\mathcal{S} = \{1, 2, \ldots, S\}$, where S is the number of semantic classes in the dataset. For calculating the computational complexity, execution time (in seconds), number of floating point operations (FLOPs), and peak memory usage are generally reported [14].

Furthermore, other than the criteria mentioned above, we envision a criterion involving the robustness of the network. This robustness can be defined with respect to changing domains [17], changes in weather conditions [11], and the influence of adversarial perturbations [2, 3]. For evaluating the robustness of the networks to such corruptions in the input image, metrics such as mean performance under corruptions (mPC) [11] have been recently proposed. Additionally, to evaluate the robustness to temporal changes in the input [15], metrics such as mean temporal consistency (mTC) [22] have been published. The instantaneous temporal consistency TC_t at time t is defined as

$$\text{TC}_t = \text{mIoU}(\mathbf{m}_t, \tilde{\mathbf{m}}_t), \tag{2}$$

while the mean temporal consistency mTC is defined as

$$\text{mTC} = \frac{1}{T-1}\sum_{t=2}^{T}\text{TC}_t, \tag{3}$$

where T is the number of frames, $\mathbf{m}_t$ is the semantic segmentation prediction at time t, and the expected prediction $\tilde{\mathbf{m}}_t$ is computed based on the prediction of the network at time t–1 and the movement of the pixel between time t–1 and t.

In this work, two different evaluation metrics are employed: The accuracy-based metric mean intersection-over-union [9], and secondly, the mean temporal consistency *metric* [22], based on the influence of temporal changes in the input.

Methods to improve the temporal consistency of networks have also been investigated recently. Label propagation [1, 5, 7, 10, 18, 25] techniques have been proposed generating an additional pseudo-labelled dataset based on the video frames and optical flow estimations. Zhu *et al*. [25] introduced a modification in the class label space that allows for predicting multiple classes at the boundary of objects. This, along with the combined label-image propagation technique, helped in improving the performance of the network. As these methods only aim for improvement in the performance of networks by increasing the available training data, they are not directly comparable to our method. For improving the robustness of the neural network to adversarial attacks, Saemann *et al*. [20] proposed a strategy for warping feature maps based on their entropy-based confidence of predictions. Although the authors show an increase in the accuracy or robustness of the network with this network modification, the effect on the temporal consistency of the networks is largely unstudied. We also acknowledge some parallel work by Liu *et al*. [16], where a loss function

$$J^{\text{TL}}(\mathbf{x}_t, \mathbf{x}_{t+1}) = \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} V_{t\to t+1,i}\cdot||\mathbf{y}_{t,i} - \mathbf{y}_{t+1\to t,i}||_2^2, \tag{4}$$

is introduced to enforce stability within a teacher-student learning environment. Here, $\mathbf{x}_t \in \mathbb{G}^{H\times W\times C}$ is an image with height H, width W, number of color channels C, at time instance t, and $\mathbb{G} = \{0, 1, 2, \ldots, 255\}$. The pixel position i of the input image $\mathbf{x}$ is defined as $i \in \mathcal{I} = \{1, \ldots, H\cdot W\}$, and $|\mathcal{I}| = H\cdot W$. The image $\mathbf{x}_t$ is passed as an input through the neural network $\mathbf{F}(\mathbf{x}_t, \boldsymbol{\theta})$, with $\boldsymbol{\theta}$ being its network parameters. The class scores $\mathbf{y}_t$ are defined as

$$\mathbf{y}_t = \mathbf{F}(\mathbf{x}_t, \boldsymbol{\theta}) \in \mathbb{I}^{H\times W\times S}, \tag{5}$$

where $\mathbb{I} = [0, 1]$. Each element in $\mathbf{y}_t$ can be understood as a posterior probability $y_{t,i,s}(\mathbf{x})$ for the class $s \in \mathcal{S} = \{1, 2, \ldots, S\}$ at pixel position i of the input image. The *warped* class scores $\mathbf{y}_{t+1\to t}$ at time instance t are computed by warping the class scores $\mathbf{y}_{t+1}$ from time instance $t+1$ to time t. To do this, the pixel-wise displacements between input images $\mathbf{x}_{t+1}$ and $\mathbf{x}_t$ computed by `FlowNet2` [12] are used. The occlusion mask $\mathbf{V}_{t\to t+1} \in \mathbb{R}^{H\times W}$ is defined pixel-wise as $V_{t\to t+1,i} = \exp(-||\mathbf{x}_{t,i} - \mathbf{x}_{t+1,i}||_1)$, where $\mathbf{x}_{t,i} \in \mathbb{G}^C$ is the three-dimensional color pixel. On a closer inspection of the J^{TL} loss (4), we observe that the loss inherently penalizes the pixel-wise instability in the class probabilities $\mathbf{y}_{t,i}$ and $\mathbf{y}_{t+1\to t,i}$ of the individual pixels i.

In this work, we introduce and compare a novel object-based stability approach against the pixel-based stability approach by Liu *et al*. and show that our method performs consistently better. Additionally, our approach does not require any modification to the architecture of the semantic segmentation network and is unsupervised, i.e., requiring only unlabelled, sequential video frames in a second training stage.

3. Method

In a video sequence, assuming a sufficiently high frame rate, there is only a continuous and therefore limited movement of objects across frames. Therefore, it is highly unlikely for objects to be present in one frame, then be absent in the next frame, and abruptly be present again in the next. Temporally consistent (also called *stable*) predictions of semantic segmentation networks means that movement of detected objects is also limited. To correct errors due to ego motion, i.e., moving objects and actual missing objects, we warp the prediction of the semantic segmentation model from the current frame to the previous frame based on optical flow calculations. The error between semantic segmentation predictions can be used as an optimization objective to improve the stability of the segmentation predictions. In this section, we describe the idea of our proposed *temporal consistency (TC) loss* and present details of our method.

3.1. Supervised Semantic Segmentation Training

The supervised training involves a loss function that penalizes the incorrect predictions of the network when compared with ground truth labels. This supervised training is necessary for ensuring the accuracy of the semantic segmentation is maintained in the second step of the training process. The segmentation mask $\mathbf{m}_t = (m_{t,i}) \in \mathcal{S}^{H\times W}$ of the network prediction at time t consists of elements

$$m_{t,i} = \arg\max_{s\in\mathcal{S}} y_{t,i,s}, \tag{6}$$

where a class $s \in \mathcal{S}$ is assigned to each pixel i in the class score $\mathbf{y}_t$. Let $\overline{\mathbf{m}}_t \in \mathcal{S}^{H\times W}$ be the labelled ground truth in the dataset $\overline{\mathcal{M}}$ corresponding to image $\mathbf{x}_t$, having the

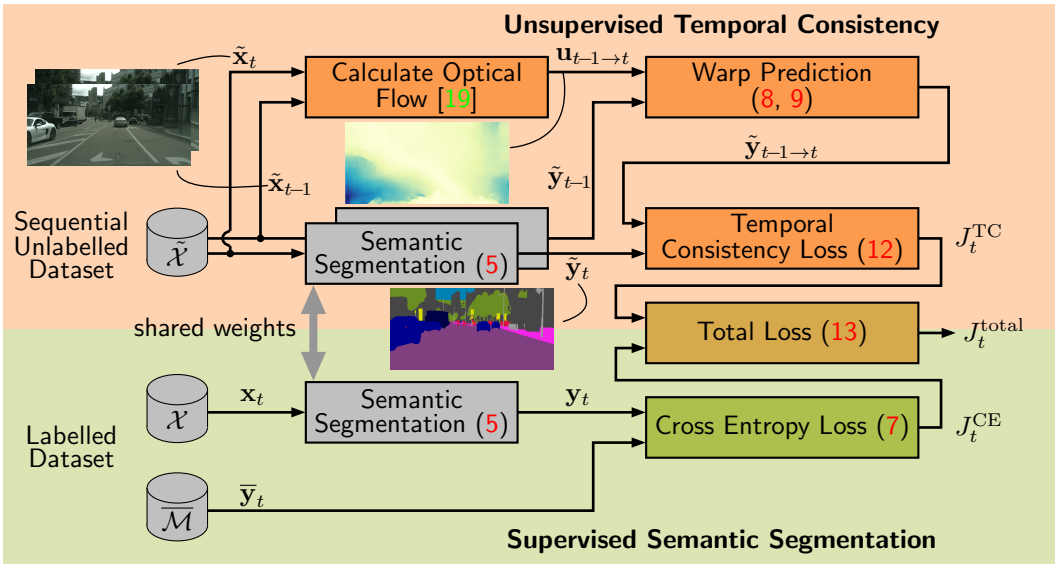


Figure 2: Overview of our **training strategy enforcing temporal consistency in a self-supervised fashion** for a semantic segmentation network. For calculating our temporal consistency (TC) loss, a pair of sequential frames $(\tilde{\mathbf{x}}_{t-1}, \tilde{\mathbf{x}}_t)$ is passed to the semantic segmentation network. Based on the optical flow $\mathbf{u}_{t-1\to t}$ [19] calculated between the input frames, the prediction of the semantic segmentation network at time $t-1$ is warped (9) to time t, yielding $\tilde{\mathbf{y}}_{t-1\to t}$. The temporal consistency loss J_t^{TC} (12) is calculated between this warped prediction $\tilde{\mathbf{y}}_{t-1\to t}$ and the prediction $\tilde{\mathbf{y}}_t$ at time t. Along with the cross-entropy loss J_t^{CE} (7), it is used in the training process.

same dimensions as the segmentation mask $\mathbf{m}_t$. Likewise, $\overline{\mathbf{y}}_t \in \{0,1\}^{H\times W\times S}$ is the one-hot encoded vector ground truth in three-dimensional tensor format. For supervised training, we optimized the network using the *cross-entropy* (CE) loss (see Figure 2) between the posterior probabilities of the network $\mathbf{y}_t$ and the labels $\overline{\mathbf{y}}_t$. Taking the mean over all pixels, the loss function for the image's posterior probabilities $y_{t,i,s} \in \mathbb{I}$ is defined as

$$J_t^{\mathrm{CE}} = -\frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}}\sum_{s\in\mathcal{S}} w_s \overline{y}_{t,i,s}\cdot\log(y_{t,i,s}), \qquad (7)$$

where $|\mathcal{I}| = H\cdot W$ is the number of pixels, and w_s are the weights assigned to each class during training as in [24].

3.2. Unsupervised Temporal Consistency (TC) Loss

We define a sequential and unlabelled dataset $\tilde{\mathcal{X}}$ with video sequences $\tilde{\mathbf{x}}_1^T = (\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \ldots, \tilde{\mathbf{x}}_t, \ldots, \tilde{\mathbf{x}}_T)$ containing image frames $\tilde{\mathbf{x}}_t$ at discrete time instants $t \in \mathcal{T} = \{1, 2, \ldots, T\}$. We use optical flow functions to capture the notion of network prediction stability, and to estimate the apparent motion within the video sequence. Optical flow estimates the displacement of each pixel between the consecutive frames $\tilde{\mathbf{x}}_{t-1}$ and $\tilde{\mathbf{x}}_t$. Following [19], the optical flow computed between $\tilde{\mathbf{x}}_{t-1}$ and $\tilde{\mathbf{x}}_t$ is defined as a tensor $\mathbf{u}_{t-1\to t} \in \mathcal{U}^{H\times W}$, where $\mathcal{U}$ is the set of two-dimensional pixel-wise displacements $\Delta h, \Delta w \in \mathbb{R}$, representing the coordinate-wise shift of each pixel from $\tilde{\mathbf{x}}_{t-1}$ to $\tilde{\mathbf{x}}_t$.

Using the optical flow tensor $\mathbf{u}_{t-1\to t}$ generated by the optical flow block in Figure 2, the prediction of the semantic segmentation network $\tilde{\mathbf{y}}_{t-1}$ is warped from time $t-1$ to time t. To do this, we first define pixel coordinates for an image as tensor $\mathbf{p} \in \mathcal{P}^{H\times W}$, where $\mathcal{P} = (h, w)$ is an index pair with $h \in \{1, \ldots, H\}$ and $w \in \{1, \ldots, W\}$. Tensor $\mathbf{p}$ thus only contains the pixel-wise coordinates of a pixel in an image and does not carry any information about pixel intensity values. Now we can add the pixel-wise displacement vectors $\mathbf{u}_{t-1\to t}$ to the original pixel positions $\mathbf{p}_{t-1}$ to receive a tensor

$$\mathbf{p}_{t-1\to t} = \mathbf{p}_{t-1} + \mathbf{u}_{t-1\to t}, \qquad (8)$$

which provides the projected pixel coordinates $\mathbf{p}_{t-1\to t} \in \mathcal{U}^{H\times W}$. Subsequently, we have to shift the segmentation output $\tilde{\mathbf{y}}_{t-1}$ to pixel positions $\mathbf{p}_{t-1\to t}$. As the pixel coordinates $\mathbf{p}_{t-1\to t}$ are non-integer numbers, we use nearest neighbour sampling $\mathrm{nearest}()$ as described by [13] to obtain valid integer coordinates in a grid-like structure as in $\mathbf{p}_t$. For the mapping of $\tilde{\mathbf{y}}_{t-1}$ to the flow-based estimate $\tilde{\mathbf{y}}_{t-1\to t}$ we thereby obtain

$$\tilde{\mathbf{y}}_{t-1\to t} = \mathrm{nearest}(\tilde{\mathbf{y}}_{t-1}, \mathbf{p}_{t-1\to t}). \qquad (9)$$

Accordingly, $\tilde{\mathbf{y}}_{t-1\to t}$ is the *expected* prediction at time t based on the optical flow, conditioned on the change in the pair of inputs $\tilde{\mathbf{x}}_{t-1}$ and $\tilde{\mathbf{x}}_t$, which compensates for the movement of the camera and the objects in the consecutive frames.

Ideally, for a good semantic segmentation model, the distance between the network output $\tilde{\mathbf{y}}_t$ and the prediction based on the optical flow $\tilde{\mathbf{y}}_{t-1\to t}$ should be small. To enforce this, we borrow from the temporal consistency *metric* interpretation in [22], defining temporal consistency as the mean intersection-over-union (mIoU) [9] of the two predictions $\tilde{\mathbf{y}}_t$ and $\tilde{\mathbf{y}}_{t-1\to t}$. As per definition, the mIoU between the segmentation masks $\tilde{\mathbf{m}}_t$ and $\tilde{\mathbf{m}}_{t-1\to t}$ is given as

$$\text{mIoU}(\tilde{\mathbf{m}}_{t-1\to t}, \tilde{\mathbf{m}}_t) = \frac{1}{|S|} \sum_{s\in\mathcal{S}} \frac{\text{TP}_{t,s}}{\text{TP}_{t,s} + \text{FP}_{t,s} + \text{FN}_{t,s}}, \tag{10}$$

where $\text{TP}_{t,s}$, $\text{FP}_{t,s}$ and $\text{FN}_{t,s}$ are the class-specific true positives, false positives and false negatives, respectively, which are calculated for $\tilde{\mathbf{m}}_t$, considering $\tilde{\mathbf{m}}_{t-1\to t}$ as reference. A value of $\text{mIoU}(\tilde{\mathbf{m}}_{t-1\to t}, \tilde{\mathbf{m}}_t) = 1$ indicates that both overlap perfectly and the prediction of the network is completely stable. However, the mIoU metric cannot be optimized by gradient descent as set operations are non-differentiable. To be able to still use (10) as part of a loss function, we follow the suggestions by Berman *et al*. [4] to approximate (10) to ensure differentiability for gradient descent. An approximation of the mIoU (10) is made using class probabilities $\tilde{\mathbf{y}}_{t-1\to t}$ and $\tilde{\mathbf{y}}_t$ and this approximation is given by

$$\widetilde{\text{mIoU}}_t = \frac{1}{S} \sum_{s\in\mathcal{S}} \frac{\sum_{i\in\mathcal{I}} |y'_{t,s,i} \cdot \tilde{y}_{t,s,i}|}{\sum_{i\in\mathcal{I}} |y'_{t,s,i} + \tilde{y}_{t,s,i} - (y'_{t,s,i} \cdot \tilde{y}_{t,s,i})|}, \tag{11}$$

where $y'_{t,s,i} = \tilde{y}_{t-1\to t,s,i}$ and $\widetilde{\text{mIoU}}_t = \widetilde{\text{mIoU}}(\tilde{\mathbf{y}}_{t-1\to t}, \tilde{\mathbf{y}}_t)$. The *temporal consistency* (TC) loss between the prediction at time t and the warped prediction from time $t-1$ (see Figure 2) is then defined as

$$J_t^{\text{TC}} = 1 - \widetilde{\text{mIoU}}_t. \tag{12}$$

Note that since $\widetilde{\text{mIoU}}_t$ is larger for stable predictions, the loss needs to be minimized. The temporal consistency loss J_t^{TC} therefore enforces the stability of the predictions of a semantic segmentation model by motion flow calculations, given sequential images in a self-supervised manner, i.e, without requiring labels.

3.3. Total Loss

For the total loss, we finally combine both the losses during the second-stage training, i.e., the fine-tuning process. The cross-entropy loss and the temporal consistency loss are combined as

$$J_t^{\text{total}} = (1-\alpha) J_t^{\text{CE}} + \alpha J_t^{\text{TC}}, \tag{13}$$

where α is the loss weight that controls the influence of the individual losses, and J_t^{CE} is computed on dataset $\mathcal{X}$, while J_t^{TC} is computed on dataset $\tilde{\mathcal{X}}$.

Table 1: **Details of the datasets** used in the experiments. The image resolution of the dataset images and split into training, validation and test sets are described. The training set of the unlabelled dataset is used for calculating J_t^{TC} (12), and the test set is used for evaluating the mTC (3) metric for the semantic segmentation networks. A star ($\star$) indicates availability of ground truth labels in the synthetic dataset.

Dataset	Resolution	Labelled $\mathcal{X}$			Unlabelled $\tilde{\mathcal{X}}$		
		$\mathcal{X}^{\text{train}}$	$\mathcal{X}^{\text{val}}$	$\mathcal{X}^{\text{test}}$	$\tilde{\mathcal{X}}^{\text{train}}$	$\tilde{\mathcal{X}}^{\text{val}}$	$\tilde{\mathcal{X}}^{\text{test}}$
`Cityscapes`	2048 × 1048	2,975	500	1,525	2,300	399	200
`Sim KI-A`	1920 × 1080	4,547	386	386	1,276*	111*	111*

4. Experimental Setup

In this section, we initially describe the road-scenes datasets that have been used in our experiments. Next, we present the implementation details of the semantic segmentation networks that we employ.

4.1. Datasets

In the following, we will briefly give an overview of the `Cityscapes` dataset [8][1] and the `Sim KI-A`[2] dataset, focusing on the availability, split and resolution of the images and labels. This overview is given in Table 1.

For **`Cityscapes`**, the baseline networks are trained with 2,975 images of the training set $\mathcal{X}^{\text{train}}$. A total of 500 images $\mathcal{X}^{\text{val}}$ and 1,525 images $\mathcal{X}^{\text{test}}$ are used for validation and testing, respectively. For the calculation of J_t^{TC} (12) we make use of the unlabelled image sequences $\tilde{\mathcal{X}}$ from Stuttgart that are provided by `Cityscapes`. For our experiments, we use the `stuttgart_01` (1,100 frames) and `stuttgart_02` (1,200 frames) sequences for fine-tuning, i.e., $\tilde{\mathcal{X}}^{\text{train}}$. We divide the `stuttgart_00` (599 frames) sequence into a validation sequential set (frames 0-399) $\tilde{\mathcal{X}}^{\text{val}}$, and test sequential set (frames 400-599) $\tilde{\mathcal{X}}^{\text{test}}$. These frames are sampled by a video camera at a frame rate of 17 Hz. We chose the `Cityscapes` dataset due to diverse and highly dynamic objects present in road scenes. `Cityscapes` is also a widely used and accepted benchmark for semantic segmentation in general.

The **`Sim KI-A`** dataset is a synthetic dataset with 4,257 images $\mathcal{X}^{\text{train}}$, 387 images $\mathcal{X}^{\text{val}}$, and 387 images $\mathcal{X}^{\text{test}}$. The images in the `Sim KI-A` dataset are taken from video sequences sampled from various camera angles with a frame rate of 4 Hz. For calculating the loss J_t^{TC}(12) and evaluating the mTC (3), we use higher frame rate sequences $\tilde{\mathcal{X}}$ to enable more accurate optical flow estimations. The frames are sampled from a sequence containing car-mounted camera angles at a frame rate of 12 Hz. We split such a higher frame rate sequence $\tilde{\mathcal{X}}$

[1] https://www.cityscapes-dataset.com/downloads/

[2] This dataset will shortly be available online.

into 1,276 training images $\tilde{\mathcal{X}}^{\text{train}}$ for calculating (12). A total of 111 images each are used as validation ($\tilde{\mathcal{X}}^{\text{val}}$) and test ($\tilde{\mathcal{X}}^{\text{test}}$) set, respectively. The dominant characteristic of any synthetic dataset is the inherent availability of ground truth labels and additional meta-information regarding the perception environment. In addition, the sequential and labelled sequences allow for evaluation of accuracy and stability on the same test scenario (i.e., sequence), unlike `Cityscapes`.

Optical Flow Algorithms are used for calculating the pixel movement between consecutive sequential frames. The large displacement optical flow (LDOF) method from Narayanan *et al.* [19][3] is used in this work. For LDOF, a sub-sampling parameter $r \in \mathbb{N}$ defines the granularity of the optical flow estimation. A value of $r = N$, indicates that every N^{th} pixel movement is estimated. The missing pixels values, if any, are then interpolated. For all our experiments, we have used r=1, indicating dense estimation of the pixel movements, without introducing any interpolation errors. Based on a comparison between the optical flow algorithms conducted in [22] for a temporal consistency *metric*, we identified and used the neural network-based optical flow estimation `FlowNet2` [12][4] for our ablation study.

4.2. Semantic Segmentation Networks

In the following, we briefly introduce the two semantic segmentation networks, namely `HRNetv2` [24][5] and `DeepLabv3+` [6][6], that we have used for our experiments. We also provide the training parameters of the networks.

Table 2 shows an overview of the baseline performance of these networks on the aforementioned datasets. The `HRNetv2` baseline is trained on the `Cityscapes` dataset with an image resolution of 1024 × 512 and a batch size of 4 on a single `Nvidia GTX 1080 Ti` GPU. The network is trained with the class frequency-weighted cross-entropy loss J_t^{CE} (7) and optimized using SGD. An initial learning rate of 0.01 and a polynomial decay with the momentum of 0.9 are used. The network is trained to convergence for 484 epochs following [24]. Data augmentation by image flipping and multi-scale cropping is applied during the training of the baseline. A similar setting of parameters is used for training the `HRNetv2` baseline on the `Sim KI-A` dataset with image crops of 901 × 901. The network is trained to convergence for 100 epochs.

The `DeepLabv3+` baseline with `Resnet-101` backbone is trained on the `Cityscapes` dataset with image crops of 513 × 513. The network is trained with the J_t^{CE} (7) loss and optimized using SGD. An initial learning rate of 0.01 and a polynomial decay with the momentum of 0.9 are used. The network is trained to convergence for 240 epochs. Data augmentations of image flipping and multi-scale cropping are applied during the training of the baseline. A similar setting of parameters is used for training the `DeepLabv3+` baseline on the `Sim KI-A` dataset with image crops of 901 × 901. The network is trained to convergence for 50 epochs.

[3] http://lmb.informatik.uni-freiburg.de/resources/binaries/
[4] https://github.com/lmb-freiburg/flownet2-docker
[5] https://github.com/HRNet/HRNet-Semantic-Segmentation
[6] https://github.com/jfzhang95/pytorch-deeplab-xception

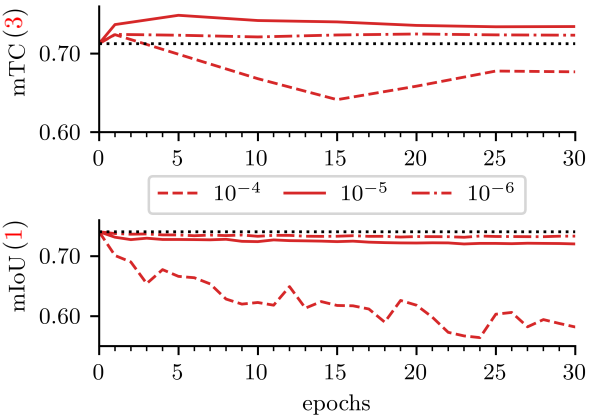
5. Simulation Results and Discussions

In this section, we evaluate our proposed training methodology from Section 3 with the datasets and networks from Section 4. We evaluate our J_t^{TC} loss (12) along with a cross-entropy loss J_t^{CE} (7) in a two-stage learning setup. In the first stage, we train the baseline network on the J_t^{CE} (7) loss alone until convergence. Thereafter in the second stage, we add an additional J_t^{TC} (12) loss and fine-tune the network using the total loss J_t^{total} (13). We randomly sample mixed batches from $\mathcal{X}$ and $\tilde{\mathcal{X}}$ to compute J_t^{CE} (7) and J_t^{TC} (12), respectively. With our initial experiments, we observed that with respect to convergence, such a two-stage training approach is better than training from scratch. In this paper, we therefore focus on only the two-stage training approach in more detail.

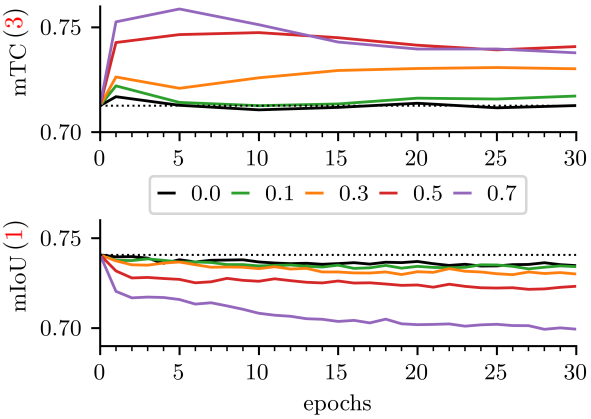
Figure 3 shows a plot of the effect a hyperparameter tuning (learning rate, and loss weight α) has on the `HRNetv2`'s accuracy and temporal consistency, when trained on `Cityscapes`. The mIoU is calculated on the validation set $\mathcal{X}^{\text{val}}$ and the mTC is calculated on the sequential validation set $\tilde{\mathcal{X}}^{\text{val}}$.

Effect of learning rate: We observe that the learning rate plays a significant role in such a two-stage learning setup. We vary the learning rate by keeping all the other parameters constant. The parameter α, that controls the contribution of the TC loss within the total loss (13), is kept constant at α=0.5. From Figure 3a, we observe that a higher learning rate showed a larger drop in the mIoU, and higher fluctuations in the mTC. A learning rate of 10^{-5} helps in improving the temporal consistency of the network and retains the accuracy of the semantic segmentation predictions. We observe similar behaviour of the learning rate on `HRNetv2` network trained on `Sim KI-A` dataset, and also with the `DeepLabv3+` network.

Effect of loss weight α: We further investigate the effect of the parameter α (13) on the fine-tuning process. A loss weight α=0 indicates that only J_t^{CE} (7) is used and α=1 indicates that only J_t^{TC} (12) is used. In Figure 3b, we vary α and we keep the learning rate constant at 10^{-5}, which gave the best results in the previous experiments. We observe that a higher α negatively affects the cross-entropy loss and the mIoU on the validation set. We also perform the experiment with α=0 excluding the effect of J_t^{TC}. A lower α>0 helps in improving the temporal consistency, while not adversely affecting the accuracy of the semantic



(a) **Effect of learning rate**: Here, we vary only the learning rate while keeping the other parameters constant, with $\alpha = 0.5$ being chosen.



(b) **Effect of loss weight** α: Here, we vary only the loss weight α, keeping the learning rate constant at 10^{-5}.

Figure 3: **Effect of hyperparameters** for our two-stage training approach for the `HRNetv2` network trained on the `Cityscapes` dataset. The mIoU (1) is calculated on the validation set $\mathcal{X}^{\text{val}}$ of `Cityscapes` and the mTC (3) is calculated on the `stuttgart_00` unlabelled sequential validation set, $\tilde{\mathcal{X}}^{\text{val}}$. The black dotted line indicates the performance of the baseline network.

segmentation predictions. We observe that α=0.5 shows a large gain in mTC, with marginal decrease in mIoU. Similar to the learning rate hyperparameter above, we observe similar behaviour of the loss weight parameter α on the `HRNetv2` network trained on the `Sim KI-A` dataset, and also with the `DeepLabv3+` network.

Observations on a real dataset: Table 2 shows a summary of the results of evaluating the mTC (3) on the `Cityscapes` test set $\tilde{\mathcal{X}}^{\text{test}}$. The sequential test set $\tilde{\mathcal{X}}^{\text{test}}$ is only used for evaluation of mTC, and is not used in the training and the fine-tuning steps. The best hyperparameters for evaluation are chosen based on tuning the parameters (α=1, learning rate=10^{-5}, see Figure 3). In comparison to the `HRNETv2` baseline, we observe that the mean temporal consistency (mTC) is improved on the unlabelled, sequential test sequences $\tilde{\mathcal{X}}^{\text{test}}$ by absolute 4.25 %, with a drop of 1.33% mIoU on the labelled validation set $\mathcal{X}^{\text{val}}$. For the `DeepLabv3+` network, we observe an improvement of 2.78 % mTC on $\tilde{\mathcal{X}}^{\text{test}}$, with a drop of around 6 % mIoU on the labelled validation set $\mathcal{X}^{\text{val}}$. Furthermore, we observe that our approach is consistently better than the approach suggested by Liu *et al.* [16], in terms of improving temporal consistency, but falls behind in mIoU. It is important to note, however, that for `Cityscapes` we are not able to report mIoU and mTC on the same *sequential* test dataset, due to the lack of labelled videos in `Cityscapes`. It is, however, intuitive that the mIoU should improve with the improvement in mTC.

Observations on a synthetic dataset: To investigate this further, we make use of our synthetic dataset, `Sim KI-A`, that allows for calculating the mIoU and mTC on the *same sequential* test set $\tilde{\mathcal{X}}^{\text{test}}$, due to the inherent availability of labels in the synthetic dataset. Table 2 also shows a summary of the results on the `Sim KI-A` dataset. In comparison to the `HRNETv2` baseline, we observe that

Table 2: **Test set evaluation** based on mIoU (1) and mTC (3) for the baseline models and the models fine-tuned with the TC loss. We also compare this against the pixel-based approach J^{TL} (4). The test set was used neither during training/fine-tuning, nor during hyperparameter tuning. Best numbers in **bold**.

Dataset	Method	$\mathcal{X}^{\text{val}}$ mIoU [%]	$\tilde{\mathcal{X}}^{\text{val}}$ mTC [%]	$\tilde{\mathcal{X}}^{\text{test}}$ mTC [%]
`Cityscapes`	`HRNetv2` [24]	**74.07**	71.29	64.05
	with J^{TL} [16] (4)	73.34	72.56	64.10
	with J^{TC} (ours) (12)	72.74	**74.74**	**68.30**
	`DeepLabv3+` [6]	**69.06**	70.05	65.08
	with J^{TL} [16] (4)	64.99	71.78	65.71
	with J^{TC} (ours) (12)	62.20	**72.05**	**67.86**
Dataset	Method	$\mathcal{X}^{\text{val}}$ mIoU [%]	$\tilde{\mathcal{X}}^{\text{test}}$ mIoU [%]	$\tilde{\mathcal{X}}^{\text{test}}$ mTC [%]
`Sim KI-A`	`HRNetv2` [24]	85.05	50.66	76.04
	with J^{TL} [16] (4)	84.36	50.28	74.93
	with J^{TC} (ours) (12)	**85.23**	**51.32**	**76.38**
	`DeepLabv3+` [6]	80.44	52.26	83.16
	with J^{TL} [16] (4)	81.34	53.69	80.21
	with J^{TC} (ours) (12)	**84.52**	**54.45**	**83.37**

Table 3: **Ablation study for optical flow** on the HRNetv2 network, trained on the Cityscapes dataset. We only vary the choice of optical flow algorithm keeping all other training parameters constant. A lower MSE indicates higher accuracy of optical flow estimation. The - indicates that optical flow is not used in the baseline training. Best results are highlighted in **bold**.

Method	MSE	$\mathcal{X}^{\text{val}}$	$\tilde{\mathcal{X}}^{\text{val}}$	$\tilde{\mathcal{X}}^{\text{test}}$
		mIoU [%]	mTC [%]	mTC [%]
HRNetv2 [24]	-	**74.07**	71.29	64.05
with FlowNet2 [12]	0.0031	72.46	73.62	67.86
with LDOF [19]	0.0016	72.74	**74.74**	**68.30**

our J^{TC} loss improves mTC on the test sequences $\tilde{\mathcal{X}}^{\text{test}}$ by absolute 0.34 %. Here, we also observe an improvement of absolute 0.66 % in mIoU when calculated on the same sequential test set $\tilde{\mathcal{X}}^{\text{test}}$. For the DeepLabv3+ network, we observe an improvement of 0.21 % mTC on $\tilde{\mathcal{X}}^{\text{test}}$, and an improvement of absolute 2.19 % in mIoU when calculated on the same sequential test set $\tilde{\mathcal{X}}^{\text{test}}$. With the Sim KI-A dataset, we observe that we are always better than the approach by Liu *et al*. [16], in terms of *both* mIoU and mTC.

Ablation study on the effect of optical flow: We can observe from our training strategy (see Figure 2), that the results of the computation of J_t^{TC} (12) are dependent on the accuracy of the optical flow algorithm. To better understand this dependency, we perform an ablation study on the choice of optical flow algorithm. From Table 2, we observe that although we excel the baseline network and the state-of-the-art method [16] with respect to temporal consistency, the improvements in mTC (3) of both the networks (HRNetv2 and DeepLabv3+) on the Sim KI-A dataset are not as large as their respective improvements with the Cityscapes dataset. Cityscapes sequences are sampled at a higher frequency (17 Hz) in comparison to Sim KI-A sequences (12 Hz), which means a lower error in optical flow estimations for the Cityscapes dataset. This raises the question: *Is the difference in the effectiveness of our strategy to enforce temporal consistency in DNNs due the optical flow estimation error? Does a lower optical flow estimation error lead to better results?* To investigate these questions, we fine-tune the HRNetv2 network on the Cityscapes dataset using two different optical flow algorithms, LDOF [19] and FlowNet2 [12]. The results are shown in Table 3. The mean-squared error

$$\text{MSE}(\tilde{\mathbf{x}}_t, \tilde{\mathbf{x}}_{t-1\to t}) = \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}}(\tilde{\mathbf{x}}_{t,i} - \tilde{\mathbf{x}}_{t-1\to t,i})^2 \qquad (14)$$

between the image $\tilde{\mathbf{x}}_t$ at time t and the expected image $\tilde{\mathbf{x}}_{t-1\to t}$ is calculated based on the optical flow estimation between images $\tilde{\mathbf{x}}_{t-1}$ and $\tilde{\mathbf{x}}_t$. We report a larger improvement in both mIoU (1) and mTC (3), when using LDOF [19], which has lower MSE in comparison to FlowNet2 [12]. An unlabelled sequence with a higher frame rate will naturally have a lower MSE, and will further aid our proposed method. We are currently restricted by the unavailability of higher frame rate sequences in road-scenes datasets to report an optimal frame rate. The authors envision the use of synthetic datasets for such an evaluation of optimal frame rate, as possible future work.

6. Conclusions

As deep neural networks for highly automated driving are trained and evaluated on a per-frame basis, they often times lead to temporally unstable predictions. In this paper, we investigated the temporal consistency of semantic segmentation networks in more detail. We proposed and formulated the problem of temporal consistency of predictions as an additional fine-tuning step on unlabelled image sequences. We introduced a novel temporal consistency (TC) loss function that penalizes unstable semantic segmentation predictions and then evaluated a two-stage training strategy to jointly optimize for both, accuracy of semantic segmentation predictions, and its temporal consistency based on video sequences. Using our proposed method, we report improvements of temporal consistency of two state-of-the-art semantic segmentation models on two different datasets. In comparison to the existing state-of-the-art, we show that our method provides the better temporal consistency. We reported an absolute 4.25% improvement in the temporal consistency (mTC) of the HRNetV2 model and an absolute 2.78% on the DeepLabv3+ model on Cityscapes videos with some decrease in the accuracy (mIoU) of the network on isolated images. For the first time, we extend such a temporal consistency investigation to a synthetic dataset to enable accuracy and temporal consistency evaluations on the same sequential test set. With the synthetic dataset, we show that our method improves both, accuracy and temporal consistency of the semantic segmentation networks, and excels state of the art consistency improvement methods also both in mIoU and mTC.

References

[1] Vijay Badrinarayanan, Fabio Galasso, and Roberto Cipolla. Label Propagation in Video Sequences. In *Proc. of CVPR*, pages 3265–3272, San Francisco, CA, USA, June 2010. 2, 3

[2] Andreas Bär, Marvin Klingner, Serin Varghese, Fabian Hüger, Peter Schlicht, and Tim Fingscheidt. Robust Semantic Segmentation by Redundant Networks With a Layer-Specific Loss Contribution and Majority Vote. In *Proc. of CVPR - Workshops*, pages 1348–1358, Seattle, WA, USA, June 2020. 2

[3] Andreas Bär, Jonas Löhdefink, Nikhil Kapoor, Serin Varghese, Fabian Hüger, Peter Schlicht, and Tim Fingscheidt. The Vulnerability of Semantic Segmentation Networks to Adversarial Attacks in Autonomous Driving: Enhancing Extensive Environment Sensing. *IEEE Signal Processing Magazine*, 38(1):42–52, Jan. 2021. 2

[4] Maxim Berman, Amal Triki, and Matthew Blaschko. The Lovasz-Softmax Loss: A Tractable Surrogate for the Optimization of the Intersection-Over-Union Measure in Neural Networks. In *Proc. of CVPR*, pages 4413–4421, Salt Lake City, UT, USA, June 2018. 5

[5] Ignas Budvytis, Patrick Sauer, Thomas Roddick, Kesar Breen, and Roberto Cipolla. Large Scale Labelled Video Data Augmentation for Semantic Segmentation in Driving Scenarios. In *Proc. of ICCV*, pages 230–237, Venice, Italy, Oct. 2017. 3

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. DeepLab: Semantic Image Segmentation With Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, Apr. 2018. 6, 7

[7] Po-Yi Chen, Alexander H. Liu, Yen-Cheng Liu, and Yu-Chiang Frank Wang. Towards Scene Understanding: Unsupervised Monocular Depth Estimation With Semantic-Aware Representation. In *Proc. of CVPR*, pages 2624–2632, Long Beach, CA, USA, June 2019. 3

[8] 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, 5

[9] Mark Everingham, S. M. Eslami, Luc Gool, Christopher K. Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes Challenge: A Retrospective. *International Journal of Computer Vision (IJCV)*, 111(1):98–136, Jan. 2015. 2, 3, 5

[10] Raghudeep Gadde, Varun Jampani, and Peter V. Gehler. Semantic Video CNNs Through Representation Warping. In *Proc. of ICCV*, pages 4463–4472, Venice, Italy, Oct. 2017. 3

[11] Dan Hendrycks and Thomas Dietterich. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In *Proc. of ICLR*, pages 1–15, New Orleans, LA, USA, May 2019. 2

[12] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. FlowNet 2.0: Evolution of Optical Flow Estimation With Deep Networks. In *Proc. of CVPR*, pages 2704–2713, Honulu, HI, USA, July 2017. 3, 6, 8

[13] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial Transformer Networks. In *Proc. of NIPS*, pages 2017–2025, Montréal, QC, Canada, Dec. 2015. 4

[14] Hyeji Kim, Muhammad Umar Karim Khan, and Chong-Min Kyung. Efficient Neural Network Compression. In *Proc. of CVPR*, pages 12561–12569, Prague, Czech Republic, Feb. 2019. 2

[15] Abhijit Kundu, Vibhav Vineet, and Vladlen Koltun. Feature Space Optimization for Semantic Video Segmentation. In *Proc. of CVPR*, pages 3168–3175, Las Vegas, NV, USA, June 2016. 2

[16] Yifan Liu, Chunhua Shen, Changqian Yu, and Jingdong Wang. Efficient Semantic Video Segmentation With Per-Frame Inference. In *Proc. of ECCV*, pages 1–22, Glasgow, UK, Aug. 2020. 2, 3, 7, 8

[17] Jonas Löhdefink, Justin Fehrling, Marvin Klingner, Fabian Hüger, Peter Schlicht, Nico M. Schmidt, and Tim Fingscheidt. Self-Supervised Domain Mismatch Estimation for Autonomous Perception. In *Proc. of CVPR - Workshops*, pages 1359–1368, Seattle, WA, USA, June 2020. 2

[18] Siva Karthik Mustikovela, Michael Ying Yang, and Carsten Rother. Can Ground Truth Label Propagation from Video Help Semantic Segmentation? In *Proc. of ECCV - Workshops*, pages 804–820, Amsterdam, Netherlands, Oct. 2016. 3

[19] Sundaram Narayanan, Brox Thomas, and Keutzer Kurt. Dense Point Trajectories by GPU-Accelerated Large Displacement Optical Flow. In *Proc. of ECCV*, pages 438–451, Heraklion, Greece, Sept. 2010. 4, 6, 8

[20] Timo Sämann, Karl Amende, Stefan Milz, and Horst-Michael Groß. Robust Semantic Video Segmentation Through Confidence-Based Feature Map Warping. In *Proc. of CSCS*, pages 1–9, Kaiserslautern, Germany, Oct. 2019. 3

[21] Irem Ulku and Erdem Akagunduz. A Survey On Deep Learning-based Architectures for Semantic Segmentation on 2D Images. *arXiv*, (1912.10230), May 2020. 2

[22] Serin Varghese, Yasin Bayzidi, Andreas Bär, Nikhil Kapoor, Sounak Lahiri, Jan David Schneider, Nico Schmidt, Peter Schlicht, Fabian Hüger, and Tim Fingscheidt. Unsupervised Temporal Consistency Metric for Video Segmentation in Highly-Automated Driving. In *Proc. of CVPR - Workshops*, pages 336–337, Seattle, WA, USA, June 2020. 2, 3, 5, 6

[23] Riccardo Volpi, Hongseok Namkoong, O. Sener, John C. Duchi, Vittorio Murino, and S. Savarese. Generalizing to Unseen Domains via Adversarial Data Augmentation. In *Proc. of NeurIPS*, pages 5339—-5349, Montréal, QC, Canada, Dec. 2018. 2

[24] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep High-Resolution Representation Learning for Visual Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (42):1–23, Aug. 2020. 4, 6, 7, 8

[25] Yi Zhu, Karan Sapra, Fitsum A. Reda, Kevin J. Shih, Shawn Newsam, Andrew Tao, and Brya Catanzaro. Improving Semantic Segmentation via Video Propagation and Label Relaxation. In *Proc. of CVPR*, pages 8856–8865, Long Beach, CA, USA, June 2019. 3