applied sciences

MDPI

Article

Unsupervised Semantic Segmentation Inpainting Network Using a Generative Adversarial Network with Preprocessing

Woo-Jin Ahn [1], Dong-Won Kim [2,*], Tae-Koo Kang [3], Dong-Sung Pae [4] and Myo-Taeg Lim [1,*]

[1] School of Electrical Engineering, Korea University, Seoul 02841, Republic of Korea
[2] Department of Digital Electronics, Inha Technical College, Incheon 22212, Republic of Korea
[3] Department of Human Intelligence and Robot Engineering, Sangmyung University, Cheonan 31066, Republic of Korea
[4] Department of Software, Sangmyung University, Cheonan 31066, Republic of Korea
* Correspondence: dwnkim@inhatc.ac.kr (D.-W.K.); mlim@korea.ac.kr (M.-T.L.)

Abstract: The generative adversarial neural network has shown a novel result in the image generation area. However, applying it to a semantic segmentation inpainting task exhibits instability due to the different data distribution. To solve this problem, we propose an unsupervised semantic segmentation inpainting method using an adversarial deep neural network with a newly introduced preprocessing method and loss function. For stabilizing the adversarial training for semantic segmentation inpainting, we match the probability distribution of the segmentation maps with the developed preprocessing method. In addition, a new cross-entropy total variation loss for the probability map is introduced to improve the segmentation inpainting work by smoothing the segmentation map. The experimental results demonstrate the proposed algorithm's effectiveness on both synthetic and real datasets.

Keywords: deeplearning; convolutional neural network; data preprocessing; binary total variation

check for updates

Citation: Ahn, W.-J.; Kim, D.-W.; Kang, T.-K.; Pae, D.-S.; Lim, M.-T. Unsupervised Semantic Segmentation Inpainting Network Using a Generative Adversarial Network with Preprocessing. *Appl. Sci.* **2023**, *13*, 781. https://doi.org/10.3390/app13020781

Academic Editor: Andrea Prati

Received: 8 December 2022
Revised: 26 December 2022
Accepted: 30 December 2022
Published: 5 January 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In autonomous driving systems, understanding the vehicle's surroundings and free-space information [1,2] is essential in driving the vehicle safely to its destination. Thus, it is crucial to have predefined free-space maps which do not include dynamic objects (i.e., cars, pedestrians, and bicycles) [3]. Semantic segmentation is one way to draw free-space maps, which have become widely used due to an improved performance with advances in fully convolutional neural network (CNN) approaches [4–7].

However, it is challenging to obtain a fully free-space segmentation map directly from the real world, where dynamic objects are ubiquitous. As illustrated in Figure 1, the static map contains only static objects, and the dynamic map includes both static and dynamic objects. Therefore, an unsupervised method of deleting dynamic objects from the segmentation map and treating them as static objects is required to obtain a free-space map from the real world (Figure 2).

In this paper, we propose an unsupervised semantic segmentation inpainting method that replaces dynamic segmentation labels with static labels in situations where the data are unpaired. We employ a generative adversarial network (GAN) [8] for the unsupervised semantic segmentation inpainting and introduce a preprocessing method and loss function to solve the instability of the GAN on the segmentation map. Our main contributions are summarized as follows:

- We introduce a preprocessing method that matches the maxima of the distributions of the predicted free-map and ground-truth map. This prevents the discriminator from learning the probability distribution and rather focuses on the segmentation inpainting, allowing the generator to learn how to inpaint the masked part.

- We propose a new cross-entropy total variation loss term to facilitate the generator network to smooth the nearby static segmentation. The proposed term is a denoising loss function for binary data, which removes the noise caused by the instability of the unsupervised learning process.
- The experimental results on the Media Integration and Communication Center Semantic Road Inpainting (MICC-SRI) [9] and Cityscapes [10] datasets validate the effectiveness of the proposed unsupervised semantic segmentation inpainting framework.

Section 2 introduces the related works of inpainting on an image and segmentation map. Section 3 describes the proposed unsupervised segmentation inpainting method. Section 4 introduces the MICC-SRI and Cityscapes datasets, experimental settings, and the outcomes of the proposed framework. Lastly, Section 5 summarizes the overall research of this paper.

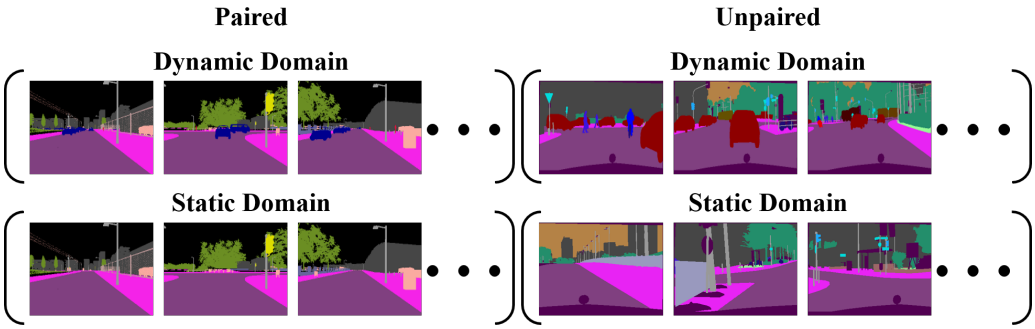


Figure 1. Paired static and dynamic maps for synthetic data (**left**). Unpaired static and dynamic maps for real-world data (**right**).

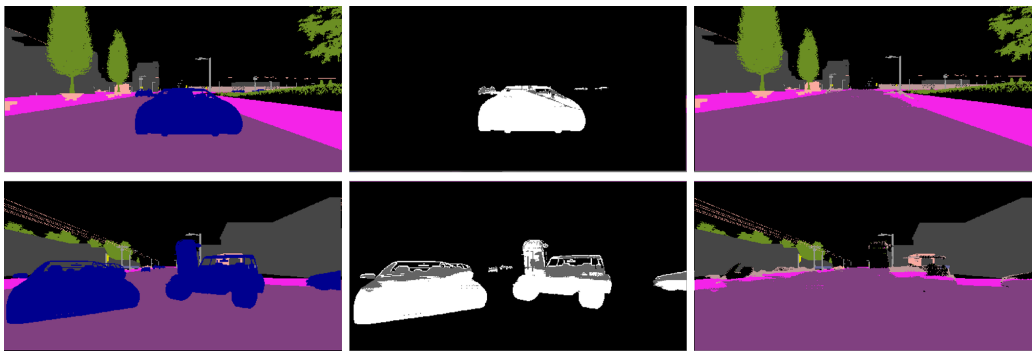


Figure 2. Removing the dynamic objects at the segmentation map using the purposed method: **left**: input (dynamic segmentation map); **center**: dynamic object mask; **right**: output (inpainted segmentation result).

2. Related Works

2.1. Image Inpainting

Image completion, or removing unwanted objects from the image, is a well-researched topic with various approaches. An early diffusion-based inpainting approach [11] propagates the local features from general information around the targeted area. However, it is only efficient when filling in a small missing area. In addition, one of the more traditional approaches is the patch-based image method [12] which borrows the most fitting patches from nearby pixels using the gradients and distance metrics of the image. Patch-based methods demonstrate filling a plausible texture that can fill large missing regions in images, but the method depends on low-level features unsuitable for filling missing areas for complicated structures. Recently, methods based on a CNN were also used for image completion to tackle the problem of generating complicated structural features to fill in missing regions of a structured scene [13]. Mao et al. [14] filled the plausible features of the image but generated blurry images for a large missing region. Pathak et al. [15]

adopted adversarial training to inpaint the missing region with high-order image features. However, this method is only trained on a fixed square area. To alleviate this problem, Iizuka et al. [16] proposed an upgraded network architecture by employing local and global discriminators that improve the result's global consistency and local textures. Moreover, Yu et al. [17] proposed an improved method with a coarse-to-fine network and the contextual attention module. It first fills the missing region roughly in the coarse stage and calculates the similarity between the result and background. The network completes the work with additional details using similarity information in the fine stage. Saharia et al. [18] introduced a condition diffusion-based method for an inpainting method by suggesting a new loss function. Even with the existing state-of-art image inpainting methods, these primarily focus on completing the RGB image data.

2.2. Supervised Segmentation Inpainting

For the segmentation inpainting method, Song et al. [19] proposed an inpainting method using the segmentation map for guidance. It predicts the segmentation map (SP-Net) and restores the missing region with segmentation guidance (SG-Net). Simple adversarial training for the segmentation map is used in the SP-Net stage based on supervised learning. Berlincioni et al. [9] also used supervised inpainting on the segmentation map with the MICC-SRI dataset based on the contextual attention model [17] to remove dynamic objects from the segmentation map. These segmentation inpainting methods revealed novel results on supervised learning, but when unsupervised, they exhibit instability.

2.3. Unsupervised Segmentation Inpainting

Related to the unsupervised problem, the GAN [8,20,21] revealed a novel result in image generation. A GAN consists of a generator and discriminator and trains the network in the adversarial process. This allows the generator to learn higher-order features and generates unseen realistic images [22]. Moreover, it is widely used in various areas, such as object removal in images [14–17,19,23,24] and translating images to different domains [25,26], which helps reduce costly acquisitions from different environments [27,28].

However, directly applying a GAN to the segmentation generation tends to fail due to the difference in probability distributions between the ground-truth map and the predicted map. In detail, the generator creates a probability segmentation map from 0 to 1 from the softmax activation function, whereas the ground-truth segmentation map contains only two data points, 0 or 1. The discriminator learns the most specific feature between the two domains, the data distribution difference. The discriminator becomes dominant in discriminating the ground truth from the predicted map, breaking the Nash equilibrium and ending the GAN training. When training a semantic segmentation map, GAN-based adversarial training, the training strategy needs to be different from the existing image methods.

Recently, the GAN-based segmentation method was studied in various research areas with additional conditions to match the data distribution (e.g., semantic segmentation [29], semi-supervised learning [30], and domain adaptation [27]). Luc et al. [29] introduced a preprocessing method called scaling for adversarial segmentation training. It changes the distribution of the ground-truth label similarly to the network output by setting the least mass, limiting the minimum value of the representative, and matching the distribution of the network result. It was proven that preprocessing the labels by matching the distribution increases the performance of adversarial training. However, the least mass limitation overfits the discriminator at the initial learning when the generator does not have the confidence of each label. Hung et al. [30] added a term for matching the distribution of the network to the ground-truth label for semi-supervised learning for semantic segmentation. The semi-supervised loss term targets the self-binarized loss function with the cross-entropy term and makes the distribution look like the ground-truth label. Vu et al. [27] used an adversarial entropy minimization term for domain adaptation in semantic segmentation. Similarly, the network output distribution targets the ground-truth label with the self-

entropy term, which helps stabilize the non-representative label, unlike the self-binarize method. Despite the promising semantic segmentation domain adaptation results, the data polarization reduces the backpropagation, which slows the learning.

3. Proposed Method

3.1. Overall Framework

As shown in Figure 3, our approach consists of a semantic segmentation inpainting network (SSIN) and discriminator network. The inpainting network SSIN is placed to replace the dynamic labels of the segmentation map with the static label, and the discriminator attempts to differentiate the predicted segmentation map from the ground-truth map. Before feeding the raw input of the segmentation map to the discriminator, preprocessing is applied to both maps, which matches the probability distribution of the ground-truth label and inpainted map.

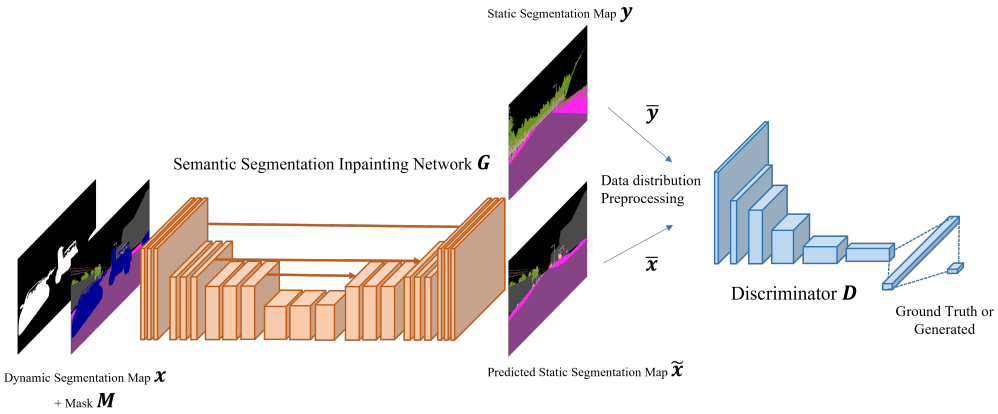


Figure 3. Overview of our architecture for the semantic segmentation inpainting network (SSIN).

Proposed SSIN $G(x, M)$ takes x (the dynamic segmentation map) and M (the mask) as input to replace the dynamic label with a static label. The dynamic segmentation map x includes both dynamic and static objects with a probability of $x \in \{0, 1\}$ and its shape is H×W×C, where W is the width of the map, H is the height, and C is the number of segmentation classes. The binary mask image M separates the dynamic and static objects (1 for a static object and 0 for a dynamic object) in the shape of H×W×1. The output of the SSIN $\tilde{x}$ has the same shape as the input, where dynamic objects are replaced with nearby static objects.

The discriminator network $D(x)$ takes the segmentation map and discriminates whether the segmentation map is generated or real with the output size of 1. In detail, the input of the discriminator is either the preprocessed SSIN output map $\bar{x}$ or the preprocessed static map $\bar{y}$. While the discriminator focuses on discerning the domain of the segmentation map with the preprocessed maps, the generator aims to deceive the discriminator and learns how to remove and change the dynamic class to the static class similar to the given ground-truth data.

3.2. Preprocessing for Probability Map Distribution Matching

The SSIN output probability map $\tilde{x}$ and the ground-truth static probability map y have a noticeable difference in the probability distribution. More specifically, $\tilde{x}$ has a data distribution from 0 to 1 because the last layer activation function in the SSIN is the softmax function which presents the probability of each class. On the other hand, one-hot encoded y includes only two numbers, 1 representing the label and 0 for the other non-representing labels.

The purpose of the discriminator is to learn and compete with the generator for the adversarial unsupervised training of the inpainting segmentation map. However, the SSIN output $\tilde{x}$ and the static map y have definite differences in data distributions. Such mismatch

bothers the discriminator from learning structural features for the segmentation map and consequently lowers the performance of SSIN, concentrating on learning how to match the data distribution. Therefore, it is important to match the probability distribution of $\widetilde{\boldsymbol{x}}$ and $\boldsymbol{y}$ to make the SSIN produce static label maps that cannot be distinguished from ground-truth maps.

We introduce a new preprocessing method for the probability map by matching the probability distribution. Our basic idea is to match the distribution of $\boldsymbol{y}$ to $\widetilde{\boldsymbol{x}}$ by copying both the maximum value of $\widetilde{\boldsymbol{x}}$ and the non-representing values to $\boldsymbol{y}$. In detail, we limit the $\widetilde{\boldsymbol{x}}$ with a maximum value and fix the summation probability to 1 for each pixel. Then, by swapping the maximum position as $\boldsymbol{y}$, we obtain preprocessed label map $\overline{\boldsymbol{y}}$.

To prevent the representative label from having a maximum value of 1 and make the other classes have a zero value, τ is set to limit the maximum value $\widetilde{\boldsymbol{x}}$, and the residuals are dispersed partially to other non-maximum classes to fool the discriminator about the probability distribution.

Mathematically, by setting an upper bound τ for the SSIN output tensor $\widetilde{\boldsymbol{x}}$, the values of non-true labels are shifted close to one, and backpropagation becomes active. The output tensor $\widetilde{\boldsymbol{x}}$ is clipped with a maximum value of τ as in (1):

$$\widetilde{\boldsymbol{x}}_{\boldsymbol{min}} = \min(\tau, \widetilde{\boldsymbol{x}}), \tag{1}$$

where each pixel-wise component of $\widetilde{\boldsymbol{x}}$ has upper limitation of τ. This hyperparameter value is set near 1 to limit only the representative components. The summation of the label map per pixel should equal 1 because it represents the probability map. Therefore, the residual should be distributed to other class labels after clipping the segmentation map. Before spreading the residual to other nonrepresentative classes, the residual value can be defined as follows in (2):

$$\widetilde{x}_{res} = \widetilde{\boldsymbol{x}}_{\boldsymbol{il}} - (\widetilde{\boldsymbol{x}}_{\boldsymbol{min}})_{\boldsymbol{il}}, \tag{2}$$

where $\boldsymbol{i} \in N^{H \times W}$implies the spatial position, given its representative label $\boldsymbol{l}$, where $\widetilde{x}_{res}$ is a constant residual value. If the maximum value of $\widetilde{\boldsymbol{x}}_{\boldsymbol{i}}$ is smaller than τ, $\widetilde{x}_{res}$ becomes 0. After calculating the residual, it spreads proportionally, which means a larger label value has residuals, and a smaller label value has fewer. The normalized residual vector $\widetilde{\boldsymbol{x}}_{\boldsymbol{i,norm}}$ can be expressed as shown in (3):

$$\widetilde{\boldsymbol{x}}_{\boldsymbol{i,norm}} = \frac{\widetilde{\boldsymbol{x}}_{\boldsymbol{res}}}{1 - \widetilde{\boldsymbol{x}}_{\boldsymbol{il}}}(\widetilde{\boldsymbol{x}}_{\boldsymbol{res}} - \widetilde{\boldsymbol{x}}_{\boldsymbol{res}} \odot \widetilde{\boldsymbol{x}}_{\boldsymbol{i,oh}}), \tag{3}$$

where $\odot$ denotes element-wise multiplication, $\widetilde{\boldsymbol{x}}_{\boldsymbol{i,oh}} \in \mathbb{R}^C$ represents the one-hot encoded version of $\boldsymbol{x}$, changing the representative label to 1 and other labels to 0 for each spatial position $\boldsymbol{i}$. By adding $\widetilde{\boldsymbol{x}}_{\boldsymbol{min}}$ and $\widetilde{\boldsymbol{x}}_{\boldsymbol{norm}}$, from (1) and (3), the preprocessed SSIN output $\overline{\boldsymbol{x}}_{\boldsymbol{i}}$ is defined as follows in (4):

$$\overline{\boldsymbol{x}}_{\boldsymbol{i}} = \widetilde{\boldsymbol{x}}_{\boldsymbol{i,min}} + \widetilde{\boldsymbol{x}}_{\boldsymbol{i,norm}} + \boldsymbol{N_t}(m, \sigma), \tag{4}$$

where $\overline{\boldsymbol{x}}_{\boldsymbol{i}}$ is preprocessed data which the discriminator takes. By concatenating $\widetilde{\boldsymbol{x}}_{\boldsymbol{i}}$ for all spatial positions $\boldsymbol{i}$, $\overline{\boldsymbol{x}} \in \mathbb{R}^{H \times W \times C}$ is obtained. If the representative values of $\widetilde{\boldsymbol{x}}_{\boldsymbol{i}}$ are smaller than τ (4) becomes $\overline{\boldsymbol{x}}_{\boldsymbol{i}} = \widetilde{\boldsymbol{x}}_{\boldsymbol{i}}$. Random noise $\boldsymbol{N_t}(m, \sigma) \in \mathbb{R}^C$ with a truncated normal distribution is added to blur the other unknown factors that may confuse the discriminator, discerning the static/dynamic domain boundary. The truncated normal distribution resamples that value whose values exceed the range of $(m - 2\sigma, m + 2\sigma)$. The standard deviation σ is set as $m/2$ so that the range of $\overline{\boldsymbol{x}}$ does not go negative.

After preprocessing the SSIN output, the static label map $\boldsymbol{y}$ matches the probability distribution of $\overline{\boldsymbol{x}}$. At each pixel $\boldsymbol{i}$, the data position of $\overline{\boldsymbol{x}}_{\boldsymbol{il}}$ and $\overline{\boldsymbol{x}}_{\boldsymbol{ik}}$ is swapped where $\boldsymbol{l}$ is the representative label at $\overline{\boldsymbol{x}}_{\boldsymbol{i}}$ and $\boldsymbol{k}$ is the representative label at $\boldsymbol{y}_{\boldsymbol{i}}$, and the preprocessed static label map $\overline{\boldsymbol{y}}_{\boldsymbol{i}}$ can be expressed as indicated in (5):

$$\begin{aligned}\overline{\boldsymbol{y}}_{\boldsymbol{i}} = \boldsymbol{y_i} \odot \overline{\boldsymbol{x}}_{\boldsymbol{i,max}} + (\max(\boldsymbol{y_i}, \widetilde{\boldsymbol{x}}_{\boldsymbol{i,oh}}) - \boldsymbol{y_i}) \odot \overline{\boldsymbol{x}}_{\boldsymbol{ik}} \\ + (1 - \max(\boldsymbol{y_i}, \boldsymbol{x}_{\boldsymbol{i,oh}})) \odot \boldsymbol{x_i} + \boldsymbol{N_t}(m, \sigma),\end{aligned} \tag{5}$$

where $\overline{\boldsymbol{y}}_{\boldsymbol{i}}$ is preprocessed data of $\boldsymbol{y_i}$, $\odot$ is the elemental-wise multiplication, and $\overline{\boldsymbol{x}}_{\boldsymbol{i,max}}$ is the largest value among $\overline{\boldsymbol{x}}_{\boldsymbol{i}}$. Concatenating $\overline{\boldsymbol{y}}_{\boldsymbol{i}}$ for all spatial positions, $\overline{\boldsymbol{y}} \in \mathbb{R}^{H \times W \times C}$ is obtained and evaluated by the discriminator.

3.3. Objective Function

Adversarial learning helps the network learn the features of different domains, which has shown good performance in unsupervised training. We adopted adversarial learning to train the SSIN to remove dynamic objects without knowing the ground-truth label. In addition, other loss functions are added to stabilize the network during adversarial training, which is a weighted summation of three different objective loss functions. The second loss term is a multi-class cross-entropy term that helps the SSIN preserve the input map's static part $\boldsymbol{x}$. The third loss function is based on the total variation [31], which performs denoising and smoothing. The fourth loss function, the self-entropy term, helps stabilize the second loss function. Training is performed using the backpropagation method [32].

The crucial part of adversarial training in the SSIN is the GAN loss. This involves changing the standard optimization problem to a minimax optimization problem in which the SSIN and discriminator are jointly updated. The loss function for the SSIN and discriminator is defined as follows:

$$L_{GAN} = \min_{G} \max_{D} E[\log(D(\overline{\boldsymbol{y}}_{\boldsymbol{i}})) + \log(1 - D(\overline{\boldsymbol{x}})), \tag{6}$$

where $\overline{\boldsymbol{y}}$ implies the preprocessed static map, and $\overline{\boldsymbol{x}}$ implies the preprocessed SSIN output. $D(\cdot)$ denotes the discriminator function, and G denotes the SSIN. The SSIN aims to minimize the loss function to fool the discriminator, whereas the discriminator maximizes the loss function to differentiate the given data.

The second objective of the SSIN is to remove only the dynamic labels from the input map $\boldsymbol{x}$. In other words, the static labels in $\boldsymbol{x}$ should be unchanged in the SSIN output from the given output. Instead of mean squared error [33], which is used in other inpainting methods for RGB data, the cross-entropy term is used to preserve the static label for the segmentation label set, which reveals the probability of each label. The cross-entropy loss term is defined as follows:

$$L_{CE}(\boldsymbol{x}, \tilde{\boldsymbol{x}}, \boldsymbol{M}) = \sum_{i}^{H \times W} \boldsymbol{M_i} \odot \sum_{c}^{C} (\tau \boldsymbol{x_{ic}} \log \widetilde{\boldsymbol{x}}_{\boldsymbol{ic}} + (1 - \tau \boldsymbol{x_{ic}} \log(1 - \widetilde{\boldsymbol{x}}_{\boldsymbol{ic}}))), \tag{7}$$

where L_{CE} denotes the cross-entropy loss function, $\boldsymbol{x}$ denotes the input of the SSIN, and $\tilde{\boldsymbol{x}}$ denotes the preprocessed output of the SSIN. Moreover, $\widetilde{\boldsymbol{x}}_{\boldsymbol{ic}}$ targets $\boldsymbol{x_{ic}}$, only learning the static map by multiplying mask $\boldsymbol{M}$, which ignores the dynamic label. The τ value is multiplied in the target map to help preserve the preprocessing value.

Training SSIN is unstable due to the unsupervised learning process, which results in noisy features to the output, as illustrated in Figure 4c. From the idea that the recovered pixels must follow the nearby segmentation, we introduce a cross-entropy-based total variation loss function for the binary data to adopt a smoothing effect for the output. The loss function is used during the training stage, which is defined as follows:

$$L_{TV}(\widetilde{\boldsymbol{x}}) = \frac{1}{H \times W} \sum_{h,w}^{H \times W} \boldsymbol{L_{ce}}(\widetilde{\boldsymbol{x}}_{\boldsymbol{h+1,w}}, \widetilde{\boldsymbol{x}}_{\boldsymbol{h,w}}, I) + \boldsymbol{L_{ce}}(\widetilde{\boldsymbol{x}}_{\boldsymbol{h,w+1}}, \widetilde{\boldsymbol{x}}_{\boldsymbol{h,w}}, I), \tag{8}$$

where $\widetilde{\boldsymbol{x}}$ denotes the input segmentation map. From the equation, $\widetilde{\boldsymbol{x}}_{\boldsymbol{h,w}}$ targets nearby pixels $\widetilde{\boldsymbol{x}}_{\boldsymbol{h,w+1}}$ and $\widetilde{\boldsymbol{x}}_{\boldsymbol{h+1,w}}$ with the cross-entropy term. Normally, the L2 norm-based total variation loss function is used for the RGB data [23]. Moreover, different from the Sobel-

based total variation [34], our loss function applies the cross-entropy term for derivable total variation for the segmentation map.

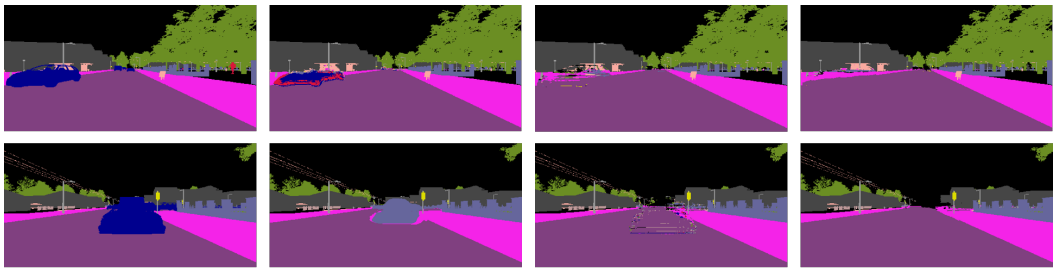
The total variation helps denoise the segmentation map by smoothing the noisy labels with nearby classes; however, total variation averages the nearby values disturbing the data distribution, which drops the confidence of the output map. To maintain the segmentation data close to 0 or 1, a self-entropy-based regularization term is added, which is defined as follows:

$$L_{SE}(\widetilde{\boldsymbol{x}}) = \frac{1}{H \times W} \sum_{i}^{H \times W} -\widetilde{\boldsymbol{x}}_i \log \widetilde{\boldsymbol{x}}_i, \tag{9}$$

where $\boldsymbol{x}$ denotes the 2D input signal. The total variation and self-entropy take $\widetilde{\boldsymbol{x}}_{h,w}$ as input for learning. By combining the loss functions, the loss function for the SSIN trains is based on the following equation:

$$L_{TV} = aL_{GAN} + bL_{CE} + cL_{TV} + dL_{SE}. \tag{10}$$

where a, b, c, and d are weights to balance the different losses.



(a) input image (b) M1 (c) M2 (d) M3

Figure 4. Dynamic removal results under different settings of the model: (**a**) Input Image; (**b**) M1: L_{CE}; (**c**) M2: $L_{CE} + L_{GAN} + L_{SE}$; (**d**) M3: $L_{CE} + L_{GAN} + L_{SE} + L_{TV}$.

3.4. Training Networks

To effectively train the network, the curriculum learning strategy [35] is used by increasing the difficulty level step by step. The training process is performed in three stages. First, the SSIN is trained using the cross-entropy loss to obtain the content of the original static image. Second, when the SSIN is stabilized with the first step, it learns the complicated features from the adversarial loss while the discriminator also continues learning. Finally, the network is fine-tuned with the total variation loss, removing the noise factors. Each stage prepares features for the next one to improve and hence dramatically increases the effectiveness and efficiency of network training. The discriminator is trained after the first stage of the SSIN is finished. As depicted in Figure 4, the reconstruction stage (b) restores the existing static labels from the input, although the prediction for the dynamic labels is unnatural. Then, the adversarial stage (c) helps generate more structural details to make a realistic static map. The total variation (d) finally stabilizes the training and removes the noise. The discriminator learns just before the (c) stage to avoid the case in which the discriminator is too strong and stops learning.

3.5. Network Architecture

SSIN structure is based on deep convolutional GAN (DCGAN) [20] with one convolutional generator and a convolutional discriminator. We adopt the U-Net-based [5] convolutional encoder–decoder [14] model for better generator performance. The discriminator follows the basic structure of DCGAN, consisting of 6 convolution layers and 1 fully connected layer. The details of the generator structure are shown in Table 1. At the encoder stage of the SSIN, it extracts features from the segmentation map with convolutional filters, reducing the resolution. Strided convolution with a rate of 2 is used to reduce it by one-fourth of the original size to improve the performance. After the encoder stage,

the decoder restores the missing area with the extracted features of the input map. Skip connections are added to help the decoder directly obtain features from the encoder for better restoration [14]. Unlike other architectures that use deconvolution to increase the resolution, our model increases the resolution twice using nearest neighbor interpolation to avoid generating checkboard artifacts of the output map [36]. The discriminator network architecture is based on convolutional layers that extract features by steadily reducing the resolution. In the last layer, the fully connected layer is connected to provide domain information on the input. The discriminator structure follows [20] for a discriminator network as listed in Table 2.

Leaky rectified linear unit [37], defined as $\sigma(x) = \max(a \times x, x)$), is used for the mid-layer activation function of the networks and is placed after the convolutional layers for nonlinear activation functions, which prevents the dying problem of ReLU. In addition, batch normalization [38] is performed between the activation function and the convolutional layer to stabilize the propagation of gradients through the network. The normalization forms the output of each layer with the Gaussian distribution of the zero mean and unit variance, which resolves the gradient vanishing problem and boosts the model to learn. This structure applies to all generator and discriminator layers except the last ones. The softmax function is applied in the last layer of the generator to make the segmentation map, and the sigmoid is placed on the discriminator's last layer to represent the probability that the input is real rather than restored.

Table 1. Semantic segmentation inpainting network architecture details.

Semantic Segmentation Inpainting Network				
Layer	**Kernel Size**	**Stride**	**Output Size**	**Skip Connections**
Input			C × 256 × 256	
Encoder-conv 1-1	5 × 5	1 × 1	64 × 256 × 256	
Encoder-conv 1-2	3 × 3	1 × 1	64 × 256 × 256	
Encoder-conv 1-3	3 × 3	1 × 1	64 × 256 × 256	Decoder-conv 1-1
Encoder-conv 2-1	3 × 3	2 × 2	128 × 128 × 128	
Encoder-conv 2-2	3 × 3	1 × 1	128 × 128 × 128	
Encoder-conv 2-3	3 × 3	1 × 1	128 × 128 × 128	Decoder-conv 2-1
Encoder-conv 3-1	3 × 3	2 × 2	256 × 64 × 64	
Encoder-conv 3-2	3 × 3	1 × 1	256 × 64 × 64	
Encoder-conv 3-3	3 × 3	1 × 1	256 × 64 × 64	Decoder-conv 3-1
Encoder-conv 4-1	3 × 3	2 × 2	512 × 32 × 32	
Encoder-conv 4-2	3 × 3	1 × 1	512 × 32 × 32	
Encoder-conv 4-3	3 × 3	1 × 1	512 × 32 × 32	
Upsampling			512 × 64 × 64	
Decoder-conv 1-1	3 × 3	1 × 1	256 × 64 × 64	Encoder-conv 1-3
Decoder-conv 1-2	3 × 3	1 × 1	256 × 64 × 64	
Decoder-conv 1-3	3 × 3	1 × 1	256 × 64 × 64	
Upsampling			512 × 64 × 64	
Decoder-conv 2-1	3 × 3	1 × 1	128 × 128 × 128	Encoder-conv 2-3
Decoder-conv 2-2	3 × 3	1 × 1	128 × 128 × 128	
Decoder-conv 2-3	3 × 3	1 × 1	128 × 128 × 128	
Upsampling			512 × 64 × 64	
Decoder-conv 3-1	3 × 3	1 × 1	64 × 256 × 256	Encoder-conv 3-3
Decoder-conv 3-2	3 × 3	1 × 1	64 × 256 × 256	
Output	3 × 3	1 × 1	C × 256 × 256	

Table 2. Discriminator architecture details.

Discriminator Network			
Layer	**Kernel Size**	**Stride**	**Output Size**
Input			C × 256 × 256
Encoder-conv 1	5	2 × 2	64 × 128 × 128
Encoder-conv 2	5	2 × 2	128 × 64 × 64
Encoder-conv 3	5	2 × 2	256 × 32 × 32
Encoder-conv 4	5	2 × 2	512 × 16 × 16
Encoder-conv 5	5	2 × 2	512 × 8 × 8
Encoder-conv 6	5	2 × 2	512 × 4 × 4
Fully Connected			1024
Output			1

4. Experiment and Results

In this section, we evaluate the performance of the unsupervised SSIN through experiments. Synthetic and real-world road datasets are used for the experiment. The synthetic dataset MICC-SRI [9] includes paired ground-truth static labels, but the real-world dataset Cityscapes [10] does not have ground truth for the static labels. A comparative experiment was conducted with the proposed SSIN and non-preprocessing methods, the DCGAN [20], an adversarial entropy network (AdvEnt) [27], and the preprocessed method, the segmentation adversarial network (SegAdv) [29], using MICC-SRI. The results for the Cityscapes images are also compared qualitatively and quantitatively.

4.1. Datasets

Mainly two datasets were used to conduct the experiment. First, the MICC-SRI [9] dataset is made from the Car Learning to Act [39] simulation, an open-source driving simulator. It contains 11,913 segmentation maps of 13 classes of scene elements. It includes both a static map and a dynamic map for the same scene, which helps quantitatively evaluate the inpainting segmentation result. We split the MICC-SRI dataset into 10,000 maps for training and 1913 maps for testing. For unsupervised learning, the training set was split again into 5000 maps for the dynamic map domain and 5000 maps for the static map domain. The segmentation maps were resized from 800 × 600 to 512 × 256. During the training stage, the maps were randomly cropped to a size of 256 × 256.

To match both classes of the MICC-SRI and Cityscapes datasets, we removed some of the segmentation classes and defined a new set as follows:

$$D = \{Person, Veichle\}$$

$$D = \left\{ \begin{array}{c} Road, Sidewalk, Building, Wall, Fence, Pole, \\ TrafficMaker, Vegetation, Other, Unlabeld \end{array} \right\},$$

where D denotes the dynamic class and S denotes the static class. Other classes not included in the set were merged into the defined class.

All the procedures were implemented in Tensorflow version 1.14. An Adam optimization method [40] was used for training the SSIN with a learning rate of 10^{-6} and the momentum of 0.9 to minimize (10) for the SSIN. The discriminator follows the same learning rule as the SSIN. The parameters of the loss function (10) for the training-loss weighting factor of the SSIN were set as follows: $a = 1$ for the static-preserving cross-entropy term, $b = 3$ for the adversarial training, $c = 40$ for the total variation, and $d = 60$ for the self-entropy term.

4.2. Result on MICC-SRI Dataset

We compared other methods based on unsupervised learning where the ground-truth labels were not used during training. We also added supervised training results using ground-truth labels during the training stage.

We reported our evaluation results in terms of the per-pixel score, F1, and mean intersection of union (mIOU). The per-pixel accuracy reveals the overall accuracy of the segmentation. The F1 and mIOU are used to calculate the average of the intersecting area of every class, showing the details of each class reconstruction performance. Moreover, the inference speed is as reported in terms of the frame per second, for which a higher value indicates a faster inference speed. As indicated in Table 3, the self-entropy term in AdvEnt [27] and our methods performed better than the vanilla GAN loss-term-based methods. This is because the self-entropy term directly stabilized the adversarial learning for the probability map. In addition, our preprocessing increases the performance, emphasizing the need to use a preprocessing method. The total variation proved that smoothing the label during training increases the performance.

The details of the mIOU results per class are depicted in Table 4. The result reveals that the large area classes (i.e., sky, roads, and sidewalks) provide comparable results. This exhibits that adversarial training recovers the overall structural information but fails to give details due to the unstable training. On the other hand, the self-entropy methods (AdvEnt [27] and ours) showed best at restoring the detail classes (i.e., poles and fences).

Table 3. Results on MICC-SRI dataset.

	Method	Pixel Accuracy	F1	mIOU	FPS
Unsupervised	DCGAN [20]	66.44	38.09	30.64	18.2
	SegAdv [29]	67.79	38.33	32.11	17.4
	AdvEnt [27]	73.31	52.11	42.87	17.9
	DCGAN + Our Preprocessing (M1)	67.95	39.10	30.72	16.7
	AdvEnt + Our Preprocessing (M2)	73.57	51.25	41.61	16.5
	Ours	**75.59**	**57.86**	**44.42**	16.1
Supervised	DCGAN [20]	86.70	62.31	56.32	18.2
	SegAdv [29]	**87.18**	62.25	57.17	17.4
	AdvEnt [27]	86.39	63.02	57.53	17.9
	Ours	86.56	**63.12**	**57.86**	16.1

Table 4. Semantic segmentation restoration performance mIOU of unsupervised learning on MICC-SRI (%).

	Sky	Building	Fence	Pole	Road	Sidewalk	Vegetation	Wall	Sign	mIOU
DCGAN [20]	21.2	37.5	11.1	26.0	77.8	37.4	19.4	23.4	21.6	30.6
SegAdv [29]	2.1	34.8	3.5	29.4	76.2	29.0	26.4	25.5	52.5	31.1
AdvEnt [27]	48.3	44.7	9.6	30.4	78.8	**49.0**	31.5	**37.4**	**56.3**	42.9
Ours	**51.6**	**46.2**	**27.1**	**43.9**	**80.9**	40.3	**33.5**	32.4	42.5	**44.2**

The qualitative results are illustrated in Figures 5 and 6. The first and third pictures demonstrate that the self-entropy methods try to recover more detailed information than the DCGAN, in which the sidewalk was recovered with the structural information. This can also be observed in the second example on the fence on the left side. The fence patterns were recovered plausibly. Moreover, our method offers more stable results that include less noise than the AdvEnt. The examples are observed in the fourth and fifth images of Figure 6, where the dynamic labels are filled with nearby static labels. In addition, the fifth example exhibits the smoothing effect on the sky and buildings, recovering the car, which demonstrates the effect of the entropy-based total variation loss function. More details of the result are illustrated in Figure 7, where the grayscale of the mask error map indicates the prediction error from the ground-truth labels, and the white space reveals the correct answer. The mask result represents the restored map in the dynamic map. For the first and

second pictures in Figure 7, the structural information for the buildings and road can be restored naturally.

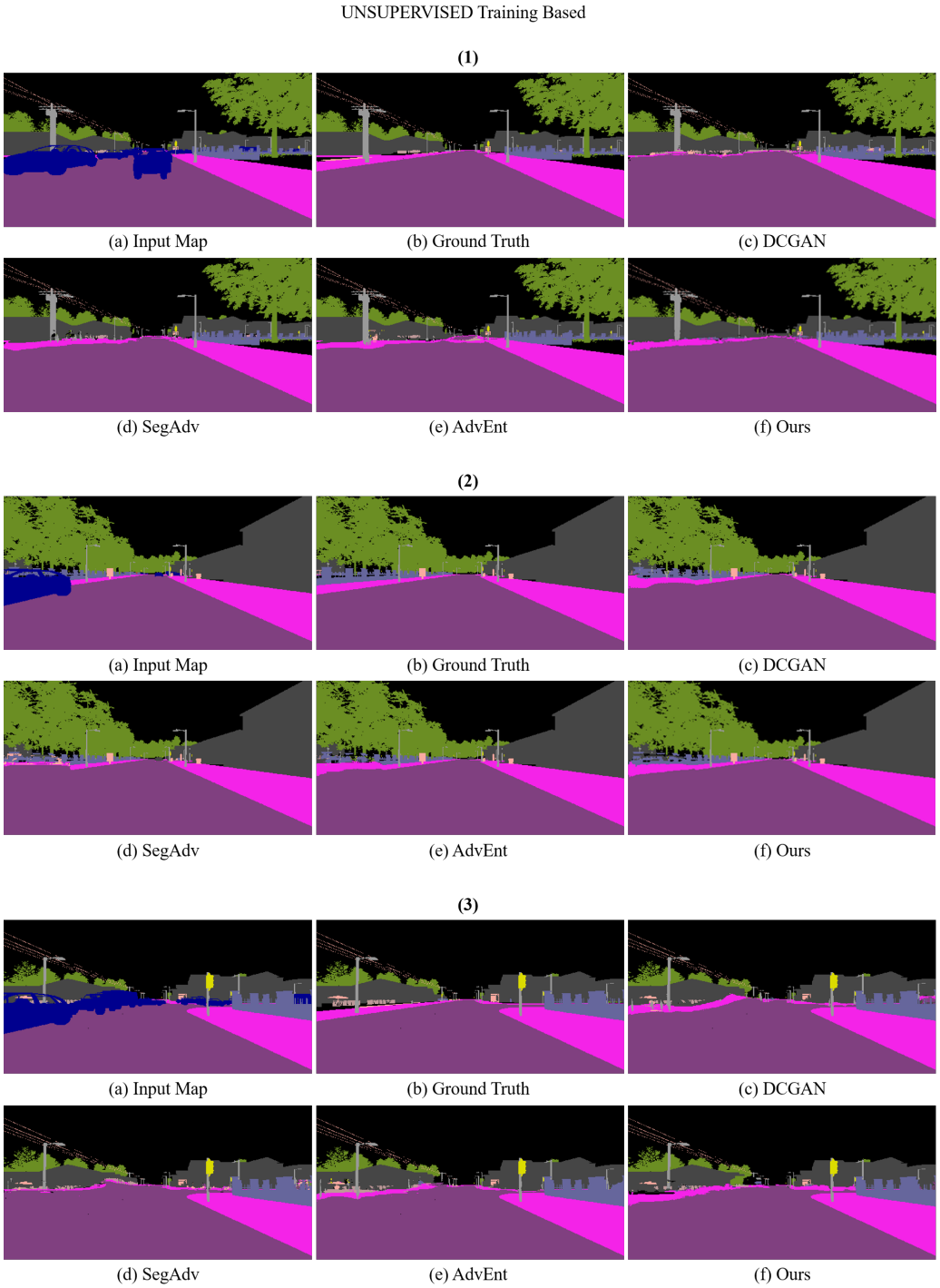


Figure 5. Unsupervised learning-based Media Integration and Communication Center Semantic Road Inpainting (MICC-SRI) qualitative results: (**a**) input map, (**b**) ground truth, (**c**) deep convolutional generative adversarial network (DCGAN) [20], (**d**) segmentation adversarial network (SegAdv) [29], (**e**) adversarial entropy network (AdvEnt) [27], and (**f**) semantic segmentation inpainting network (SSIN).

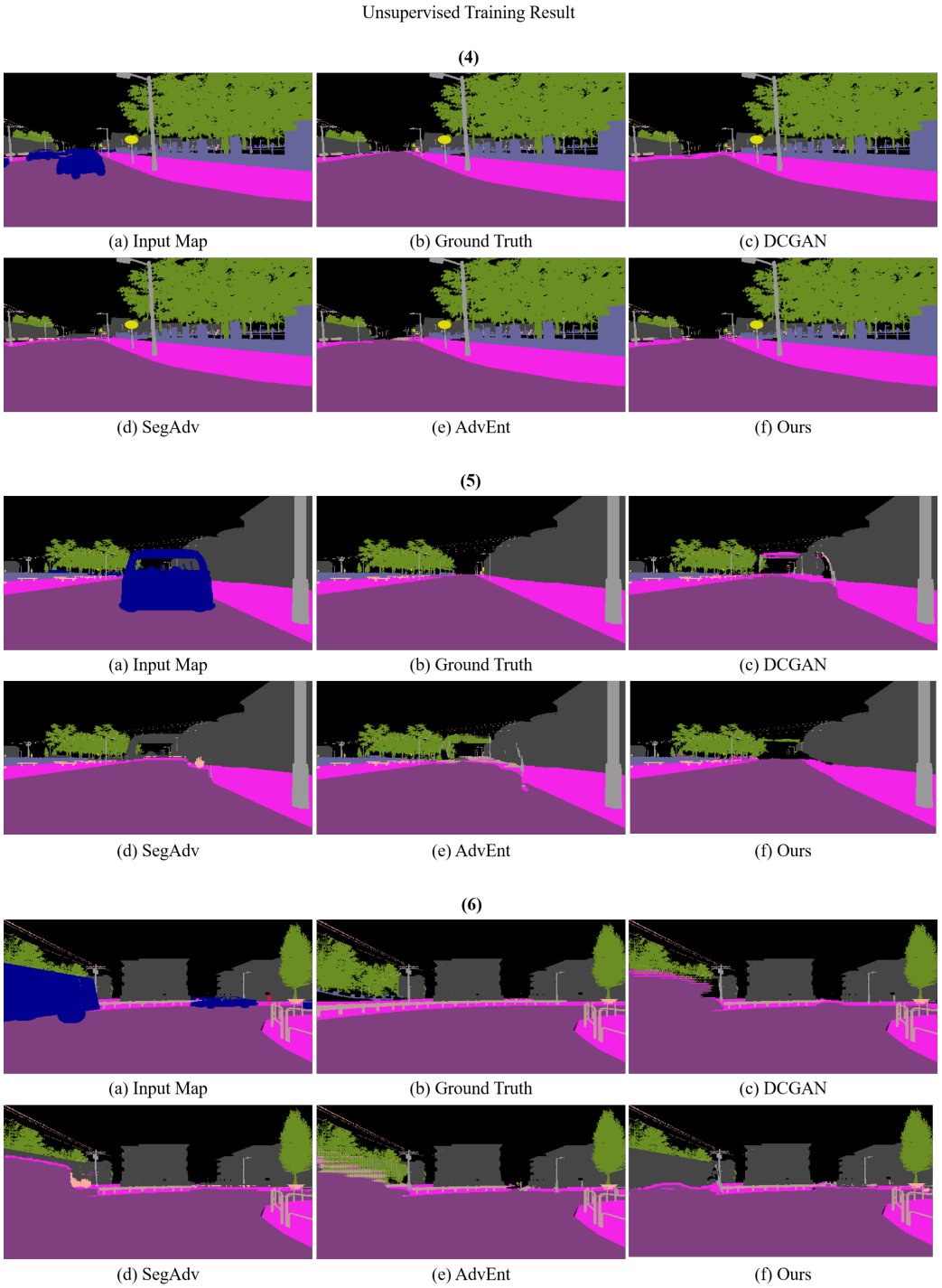


Figure 6. Additional results for the unsupervised learning-based Media Integration and Communication Center Semantic Road Inpainting (MICC-SRI) qualitative results: (**a**) input map, (**b**) ground truth, (**c**) deep convolutional generative adversarial network (DCGAN) [20], (**d**) segmentation adversarial network (SegAdv) [29], (**e**) adversarial entropy network (AdvEnt) [27], and (**f**) semantic segmentation inpainting network (SSIN).

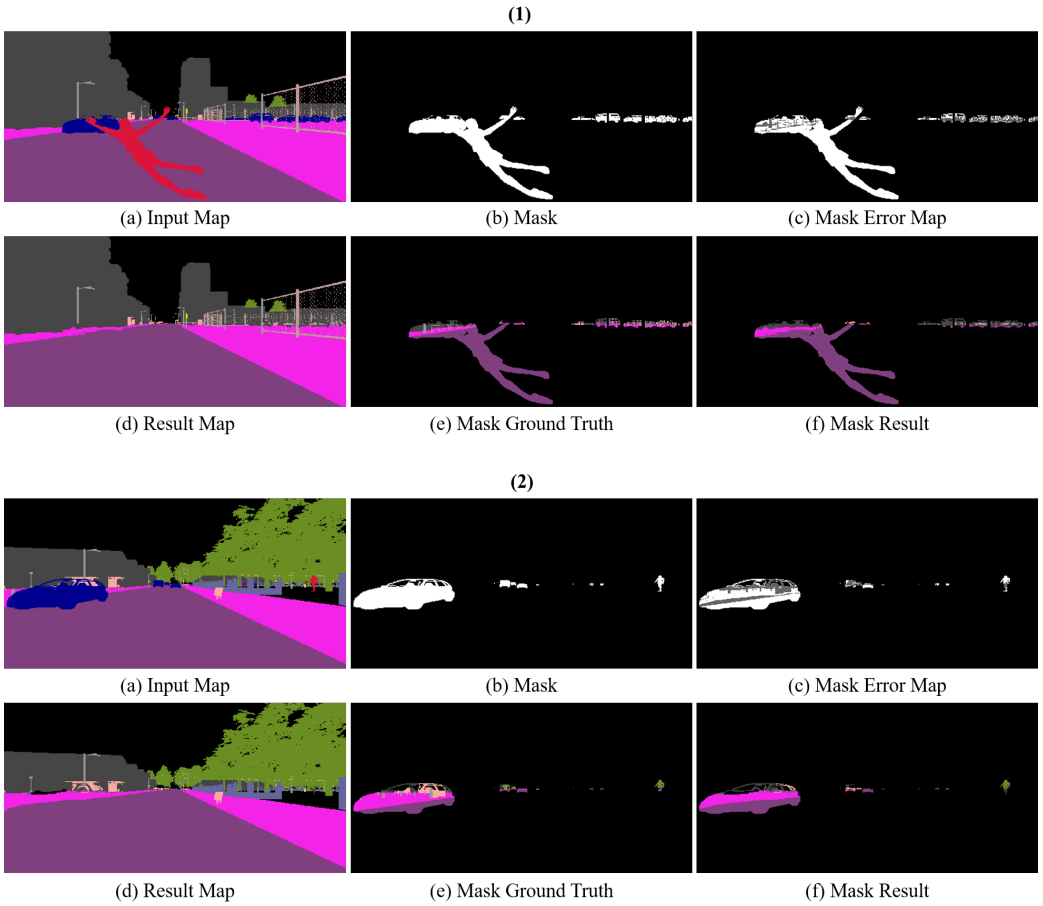


Figure 7. Detailed results of the purposed semantic segmentation inpainting network (SSIN) on Media Integration and Communication Center Semantic Road Inpainting (MICC-SRI) data.

We also trained the network using the supervised method by adding the ground-truth labels from the MICC-SRI dataset to remove the dynamic objects. From the reconstruction loss function (8), the cross-entropy with the opposite mask (0 for the static objects and 1 for the dynamic objects) is added with a weighting factor of 3, and the adversarial loss weighting factor reduces to 0.001 to focus more on supervised learning.

The quantitative evaluation in supervised learning is detailed in Table 3. The results indicate that supervised training does not demonstrate a substantial difference in the results. Because the preprocessing method of the SegAdv was set up on the assumption of supervised learning, the SegAdv [29] network learns to fill in the segmentation map better than the other methods, as in Table 3. However, our method provides slightly better results for the F1 and mIOU scores, meaning that our method recovers detailed information. Regarding supervised learning, the ground truth plays a key role during training. The images illustrating the supervised learning results are shown in Figure 8, which reveals similar results for the small area because of the stabilized supervised learning process. However, even with supervised learning, it is difficult to fill large regions in the corner of the map, as observed in the first picture. The first image implies that our method learns the best structural information, such as for the sidewalk.

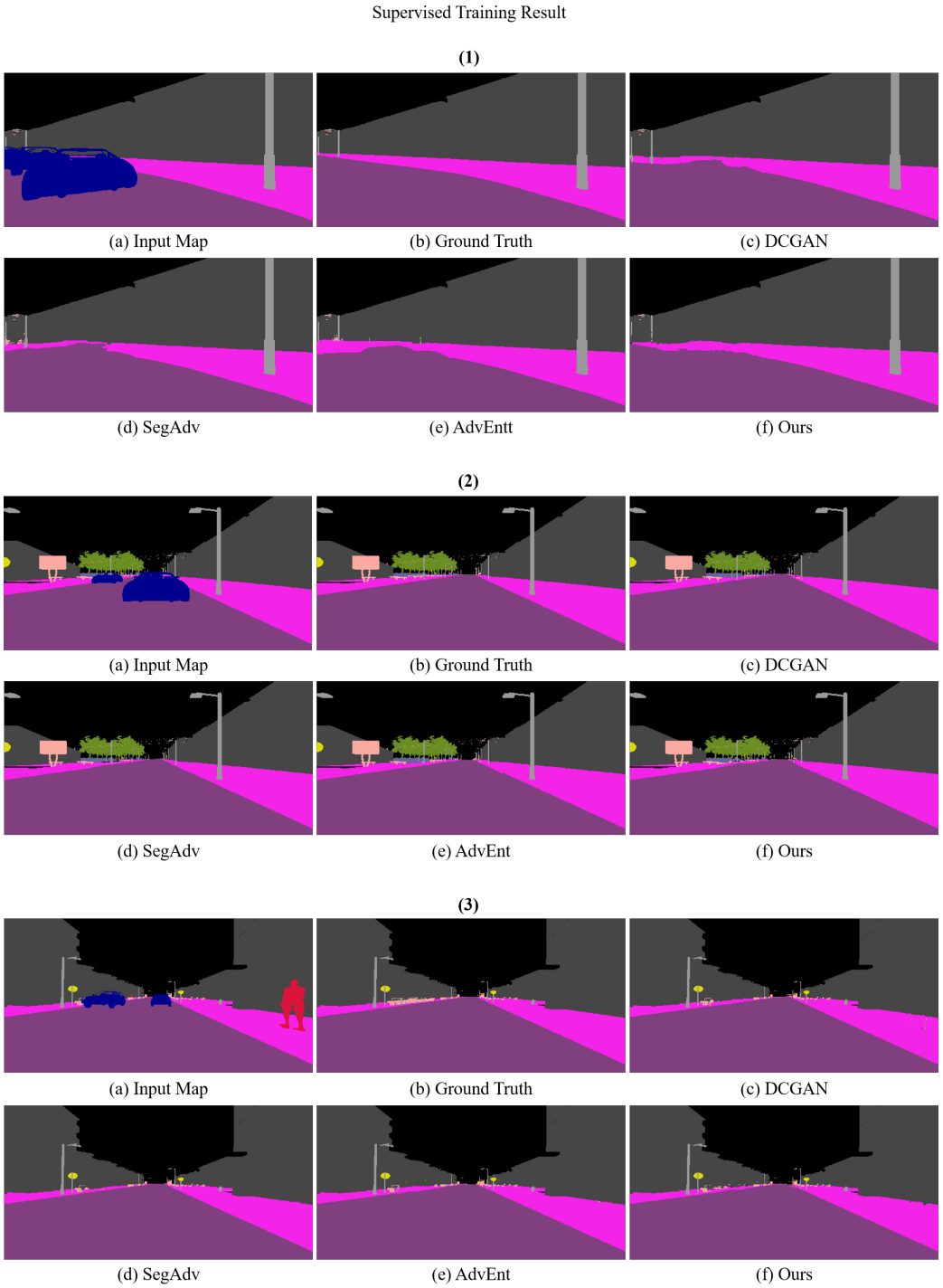
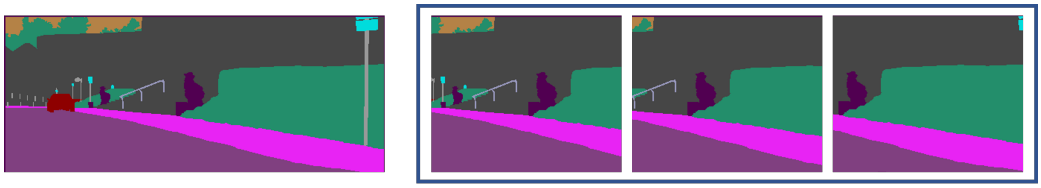


Figure 8. Supervised learning-based Media Integration and Communication Center Semantic Road Inpainting (MICC-SRI) qualitative results: (**a**) input map, (**b**) ground truth, (**c**) deep convolutional generative adversarial network (DCGAN) [20], (**d**) segmentation adversarial network (SegAdv) [29], (**e**) adversarial entropy network (AdvEnt) [27], and (**f**) semantic segmentation inpainting network (SSIN).

4.3. Result on Cityscapes

We also experimented with the Cityscapes dataset [22], a real-world dataset. The dataset contains 3475 annotated images of 20 classes and is split into 2975 data for training and 500 for testing. The segmentation maps are resized for the network from 2048 × 1024 to 512×256. To obtain the static segmentation map from Cityscapes, we cropped a 256×256 static map from the resized map with a difference of 10 pixels. A total of 977 segmentation

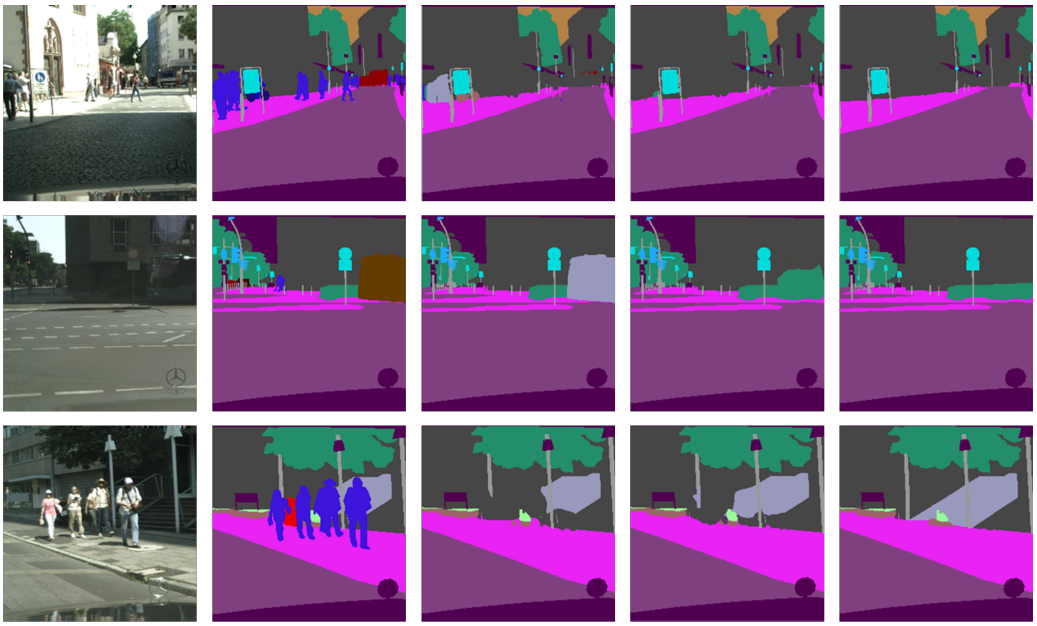
maps were extracted, as presented in Figure 9. In contrast, the 977 dynamic maps were randomly selected, except for the map used for extracting the static map. The annotation class is redefined as noted in Section 4.



(a) Original Segmentation Map (b) Cropped Static Segmentation Map

Figure 9. Shows how the static segmentation map is obtained from the Cityscape dataset. (**a**) Original Cityscape segmentation map, (**b**) static map cropped with an interval of 10 pixels from image (**a**).

We trained our method on the real-world Cityscapes dataset using only the segmentation map. The experiment was conducted using the DCGAN and our method. As shown in Figure 10, our method inpaints the dynamic objects with static objects using the map's nearby segmentation labels and structural information. In the first example in Figure 10, the DCGAN replaces the person on the left side with a chair label, which cannot be found on the map. In contrast, our method used the map information, filling in with the building and sidewalk labels. This is also shown in the second map, where the DCGAN fills the bus with stairs, whereas ours fills it with nearby vegetation. In addition, we made 30 ground truths to compare qualitatively and report the result in Table 5, which shows that our method outperforms the DCGAN in all metrics.



(a) Image (b) Input (c) DCGAN (d) Ours (e) Ground Truth

Figure 10. Unsupervised learning-based Cityscapes qualitative results. (**a**) Original image, (**b**) input map, (**c**) DCGAN, (**d**) ours, (**e**) ground truth.

Table 5. Results on Cityscapes.

Method	Pixel Accuracy	F1	mIOU
DCGAN [20]	54.63	35.89	30.47
Ours	**65.78**	**45.77**	**40.14**

4.4. Hyperparameter Study

For the introduced preprocessing method, two new hyperparameters in (1) need to be set before the training, τ and m. Both hyperparameters were selected quantitatively by comparing the results using the MICC-SRI as shown in Table 6. The best result values are used when choosing a τ value of 0.97 and a mean value of 0.01. For the best dynamic removal performance, we selected 0.97 for τ and 0.01 for the noise mean value m.

Table 6. Result on various hyperparameters.

Maximum Value τ	N(m,σ)	Pixel Accuracy
1	0.01	67.32
0.99	0.01	72.28
0.97	0.01	**75.59**
0.95	0.01	71.27
0.90	0.01	68.78
0.97	0	73.62
0.97	0.010	**75.59**
0.97	0.025	74.35
0.97	0.035	73.14
0.97	0.050	73.47

Compared to the results without the preprocessing applied ($\tau = 1, m = 0$), it implies that preprocessing helps the network's learning. In other words, limiting the maximum value of the labels helps the discriminator learn the desired features with no discriminative factor. Moreover, the noise factor helps the learning of the network, but it does not mean that the noise factor is indispensable.

5. Conclusions

Predefined free-space semantic segmentation information where dynamic objects are not included is crucial for autonomous driving systems. In real-world roads, it is challenging to obtain a pair dynamic and static segmentation map in which supervised learning is unavailable for removing the dynamic objects. For this reason, unsupervised adversarial training plays an important part. Many studies have been conducted on using adversarial training for segmentation maps, but there are limitations to probability maps to apply existing adversarial training directly. In this work, we proposed an unsupervised method for semantic segmentation inpainting with a novel preprocessing method and cross-entropy-based total variation loss function to stabilize the unsupervised adversarial learning for segmentation maps. Our network is based on the GAN, including a U-Net-based encoder–decoder generator trained with one adversarial loss function, two stabilizing loss terms, and semantic regularization as the discriminator. The proposed model successfully synthesized semantically and visually plausible content to remove dynamic objects and naturally replace the objects with static objects. We compared the qualitative and quantitative experiments on both the MICC-SRI and Cityscapes datasets, with three adversarial training methods for the segmentation map. We demonstrated that our model completion results exhibit a high perceptual quality and are flexible for various situations in unsupervised semantic segmentation inpainting.

Author Contributions: Conceptualization, W.-J.A. and D.-W.K.; methodology, D.-S.P. and M.-T.L.; software, W.-J.A.; validation, T.-K.K. and W.-J.A.; formal analysis, D.-W.K.; investigation, D.-W.K. and W.-J.A.; resources, D.-S.P.; data curation, T.-K.K.; writing—original draft preparation, W.-J.A.; writing—review and editing, D.-W.K. and W.-J.A.; visualization, D.-S.P.; supervision, M.-T.L.; project administration, M.-T.L.; funding acquisition, M.-T.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Basic Science Research Program through National Research Foundation of Korea (NRF) (Grant No. 2022R1F1A1073543).

Data Availability Statement: The MICC-SRI and Cityscapes datasets presented in this study are openly available in the MIIC-SRI paper [9] and Cityscapes paper [10], respectively.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ivanovs, M.; Ozols, K.; Dobrajs, A.; Kadikis, R. Improving semantic segmentation of urban scenes for self-driving cars with synthetic images. *Sensors* **2022**, *22*, 2252. [CrossRef] [PubMed]
2. Yao, J.; Ramalingam, S.; Taguchi, Y.; Miki, Y.; Urtasun, R. Estimating drivable collision-free space from monocular video. In Proceedings of the 2015 IEEE Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 5–9 January 2015; pp. 420–427.
3. Jiménez, F.; Clavijo, M.; Cerrato, A. Perception, positioning and decision-making algorithms adaptation for an autonomous valet parking system based on infrastructure reference points using one single LiDAR. *Sensors* **2022**, *22*, 979. [CrossRef] [PubMed]
4. Lai, X.; Tian, Z.; Xu, X.; Chen, Y.; Liu, S.; Zhao, H.; Wang, L.; Jia, J. DecoupleNet: Decoupled network for domain adaptive semantic segmentation. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 369–387.
5. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
6. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 834–848. [CrossRef] [PubMed]
7. Mehta, S.; Rastegari, M.; Caspi, A.; Shapiro, L.; Hajishirzi, H. Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 552–568.
8. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. *Generative Adversarial Networks*; ACM: New York, NY, USA, 2020; Volume 63, pp. 139–144.
9. Berlincioni, L.; Becattini, F.; Galteri, L.; Seidenari, L.; Bimbo, A.D. Road layout understanding by generative adversarial inpainting. In *Inpainting and Denoising Challenges*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 111–128.
10. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The cityscapes dataset for semantic urban scene understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3213–3223.
11. Ballester, C.; Bertalmio, M.; Caselles, V.; Sapiro, G.; Verdera, J. Filling-in by joint interpolation of vector fields and gray levels. *IEEE Trans. Image Process.* **2001**, *10*, 1200–1211. [CrossRef] [PubMed]
12. Barnes, C.; Shechtman, E.; Finkelstein, A.; Goldman, D.B. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.* **2009**, *28*, 24. [CrossRef]
13. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
14. Mao, X.; Shen, C.; Yang, Y.B. Image Restoration Using very Deep Convolutional Encoder-Decoder Networks with Symmetric Skip Connections. 2016 ; Volume 29. Available online: https://arxiv.org/abs/1603.09056v2 (accessed on 30 December 2022).
15. Pathak, D.; Krahenbuhl, P.; Donahue, J.; Darrell, T.; Efros, A.A. Context encoders: Feature learning by inpainting. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2536–2544.
16. Iizuka, S.; Simo-Serra, E.; Ishikawa, H. Globally and locally consistent image completion. *ACM Trans. Graph. (ToG)* **2017**, *36*, 1–14. [CrossRef]
17. Yu, J.; Lin, Z.; Yang, J.; Shen, X.; Lu, X.; Huang, T.S. Generative image inpainting with contextual attention. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5505–5514.
18. Saharia, C.; Chan, W.; Chang, H.; Lee, C.; Ho, J.; Salimans, T.; Fleet, D.; Norouzi, M. Palette: Image-to-image diffusion models. In Proceedings of the ACM SIGGRAPH 2022 Conference, Vancouver, BC, Canada, 7–11 August 2022 ; pp. 1–10.
19. Song, Y.; Yang, C.; Shen, Y.; Wang, P.; Huang, Q.; Kuo, C.C.J. Spg-net: Segmentation prediction and guidance network for image inpainting. In Proceedings of the British Machine Vision Conference, Newcastle, UK, 3–6 September 2018; p. 97. Available online: https://github.com/unizard/AwesomeArxiv/issues/175 (accessed on 30 December 2022).
20. Radford, A.; Metz, L.; Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In Proceedings of the International Conference on Learning Representations, San Juan, PR, USA, 2–4 May 2016.
21. Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; Chen, X. Improved techniques for training gans. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; Volume 29.
22. Wang, T.C.; Liu, M.Y.; Zhu, J.Y.; Tao, A.; Kautz, J.; Catanzaro, B. High-resolution image synthesis and semantic manipulation with conditional gans. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8798–8807.
23. Liu, G.; Reda, F.A.; Shih, K.J.; Wang, T.C.; Tao, A.; Catanzaro, B. Image inpainting for irregular holes using partial convolutions. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 85–100.

24. Li, Y.; Liu, S.; Yang, J.; Yang, M.H. Generative face completion. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3911–3919.
25. Taigman, Y.; Polyak, A.; Wolf, L. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
26. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2223–2232.
27. Vu, T.H.; Jain, H.; Bucher, M.; Cord, M.; Pérez, P. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2517–2526.
28. Hoffman, J.; Tzeng, E.; Park, T.; Zhu, J.Y.; Isola, P.; Saenko, K.; Efros, A.; Darrell, T. Cycada: Cycle-consistent adversarial domain adaptation. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 1989–1998. Available online: https://www.aminer.org/pub/5a260c8617c44a4ba8a326ee/cycada-cycle-consistent-adversarial-domain-adaptation (accessed on 30 December 2022).
29. Luc, P.; Couprie, C.; Chintala, S.; Verbeek, J. Semantic segmentation using adversarial networks. *arXiv* **2016**, arXiv:1611.08408.
30. Hung, W.C.; Tsai, Y.H.; Liou, Y.T.; Lin, Y.Y.; Yang, M.H. Adversarial Learning for Semi-Supervised Semantic Segmentation. In Proceedings of the British Machine Vision Conference, Newcastle, UK, 3–6 September 2018 .
31. Rudin, L.I.; Osher, S.; Fatemi, E. Nonlinear total variation based noise removal algorithms. *Phys. D Nonlinear Phenom.* **1992**, *60*, 259–268. [CrossRef]
32. Hecht-Nielsen, R. Theory of the backpropagation neural network In *Neural Networks for Perception* ; Academic Press: Cambridge, MA, USA, 1992; pp. 65–93.
33. Janocha, K.; Czarnecki, W.M. On loss functions for deep neural networks in classification. *arXiv* **2017**, arXiv:1702.05659.
34. Javanmardi, M.; Sajjadi, M.; Liu, T.; Tasdizen, T. Unsupervised total variation loss for semi-supervised deep learning of semantic segmentation. *arXiv* **2016**, arXiv:1605.01368.
35. Bengio, Y.; Louradour, J.; Collobert, R.; Weston, J. Curriculum learning. In Proceedings of the 26th Annual International Conference on Machine Learning, Montreal, QC, Canada, 14–18 June 2009; pp. 41–48.
36. Wojna, Z.; Ferrari, V.; Guadarrama, S.; Silberman, N.; Chen, L.C.; Fathi, A.; Uijlings, J. The devil is in the decoder: Classification, regression and gans. *Int. J. Comput. Vis.* **2019**, *127*, 1694–1706. [CrossRef]
37. Maas, A.L.; Hannun, A.Y.; Ng, A.Y. Rectifier nonlinearities improve neural network acoustic models. In Proceedings of the International Conference on Machine Learning, Atlanta, GA, USA, 16–21 June 2013; Volume 30, p. 3.
38. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 448–456.
39. Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; Koltun, V. CARLA: An open urban driving simulator. In Proceedings of the Conference on Robot Learning, Mountain View, CA, USA, 13–15 November 2017; pp. 1–16.
40. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.