WildNet: Learning Domain Generalized Semantic Segmentation from the Wild

Suhyeon Lee  Hongje Seong  Seongwon Lee  Euntai Kim*
School of Electrical and Electronic Engineering, Yonsei University, Seoul, Korea
{hyeon93, hjseong, won4113, etkim}@yonsei.ac.kr

Abstract

We present a new domain generalized semantic segmentation network named WildNet, which learns domain-generalized features by leveraging a variety of contents and styles from the wild. In domain generalization, the low generalization ability for unseen target domains is clearly due to overfitting to the source domain. To address this problem, previous works have focused on generalizing the domain by removing or diversifying the styles of the source domain. These alleviated overfitting to the source-style but overlooked overfitting to the source-content. In this paper, we propose to diversify both the content and style of the source domain with the help of the wild. Our main idea is for networks to naturally learn domain-generalized semantic information from the wild. To this end, we diversify styles by augmenting source features to resemble wild styles and enable networks to adapt to a variety of styles. Furthermore, we encourage networks to learn class-discriminant features by providing semantic variations borrowed from the wild to source contents in the feature space. Finally, we regularize networks to capture consistent semantic information even when both the content and style of the source domain are extended to the wild. Extensive experiments on five different datasets validate the effectiveness of our WildNet, and we significantly outperform state-of-the-art methods. The source code and model are available online: https://github.com/suhyeonlee/WildNet.

1. Introduction

Domain generalized semantic segmentation aims to better predict pixel-level semantic labels on multiple unseen target domains while learning only on the source domain. Unfortunately, the domain shift between the source and target domains makes a segmentation model trained on the given source data behave stupidly on the unseen target data, as shown in Fig. 1b. In domain generalization (DG), the low generalization performance for unseen domains is obviously due to overfitting to the source domain. Since the model cannot see any information about the target domains in the learning process and even unlabeled target images are not provided unlike domain adaptation (DA), it over-learns the statistical distribution of the given source data.

Recently, some studies [7, 29, 41, 42] have proposed learning the domain-generalized content feature by ‘removing’ domain-specific style information from the data to prevent overfitting to the source domain. Based on the correlation between the feature’s covariance matrix and style [13, 14], they assumed that only content features would remain if elements of features considered the domain-specific style were whitened [23, 30, 50, 53]. However, since the content and style are not orthogonal, whitening the style may cause a loss of semantic content, which is indispensable for semantic category prediction. As a result, they predict semantic categories from incomplete content features and have difficulty making accurate predictions, as shown in Fig. 1c.

In this paper, we propose a new domain generalized semantic segmentation network called WildNet, which learns...
the domain-generalized semantic feature by ‘extending’ both content and style to the wild. Although some previous works [22, 45, 62] utilized various styles from the wild, e.g., ImageNet [11] for real styles and Painter by Numbers [38] for unreal styles, they overlooked that the high generalization ability comes from learning not only various styles but also various contents. In contrast to previous studies, our main idea is to naturally learn domain-generalized semantic information by leveraging a variety of contents and styles from the wild, without forcing whitening on domain-specific styles.

To extend both content and style to the wild, we present four effective learning methods. (i) Based on the relevance of style and feature statistics, feature stylization diversifies the style of the source feature by transferring the statistics of the wild feature to the source feature over several layers. (ii) To prevent overfitting to the source contents, we propose content extension learning to increase the intra-class content variability in the latent embedding space. Extending content from source to wild helps networks make generalized predictions on unseen contents. (iii) To prevent overfitting to the source style, we propose style extension learning to encourage networks to adapt to the various styles extended to the wild. (iv) Finally, semantic consistency regularization enables networks to capture consistent semantic information even when both the content and style of the source domain are extended to the wild. With the proposed learning methods, our WildNet learns domain-generalized semantic features by leveraging a variety of contents and styles from the wild. Extensive experiments over multiple domains show that our network achieves superior performance on domain generalization for semantic segmentation.

Our main contributions are as follows:

- We present a novel domain generalized semantic segmentation network named WildNet, which learns domain-generalized semantic features by leveraging a variety of contents and styles from the wild.
- We propose four learning techniques to train domain-generalized networks by extending both the content and style of the source domain to the wild. These enable our model to make reliable predictions on various unseen target domains without training on them.
- Our network achieves superior performance in extensive experiments on domain generalization for semantic segmentation constructed over multiple domains.

2. Related Work

2.1. Domain Adaptation and Generalization

Domain adaptation (DA) aims to increase the performance on the target domain by reducing the domain gap between the source and target domains. In semantic segmentation, DA is exploited to tackle the effort of annotating pixel-level categories in an image. Most DA methods train networks using the ‘given’ target images via image translation [17, 20, 33, 60, 64], feature alignment [21, 44, 55–57], and self-training [16, 29, 32, 40, 63] strategies. However, it is hard to acquire target images from various environments during the learning process, and efforts to retrain networks are required whenever applying networks to a new target domain.

To overcome these limitations, domain generalization (DG) has recently attracted considerable attention. However, most DG studies have focused on image classification and there are only a few recent studies on semantic segmentation. In this study, we deal with DG for semantic segmentation. Unlike DA, DG does not have access to the target domains during the learning process. To make reliable predictions on various ‘unknown’ target domains, most existing studies focus on whitening [7], normalizing [41], and diversifying [22, 45, 62] styles to avoid overfitting to the style of the source domain. This paper focuses on extending both the content and style of the source domain to the wild [11], enabling networks to learn domain-generalized semantic features from diversified contents and styles.

2.2. Contrastive Learning

Contrastive learning [9, 39] is a strategy that minimizes the distance from a positive sample and maximizes the distance from a negative sample in the embedding space. Recently, He et al. [18] used a dynamic dictionary with a queue and Chen et al. [3] used two views of the same image as a positive pair to learn visual representations. To diversify a positive pair, a recent work [12] proposed to use the positive’s nearest neighbor in the latent space as a positive. After supervised contrastive learning [27] has been proposed, there are recent efforts to apply contrastive learning to fully- and semi-supervised semantic segmentation [1, 65, 66]. To obtain positive samples, these works perform image augmentation or store features using label information in a memory bank [58]. These enhance class discrimination in the seen source domain but do not guarantee improving class discrimination in various unseen domains. To adapt contrastive learning to DG for semantic segmentation, we propose a learning method using the wild-stylized feature and its closest wild content as positive samples.

2.3. Free ImageNet

Most studies regard ImageNet [11] as free and use it to pre-train networks. The ImageNet pre-trained model is commonly used in various fields such as object detection [34, 68], semantic segmentation [8, 54], panoptic segmentation [6, 35, 59], and video object segmentation [51], and is considered to be the same basis. The ImageNet pre-trained model is also used in most DA and DG for semantic segmentation methods, and ImageNet is used to borrow
3. Proposed Method

In this section, we introduce four learning techniques consisting of Feature Stylization (FS), Content Extension Learning (CEL), Style Extension Learning (SEL), and Semantic Consistency Regularization (SCR) for learning domain-generalized features by extending both the content and style of the source domain to the wild. Our WildNet achieves superior generalization ability with them and the overall learning process is outlined in Fig. 2.

3.1. Problem Setup and Overview

Domain generalization (DG) aims to enhance the generalization capability on both the seen source domain $S$ and unseen target domains $\mathcal{T} = \{T_1, ..., T_N\}$. Let $\phi$ be a semantic segmentation model that outputs pixel-wise category predictions $p$ from image $x$. This model consists of a feature extractor $\phi_{\text{feat}}$ and classifier $\phi_{\text{cls}}$. In DG, when we train the model, we have access to the source domain training dataset $D^s = \{(x^s, y^s)\}$ while inaccessible to the target domains, where $x^s \in \mathbb{R}^{H \times W \times 3}$ is an image, $y^s \in \mathbb{R}^{H \times W \times K}$ is its pixel-wise label, and $K$ is a number of semantic categories. The baseline model is trained with the segmentation loss

$$ L_{\text{orig}} = -\frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} \sum_{k=1}^{K} y_{hwk} \log(\phi(x^s)). $$

In this paper, we focus on extending both the content and style of the source domain to obtain high generalization performance on unknown target domains $\mathcal{T}$. We utilize the unlabeled wild dataset $D^w = \{x^w\}$, which has various contents and styles. At each training iteration, a random pair of source and wild images is provided as input, and the style and content of the source image are extended to the wild domain $W$ in the feature space. With the help of the wild, our network naturally learns domain-generalized semantic information from a variety of contents and styles. After the training, the model is evaluated on validation sets of both the seen source domain $S$ and unseen target domains $\mathcal{T}$.

3.2. Feature Stylization

As style is related to feature statistics [14, 15, 24, 26, 31] and the distributional shift due to style differences lies...
mainly in shallow layers of networks [41], the styles of features can be diversified by adjusting the statistics of features from shallow layers. In this work, we diversify the styles of the source features with the help of wild styles by adding several AdaIN [24] layers to the feature extractor in the learning process. This enables us to augment source features to resemble wild styles without losing spatial information.

Let \( \phi_l \) be the \( l \)-th layer of networks \( \phi \) and let \( z_l \) be the feature output from \( \phi_l \) when image \( x \) is input into \( \phi \). To allow networks to learn domain-generalized semantic information from various wild-style features, we swap the style of the source feature \( z_l^s \) from the source image \( x^s \) with the style of the wild feature \( z_l^w \) from the wild image \( x^w \). In the \( l \)-th layer, we transfer the style of \( z_l^w \) to \( z_l^s \) and obtain the wild-stylized feature \( z_l^{sw} \) as

\[
z_l^{sw} = \sigma(z_l^w) \frac{z_l^s - \mu(z_l^s)}{\sigma(z_l^s)} + \mu(z_l^w)
\]

where \( \mu(z_l^s) \) and \( \sigma(z_l^s) \) are channel-wise mean and standard deviation of feature \( z_l \) respectively. Because the distribution of \( z_l^s \) is re-normalized with channel-wise statistics of \( z_l^w \), the style of \( z_l^{sw} \) is swapped to the wild-style while maintaining the spatial information.

The wild-stylized feature \( z_l^{sw} \) is input into layer \( l + 1 \) and \( z_{l+1}^{sw} = \phi_{l+1}(z_{l+1}^{sw}) \) is output from the layer. \( z_{l+1}^{sw} \) can be swapped repeatedly in the style of \( z_{l+1} \) as

\[
z_{l+1}^{sw} := \sigma(z_{l+1}^{sw}) \frac{z_{l+1}^s - \mu(z_{l+1}^s)}{\sigma(z_{l+1}^s)} + \mu(z_{l+1}^{sw})
\]

By the above equation, feature \( z^{sw} \) is swapped in the style of \( z^w \) over multiple layers. As the layer deepens, semantic information should be captured more important than style, so FS applies only to some shallow layers in this work.

3.3. Content Extension Learning

In this subsection, we propose to extend the contents in the source domain to the wild. One of the reasons for overfitting to the source domain is that networks overlearn a limited amount of source content. We address this issue by increasing the intra-class content variability with content extension in the latent embedding space. To this end, we add a projection head \( \phi_{proj} \) independently of the classification head \( \phi_{cls} \) after the feature extractor \( \phi_{feat} \) and extend the source contents to the wild in the embedding space.

When a source image \( x^s \) and wild image \( x^w \) enter the feature extractor \( \phi_{feat} \), it outputs the source feature \( z^s \), wild-stylized source feature \( z^{sw} \), and wild feature \( z^w \). The projection head \( \phi_{proj} \) receives features \( z^s, z^{sw}, \) and \( z^w \) and outputs pixel-level projected content features \( z_{proj}^s, z_{proj}^{sw}, \) and \( z_{proj}^w \) respectively. All projected features are normalized by \( z = z/\text{max}([\|z\|_2, \epsilon]) \) along the channel dimension. At each training iteration, \( z_{proj}^{sw} \) is flattened and stored in the wild-content dictionary \( Q \in \mathbb{R}^{C_q \times N_q} \) where \( C_q \) is the number of channels of projected features and \( N_q \) is the dictionary size. Our model uses the dynamic dictionary structure

![Visualization of source-to-wild matching pixels on the cropped source and wild images.](image)

Figure 3. Visualization of source-to-wild matching pixels on the cropped source and wild images. We extend the source content to the wild content closest to the wild-stylized source content corresponding to the source pixel. The source content is encouraged to come close to the wild content in the embedding space. This improves the generalization ability of our model to unseen contents.

...in [18] as \( Q \) without a momentum update. We diversify contents by extending the source contents to the wild-stylized source contents and then to the wild domain using \( Q \).

Here we focus on that the projected source feature \( z_{1}^s \) and projected wild-stylized feature \( z_{1}^{sw} \) corresponding to the \( i \)-th pixel of the source image \( x^s \) contain exactly the same semantic information, but content perturbation exists. In order to obtain reliable semantic information from unseen contents, networks should be able to cluster contents containing the same semantic information, distinguishing them from contents containing other semantic information. To achieve this objective, we adapt the contrastive learning strategy [39] to pixel-level instances in a supervised manner and define the source content extension loss for the \( i \)-th pixel as follows:

\[
L_{SCE}^{i} = -\log \frac{\psi(z_i^s, z_i^{sw})}{\psi(z_i^s, z_i^{sw}) + \sum_{j=1}^{N_q} \mathbb{1}_{ij} \psi(z_i^s, z_j^{sw})}
\]

where \( \mathbb{1}_{ij} \) is the negative pixel indicator that equals 1 if \( y_i^s \) and \( y_j^s \) are different and 0 if they are the same, \( N_q \) is the number of pixels and the temperature parameter \( \tau \) is set to 0.07. We train the model only with reliable samples, ignoring ambiguous positive and negative samples by excluding pixels of unknown classes and pixels in other positions of the same class. Eq. (4) encourages \( z_i^s \) and \( z_i^{sw} \) to be close, while also encouraging \( z_i^s \) to move away from all negative class contents. Then, the pixel-wise loss can be applied to the entire source image by

\[
L_{SCE} = \frac{1}{N_s} \sum_{i=1}^{N_s} L_{SCE}^{i}
\]

Eq. (6) encourages networks to make generalized predictions by reducing the distance in the embedding space between source contents and wild-perturbed source contents.

Next, we further extend the source contents to the wild by utilizing the wild-content dictionary \( Q \). In the learning process, \( Q \) stores diverse pixel-level wild contents, which may not exist in the source domain. Thus, if we carefully select wild-content with semantic information that each pixel

---

1The subscript \( proj \) is sometimes omitted for convenience.
needs to learn and then use it to train networks, networks become more robust to wild-content perturbations. Since there is no class information in the wild set $D^w$, Eq. (4) cannot be directly applied to this wild content extension. We address this issue from the perspective that similar semantic contents will be located close to each other in the embedding space. Inspired by [12], we take the wild content $z^w_k$ closest to the wild-stylized source content $z^sw$ from $Q$ as

$$z^w_k = \arg \min_{q \in Q} ||z^w_k - q||_2$$

(7)

and encourage the source content $z^s_i$ to come close to it. Since $z^w_k$ and $q$ are normalized early on, Eq. (7) can be calculated efficiently using a dot product and rewritten as

$$z^w_k = \arg \max_{q \in Q} (z^w_k \cdot q).$$

(8)

Fig. 3 shows the wild content $z^w_k$ matched to the source content $z^s_i$ using the stylized source content $z^sw$. In this way, we provide various contents of the wild to the networks without category information. Now Eq. (4) can be adapted to the wild content extension as follows:

$$\mathcal{L}_{WCE} = -\log \frac{\psi(z^s_i, z^w_k)}{\sum_{j=1}^{N_s} \sum_{k=1}^{K_s} \psi(z^s_i, z^sw_j)}.$$ 

(9)

In the wild content extension, we reuse the negative samples of the source content extension. Some negative extension approaches may give better performance and we left this for future work. Then we can apply the pixel-wise wild content extension loss to the entire source image by

$$\mathcal{L}_{WCE} = \frac{1}{N_s} \sum_{i=1}^{N_s} \mathcal{L}^i_{WCE}.$$ 

(10)

By combining the source content extension loss and wild content extension loss, the CEL loss is defined as

$$\mathcal{L}_{CEL} = \mathcal{L}_{SCE} + \mathcal{L}_{WCE}.$$ 

(11)

Our model learns to capture generalized semantic information from diverse contents by using the proposed CEL loss.

### 3.4. Style Extension Learning

Another reason for overfitting to the source domain is that networks overlearn a limited amount of the source style [36, 43, 67]. To address this issue, FS has diversified the styles of the source feature with the help of the wild. Interestingly, the style of the source features has changed while preserving spatial information, but networks fail to predict semantic categories from the wild-stylized feature as shown in Fig. 4f. In this subsection, we propose SEL for adapting networks to diversified styles. SEL aims to allow networks to naturally adapt to various styles by learning task-specific information from the wild-stylized feature.

When the wild-stylized source feature $z^sw$ enters the classification head $\psi_{cls}$, it outputs the pixel-wise softmax segmentation map $p^sw \in \mathbb{R}^{H \times W \times K}$. Because $z^sw$ is the wild-stylized source feature in which the style of the feature from $x^s$ has been swapped with that of the feature from $x^w$, networks must predict the semantic label $y^s$ of $x^s$ from $z^sw$. For this objective, we train networks by minimizing the following SEL loss:

$$\mathcal{L}_{SEL} = -\frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} \sum_{k=1}^{K} y^s_{hw} \log (p^sw_{hw,k}).$$ 

(12)

Our model learns task-specific information from the wild-stylized features by applying the SEL loss. This enables our model to naturally learn domain-generalized semantic information from various styles.

### 3.5. Semantic Consistency Regularization

For high generalization capability on unseen domains, the classifier should capture consistent semantic information from features [5, 25], even if there are perturbations in both the style and content. However, as shown in Figs. 4e and 4f, the predicted result $p^sw$ of the wild-stylized source feature $z^sw$ differs from the predicted result $p^s$ of the source feature $z^s$. Although SEL allows networks to learn task-specific information from $z^sw$, this does not guarantee that $p^sw$ and $p^s$ are identical. To address this issue, we propose SCR that regularizes networks to capture consistent semantic information even when both the content and style of the source domain are extended to the wild. SCR aims to train networks so that the predicted probability distributions $p^sw = \phi_{cls}(z^sw)$ from the wild-stylized source features get closer to the $p^s = \phi_{cls}(z^s)$ from the source features. To this end, we adapt the Kullback-Leibler (KL) divergence loss as

$$\mathcal{L}_{SCR} = -\frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} \sum_{k=1}^{K} y^s_{hw} \log (p^sw_{hw,k})/p^s_{hw,k}.$$ 

(13)

With the SCR loss, our model learns consistent semantic information even with perturbations of style and content by the proposed wild extension methods.
4. Experiments

4.1. Datasets

Real semantic segmentation datasets. Cityscapes [10], BDD100K [61] and Mapillary [37] consist of 2975, 7000, and 18000 images for train set and 500, 1000, and 2000 for validation set. We consider 19 classes that are compatible with other datasets. In all of the tables, C, B, and M denote Cityscapes, BDD100K, and Mapillary, respectively.

Synthetic semantic segmentation datasets. GTAV [46] contains 24966 images rendered from the Grand Theft Auto V game engine. It has 12403, 6382, and 6181 images for train, validation, and test sets, respectively. SYNTHIA [49] contains 9400 images and we split it into 6580 and 2820 images for train and validation sets, following [7]. In all tables, G and S denote GTAV and SYNTHIA, respectively.

Wild dataset. ImageNet [11] is a large-scale image classification dataset used for network pre-training in most studies. In this paper, we use images without class labels as wild domain data. The generalization performance according to the number of images used for training our WildNet is presented in Table 6a.

4.2. Experimental Setup

We conduct extensive experiments over five different semantic segmentation datasets and report the mean intersection over union (mIoU) score on several domain generalization scenarios: GTA V → {Cityscapes, BDD100K, Mapillary, SYNTHIA, GTAV} and Cityscapes → {GTAV, BDD100K, Mapillary, SYNTHIA, Cityscapes}. For fair comparisons with other DG methods, we re-implement IBN-Net [41] and RobustNet [7] on our baseline models and † denotes our re-implemented models. Our model is trained on one source domain train set (GTAV or Cityscapes) and validated on five domain validation sets (four unseen domains and one seen domain). To show the overall domain generalization performance, we additionally report the average value of mIoU on five domain validation sets (Avg). In all of the tables, the best results for each domain are marked in bold.

4.3. Implementation Details

We adapt ResNet-50, ResNet-101 [19], and VGG-16 [52] with DeepLabV3+ [2] as segmentation networks, and all backbones are pre-trained on ImageNet [11]. In the ResNet-based models, we use the SGD optimizer [47] with a momentum of 0.9 and weight decay of 5e-4. The initial learning rate is set to 2.5e-3 and is decreased using the polynomial policy with a power of 0.9. We train the models for 60K iterations with a batch size of 8. In the VGG-based models, we use the Adam optimizer [28] with a momentum of (0.9, 0.99). The initial learning rate is set to 1e-5 and the batch size is set to 8. Following [7], we apply random scaling within a range of [0.5, 2.0] and random cropping with a size of 768×768. The output size of the projection head is 192×192 and we use uniformly sampled 64×64 size feature maps for CEL to prevent memory issues. For the diversity of the wild content dictionary, the wild feature maps are stored after uniform sampling with a size of 16×16. The FS layer replaces first batch normalization and is added immediately after the addition operation of the first two residual layers, projection head, and wild-content dictionary are stored after uniform sampling with a size of 16×16. The output size of the projection head is 192×192 and we use uniformly sampled 64×64 size feature maps for CEL to prevent memory issues. For the diversity of the wild content dictionary, the wild feature maps are stored after uniform sampling with a size of 16×16. The FS layer replaces first batch normalization and is added immediately after the addition operation of the first two residual blocks in ResNet, and it added right after the first ReLU after the first three maxpool layers in VGG. After training, all FS layers, projection head, and wild-content dictionary are removed, and our model can be applied to multiple unseen domains without further training on the target domains.

4.4. Comparison with DG methods

We compare our results with existing DG methods: IBN-Net [41], DRPC [62], ASG [4], FSDR [22], RobustNet [7], and GLTR [45]. Table 1 shows the generalization performance of the ResNet-50 model trained on GTAV. We evaluate models on five validation sets consisting of four unseen domains, including the Cityscapes, BDD100K, Map-
Table 3. Comparison of mIoU(%) using VGG-16 as backbone under the domain generalization setting $G \rightarrow \{C, B, M, S\}$.

<table>
<thead>
<tr>
<th>Methods</th>
<th>G</th>
<th>B</th>
<th>M</th>
<th>S</th>
<th>C</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline [41]</td>
<td>29.40</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>64.50</td>
</tr>
<tr>
<td>IBN-Net [41]</td>
<td>39.70</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>67.00</td>
</tr>
<tr>
<td>Baseline [7]</td>
<td>42.55</td>
<td>44.96</td>
<td>51.68</td>
<td>23.29</td>
<td>-</td>
<td>77.51</td>
</tr>
<tr>
<td>RobustNet [7]</td>
<td>45.00</td>
<td>50.73</td>
<td>58.64</td>
<td>26.20</td>
<td>-</td>
<td>76.41</td>
</tr>
<tr>
<td>Baseline</td>
<td>40.50</td>
<td>42.35</td>
<td>20.67</td>
<td>8.08</td>
<td>-</td>
<td>76.30</td>
</tr>
<tr>
<td>1IBN-Net [41]</td>
<td>45.28</td>
<td>46.61</td>
<td>56.78</td>
<td>26.41</td>
<td>-</td>
<td>75.47</td>
</tr>
<tr>
<td>1RobustNet [7]</td>
<td>45.28</td>
<td>48.21</td>
<td>56.97</td>
<td>26.59</td>
<td>-</td>
<td>74.90</td>
</tr>
<tr>
<td>WildNet (Ours)</td>
<td>47.01</td>
<td>50.94</td>
<td>58.79</td>
<td>27.95</td>
<td>-</td>
<td>75.59</td>
</tr>
</tbody>
</table>

Table 4. Comparison of mIoU(%) using ResNet-50 as backbone under the domain generalization setting $G \rightarrow \{C, B, M, S\}$.

<table>
<thead>
<tr>
<th>$L_{\text{orig}}$</th>
<th>$L_{\text{CEL}}$</th>
<th>$L_{\text{SEL}}$</th>
<th>$L_{\text{SCR}}$</th>
<th>C</th>
<th>B</th>
<th>M</th>
<th>S</th>
<th>G</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>35.16</td>
<td>29.71</td>
<td>31.29</td>
<td>27.97</td>
<td>71.17</td>
<td>39.06</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>41.25</td>
<td>35.95</td>
<td>40.06</td>
<td>31.26</td>
<td>68.75</td>
<td>43.46</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>43.61</td>
<td>38.69</td>
<td>43.17</td>
<td>31.40</td>
<td>70.52</td>
<td>45.48</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>44.62</td>
<td>38.42</td>
<td>46.09</td>
<td>31.34</td>
<td>71.20</td>
<td>46.33</td>
</tr>
</tbody>
</table>

Table 5. Effect of the proposed losses on the domain generalization setting $G \rightarrow \{C,B,M,S,G\}$ using ResNet-50 as backbone in mIoU(%). Losses $L_{\text{orig}}$, $L_{\text{CEL}}$, $L_{\text{SEL}}$, and $L_{\text{SCR}}$ are defined in Eq. (1), Eq. (11), Eq. (12), and Eq. (13), respectively.

4.5. Ablation Studies

In this subsection, extensive experiments with ResNet-50 model on the DG scenario from GTA V to Cityscapes, BDD100K, Mapillary, SYNTHIA, and GTA V are conducted to study the effectiveness of each component in the proposed method. Table 5 shows the effect of the proposed losses on domain generalization. The baseline model trained only with $L_{\text{orig}}$ overfits the source domain and has poor performance on unseen domains. Even with only $L_{\text{CEL}}$ applied, our model achieves an Avg of 43.46% with +4.40% improvement. This shows the importance of content diversification that is overlooked in many studies. Further, we make the wild-styled features learn task-specific information with $L_{\text{SEL}}$ to achieve an Avg of 45.48%, and regularize the model to learn consistent semantic information with $L_{\text{SCR}}$, finally achieving an Avg of 46.33%. Next, we conduct more ablations for important components.

Number of wild images. In Table 6a, the number of wild images used to train our model is considered. Even if only 10 wild images are used, the generalization performance is significantly enhanced by +5.54% compared with the baseline by preventing overfitting to the source domain. Moreover, the generalization performance of the model gradually improves as the number of wild images used increases. This shows that the extension of both content and style to the wild helps networks to learn domain-generalized semantic features.

Amount of FS. Table 6b shows the influence of the amount of FS on generalization performance. By replacing only the first batch normalization with FS, we can extend contents and styles to the wild based on diversified stylized features and improve generalization performance compared to baseline by +5.23%. Adding FS to some shallow layers boosts performance further. However, applying FS to deeper layers degrades performance slightly, as semantic content should be captured more important than style as the layer deepens. A suitable amount of FS, which does not disturb the semantic information, helps to train the generalized model by augmenting the source features to have various wild styles.

Size of wild content dictionary. Table 6c shows sensitivity to the size of the wild content dictionary. Extending the source content to the wild improves generalization performance, and even when extended to wild content within a mini-batch of size 2048 without a content dictionary, our model achieves higher generalization performance than without content extension. We take size of 393216.

FS with wild style. In Table 6d, we show the effect of FS using statistics of wild features on the generalization performance of the model. To apply FS without the help of the wild, the mean and standard deviation of the source features were multiplied by random values in the range [0.5, 1.5] and then used instead of the statistics of the wild features. The
random FS improves performance compared to baseline by +4.48%, which shows the importance of diversifying styles. Furthermore, wild FS demonstrates that learning a natural style of the wild is much better with a gain of +7.27%.

**Sampling methods.** By using sampled feature maps, CEL stores various wild contents in the fixed-size dictionary and reduces memory consumption due to pixel-level contrastive loss calculations. Since two adjacent pixels have almost similar semantic information, uniform sampling makes learning more diverse contents than random sampling, leading to high generalization performance as shown in Table 6e.

5. Discussion

5.1. Qualitative Analysis

To analyze the wild content extension, we visualize the wild contents closest to the stylized source contents in Fig. 5. As can be seen in the figure, the source content is extended to wild content with semantic information similar to itself, e.g., the road under the car, the wheel of the bicycle, and the head of a man wearing a hat. Our model learns domain-generalized features by inducing source content closer to these wild content in the feature space. With learning various wild contents, WildNet makes reliable predictions on unseen contents. Figs. 1 and 4 show segmentation results and visualization of wild-stylized features, and further analysis is provided in the supplementary material.

5.2. Limitations and Future Works

We have shown that the source content extends to wild content with semantic information similar to itself. However, similar semantic information in the two contents does not guarantee that the classes of the two contents are always the same, as observed for the rider of Fig. 5. Extending the rider with the hat to the person with the hat may bridge between the rider-class and person-class. Our future works will involve positive content selection using predicted class probabilities on wild images and negative content extension to further boost the class discrimination ability.

6. Conclusion

We presented WildNet which exploits unlabeled wild images for domain-generalized semantic segmentation. Our approach effectively extends style and content from source to wild, resulting in drastic performance improvement even we leverage 10 wild images. In contrast to previous studies that exploit generalization cues only from style, we additionally exploit the potential to generalize domain from content. We thoroughly ablated to demonstrate the efficacy of our WildNet and achieved superior segmentation performance under several domain generalization scenarios. We believe that our approach provides an opportunity to utilize huge amounts of unlabeled data for domain generalization.

**Acknowledgement.** This research was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF-2019R1A2C1007153).
References


[23] Lei Huang, Dawei Yang, Bo Lang, and Jia Deng. Decorrelated batch normalization. In CVPR, pages 791–800, 2018. 1


