Self-Supervised Learning of Domain Invariant Features for Depth Estimation

Hiroyasu Akada[1,2] Shariq Farooq Bhat[1] Ibraheem Alhashim[3] Peter Wonka[1]

[1]KAUST, [2]Keio University [3]National Center for Artificial Intelligence (NCAI), Saudi Data and Artificial Intelligence Authority (SDAIA)

hiroyasu5959@keio.jp shariq.bhat@kaust.edu.sa {ibraheem.alhashim, pwonka}@gmail.com

Abstract

We tackle the problem of unsupervised synthetic-to-real domain adaptation for single image depth estimation. An essential building block of single image depth estimation is an encoder-decoder task network that takes RGB images as input and produces depth maps as output. In this paper, we propose a novel training strategy to force the task network to learn domain invariant representations in a self-supervised manner. Specifically, we extend self-supervised learning from traditional representation learning, which works on images from a single domain, to domain invariant representation learning, which works on images from two different domains by utilizing an image-to-image translation network. Firstly, we use an image-to-image translation network to transfer domain-specific styles between synthetic and real domains. This style transfer operation allows us to obtain similar images from the different domains. Secondly, we jointly train our task network and Siamese network with the same images from the different domains to obtain domain invariance for the task network. Finally, we fine-tune the task network using labeled synthetic and unlabeled real-world data. Our training strategy yields improved generalization capability in the real-world domain. We carry out an extensive evaluation on two popular datasets for depth estimation, KITTI and Make3D. The results demonstrate that our proposed method outperforms the state-of-the-art on all metrics, e.g. *by 14.7% on Sq Rel on KITTI. The source code and model weights will be made available.*

1. Introduction

Unsupervised domain adaptation (UDA) for single image depth estimation deals with the following problem: given a corpus of synthetic data (RGB images) and their labels (depth maps) together with real data (RGB images) without labels, the goal is to train a *task network* (depth estimation network) to learn from the synthetic data in such a way so that it generalizes to real data. However, the large domain gap between real and synthetic data poses a significant challenge.

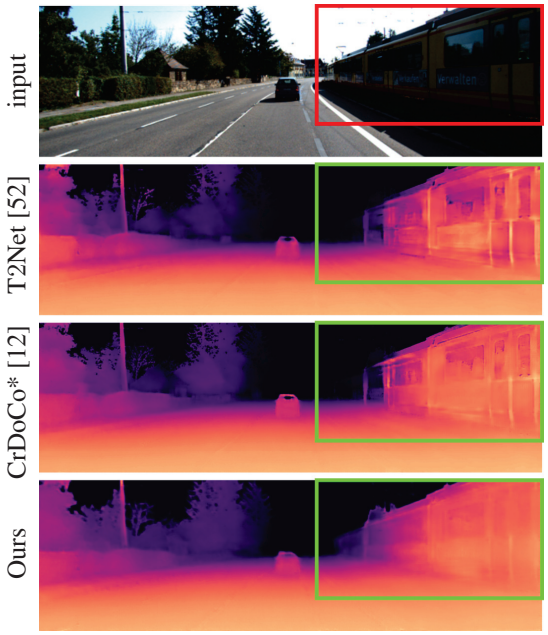

Figure 1: Predicted depth maps from our proposed method and comparison methods, T2Net [52] and CrDoCo* [12]. Our method is better at estimating consistent depth values on objects' surfaces than existing methods.

We can identify three different strategies in UDA for single image depth estimation. The first strategy is to translate synthetic images to the real domain using an image-to-image translation (I2I) network and enforce a task network to learn representations only for the real domain [52, 35]. The second strategy also uses I2I for style transfer but enforces two separate task networks to learn separate representations for synthetic and real data respectively [12, 50,

38]. In this strategy, the task network in the synthetic domain is trained in a supervised manner and is used to guide the other task network in the real domain. The third strategy is to train a task network to learn domain invariant features. The goal of such a strategy is to train a task network that can take both real and synthetic images as input and produce similar features for similar images from either domain. Our work follows this strategy as we try to train a single task network that can take input from both real and synthetic domains.

While the difference between these three strategies by itself is not that significant, we identified an important bottleneck that is common in all three of these strategies. The problem is that the task network needs to be initialized with reasonable (pre-trained) weights that work well in the real domain for the encoder, as indicated by [8], as well as the decoder. The initialization of the encoder can be easily done using the pre-trained weights on ImageNet [13] that are widely available. However, initializing the decoder is challenging. We found an elegant way to adapt recent works in self-supervised representation learning (SSRL) [10] to learn domain invariant features for the decoder. We propose several components that enable this adaptation. First, we replace data augmentation by an image-to-image translation network that learns mappings from synthetic-to-real and vice-versa. This lets us utilize similar images from either domain and use SSRL to enforce our task network to learn domain invariance. Second, we extend self-supervised learning to extract domain invariant representations for the decoder, coupled with the encoder pre-trained on ImageNet. Third, we use a channel-wise projector and predictor on high-resolution decoder features for self-supervision.

Our proposed framework consists of three stages: a style transfer stage, where we train the I2I networks, a self-supervised representation learning (SSRL) stage, which is used to learn a good initialization for the task network decoder, and a depth estimation stage, where the task network is fine-tuned using both labeled synthetic and unlabeled real-world datasets.

Extensive experiments show that our proposed method can lead to better generalization performance on the target domain and outperforms state-of-the-art UDA methods for single image depth estimation on two popular datasets, KITTI [19] and Make3D [2] on all metrics. In summary, we make the following main contributions:

- We propose a novel UDA framework that enables an encoder-decoder monocular depth estimator to learn domain invariant representations and thus, generalizes well to the target domain.
- We devise a pre-training strategy for the decoder using self-supervised learning.
- We demonstrate that our proposed method achieves the new state-of-the-art in UDA for single image depth estimation on two popular datasets, KITTI [19] and Make3D [2].

2. Related Work

2.1. Supervised Depth Estimation

Supervised learning methods are currently the top-performing approaches for the depth estimation task [14, 1, 17, 15, 48, 31, 3]. BTS [31] proposes to utilize local planar guidance layers to effectively guide feature maps to full resolution instead of using conventional upsampling layers in their decoder blocks. DORN [15] considers depth estimation as a classification task by dividing the depth range into multiple bins that are fixed with predetermined widths. More recently, AdaBins [3], expands on DORN by introducing a transformer-based architecture to dynamically change the depth bins based on the input. While these supervised methods show promising results, they require fully labeled real-world datasets that are hard to prepare.

2.2. UDA for Depth Estimation

Unsupervised domain adaptation methods aim to learn from synthetic depth maps that are much easier to produce. The core idea of such methods is to align the data distribution between a synthetic dataset with full labels (*i.e*. source domain) and a real-world dataset without labels (*i.e*. target domain). In depth estimation, these works can be divided into two groups. The first group applies GAN [23]-based image translation techniques with extra information, such as real-world stereo pair images [50, 38, 43, 37], semantic segmentation images [35], monocular video [49], pose data [30], shading information [4], or a small amount of real ground truth labels [51]. The second group, including our work, follows a fully unsupervised approach without such additional information. These methods also utilize adversarial training [23] to align data distributions at both feature and image levels [52, 12].

Our work is based on the state-of-the-art UDA method, CrDoCo [12] and we introduce a novel training strategy that allows the depth estimator to learn domain invariance in a self-supervised manner.

2.3. Representation Learning

Representation learning has been an actively researched domain in deep learning. One line of research in this area is contrastive learning. Contrastive learning methods introduce a contrastive loss [26] and a projection multi-layer perceptron (MLP) to make latent feature representations similar for similar input pairs (*i.e*. positive pairs) and dissimilar for dissimilar input pairs (*i.e*. negative pairs) [6, 7, 27, 9, 11]. Although contrastive learning have showed promising results, their training is prohibitively expensive within a

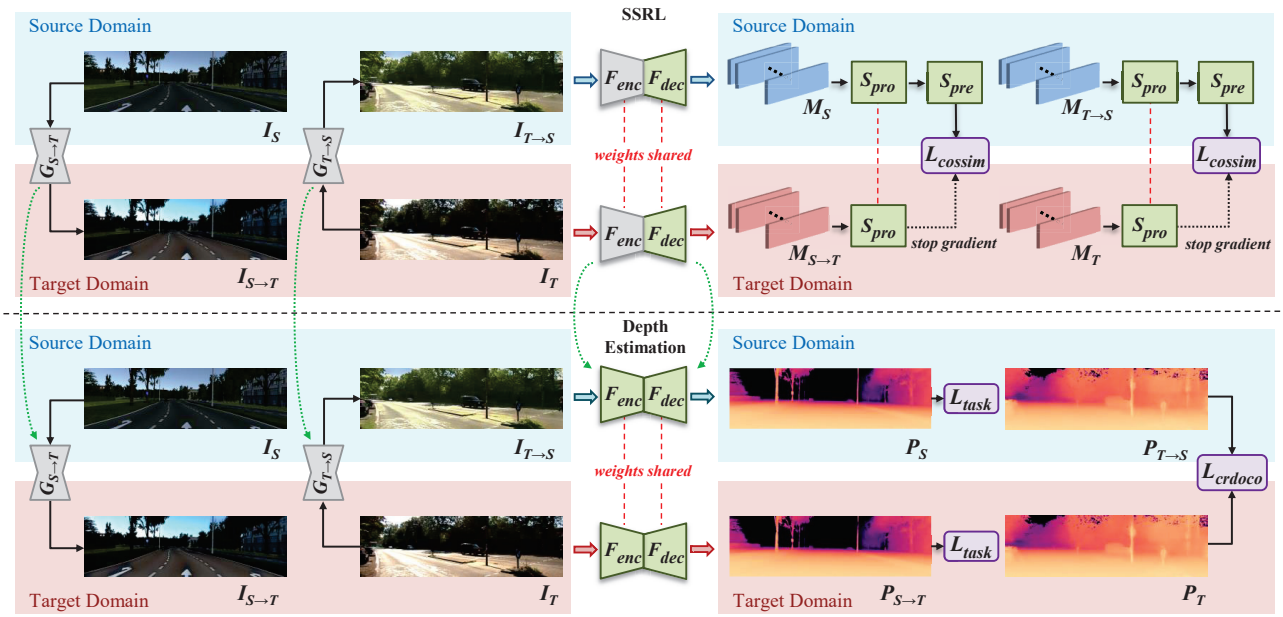

Figure 2: Overview of our proposed SSRL stage. Modules in green are trainable and modules in gray are non-trainable. Green dot arrows indicate that modules will be used in the next stage. As a preliminary step, we first train the image translation networks $G_{S\to T}$ and $G_{T\to S}$ in the style transfer stage as in most UDA methods including CrDoCo [12]. The trained image translation network $G_{S\to T}$ and $G_{T\to S}$ are used in the SSRL and depth estimation stages with their weights fixed. As our main contribution, we introduce the self-supervised representation learning (SSRL) stage to enforce our task network to learn a domain invariant representation via our Siamese network in a self-supervised way. Note that L_{cossim} is a symmetrized loss using corresponding feature maps (*e.g.* M_S and $M_{S\to T}$). Lastly, we fine-tune the final task network using the synthetic and real datasets in the depth estimation stage. Please see Section 3 and 4 for more details.

UDA framework due to the use of a large batch size. Other representation learning methods, such as BYOL [25], utilizes a momentum-based network coupled with a prediction MLP without the need of negative pairs. In addition, SimSiam [10] significantly simplifies the previously proposed networks and introduces only a prediction MLP and stop-gradient operation for representation learning. More recently, PixPro [46] is proposed to utilize pixel-level consistency on downstream dense prediction tasks, such as semantic segmentation.

We build our framework based on the prediction network and stop-gradient operation [10] to be trainable on images in one domain and the corresponding translated images in the other domain to obtain domain invariant representations.

3. UDA for single image depth estimation

We address the task of UDA for single image depth estimation. In terms of data availability, we have access to a set of synthetic RGB images and depth maps as source image data $I_S \in X_S$ and source label data $I_{S,lab} \in X_{S,lab}$, and a set of real-world RGB images as target image data $I_T \in X_T$. We aim at training a depth estimator without using real-world depth maps so that it is able to estimate an accurate depth map given a real-world single RGB image.

To this end, we propose a novel UDA framework that allows an encoder-decoder based depth estimator to learn a domain invariant representation via adversarial training and representation learning. Our main contribution lies in the introduction of the SSRL stage into current state-of-the-art UDA methods that are usually based on the style transfer stage and the depth estimation stage (task-specific stage) only. We build on CrDoCo [12] as the current state-of-the-art UDA method. In the following, we describe our method in more details. First, we describe our modifications to CrDoCo to make it better compatible with our SSRL stage. Second, we provide details on our SSRL stage in Section 4.

3.1. Review of CrDoCo

CrDoCo [12] is composed of the style transfer and depth estimation stages. In the style transfer stage, CrDoCo [12] trains a bidirectional image-to-image translation network [54] to learn a mapping between source and target domains via adversarial training. In the depth estimation stage, the image translators are then used together with two feature discriminators and two task networks [52] that are deployed in source and target domains. Specifically, the task network in each domain takes as inputs images with

domain-specific styles (*e.g*. the task network in the source domain takes as inputs source images and translated images from the target domain). Along with label supervision, CrDoCo [12] introduces a cross-domain consistency loss L_{crdoco} to enforce prediction level alignment for unlabeled images. Please see [12] for more details on their architecture.

3.2. Modifications

Following CrDoCo [12], we adopt the bidirectional image-to-image translation network [54] in our framework, *i.e*. two generators $G_{S\to T}$, $G_{T\to S}$, and two discriminators D_S, D_T. However, we remove the two feature discriminators for faster training and modify their task networks into a weights-shared model F, which is crucial to leverage our proposed SSRL stage described in Section 4.

We also design our encoder-decoder based task network with EfficientNetB5 [42] as an encoder instead of using older encoders [52, 12, 50, 38]. In addition, we pretrain the encoder on ImageNet [13], which is a recent de facto standard for many UDA methods for depth estimation [1, 17, 15, 48, 31, 3] as well as semantic segmentation [34, 41, 47, 32, 36, 33, 5, 56, 45, 39, 55].

For the style transfer and depth estimation stages, we adopt the same loss functions as CrDoCo [12]. Please see [12] and our supplementary material for more details on each stage and on the loss functions.

4. Proposed SSRL stage

In this section, we describe our proposed SSRL stage as shown in Fig. 2. In the SSRL stage, we utilize the image-to-image translation network $G_{S\to T}$ and $G_{T\to S}$, the encoder-decoder based image-to-depth task network F, and a Siamese network S. Note that the image-to-image translation networks $G_{S\to T}$ and $G_{T\to S}$ are pre-trained in the style transfer stage as in CrDoCo [12].

The purpose of the SSRL stage is to force F to learn a latent feature representation that is invariant to synthetic (*i.e*. source) and real (*i.e*. target) domains so that F can generalize well in the real domain. In the architecture of F, its encoder F_{enc} is initialized by ImageNet [13] pre-training and thus, it is already good at extracting features in the real domain for dense prediction tasks as indicated in [20, 21, 8]. Therefore, we aim to train its decoder F_{dec} jointly with S so that F_{dec} can also perform well in the real domain. Here, we fix the weights of F_{enc}.

4.1. Components of the proposed SSRL stage

The first adaptation from traditional work in SSRL is to replace data augmentation with style transfer. Instead of learning identical representations that are invariant to a given set of image transformations such as color shifts, we aim to learn domain invariant representations for synthetic and real images. We can use two types of image pairs as input: $I_{S\to T}$ together with I_S and $I_{T\to S}$ together with I_T. In our notation, $I_{S\to T}$ and $I_{T\to S}$ are the outputs of translation network $G_{S\to T}$ and $G_{T\to S}$, respectively.

Different from previously proposed representation learning methods [25, 10, 46], we aim to learn representations in the decoder. This leaves two design choices to explore. First, we need to decide which representations in the decoder to target. The final output *i.e*. the depth values are no longer useful as representations. The obvious choice would be the features in the last layer before the output layer, but earlier features could also yield better results. The decision of which layer to use for SSRL is taken using an empirical study described in Sec. 5.6. Second, different from previous work, our feature maps have a much higher resolution. In our recommended architecture, we extract corresponding high dimensional feature maps before the final output layer (convolutional layer) from F_{dec}, *i.e*. M_S, M_T, $M_{S\to T}$, and $M_{T\to S}$ as shown in Fig 2. The size of each feature map is $C \times W \times H$, where $C = 12$, $W = 960$, $H = 288$ in our experiment. Here, we consider M_S and $M_{S\to T}$ (or M_T and $M_{T\to S}$) as a set of features of augmented views from the same scene. In the following parts, we only describe the domain invariant representation learning on M_S and $M_{S\to T}$ for brevity. The second case, when using M_T and $M_{T\to S}$, works in the same way.

We also developed a component that corresponds to the projector in traditional SSRL [25, 10, 46]. In contrast to this traditional setting, we do not reduce the features dimension. In our Siamese network S, the projector S_{pro} takes the feature maps M_S and $M_{S\to T}$ as inputs to output corresponding feature embeddings of the same size $C \times W \times H$. Similar to the traditional SSRL architecture, the predictor S_{pre} aims to transform the embedding of one view and matches it to the other view. We denote these outputs as $z_S \triangleq S_{pro}(M_S)$, $p_S \triangleq S_{pre}(S_{pro}(M_S))$, $z_{S\to T} \triangleq S_{pro}(M_{S\to T})$, and $p_{S\to T} \triangleq S_{pre}(S_{pro}(M_{S\to T}))$. Here, the S_{pro} and S_{pre} share the weights for M_S and $M_{S\to T}$ as well as M_T and $M_{T\to S}$. However, our predictor architecture incorporates a bottleneck. This follows previous architectures that also incorporate a dimension reduction component in the predictor. Different from previous work, both our projector and predictor are per-pixel MLPs (convolutions with a 1×1 kernel).

4.2. Loss function of the proposed SSRL stage

We follow the previous methods [25, 10] by using a negative cosine similarity as our loss function in the SSRL stage. However, unlike the previous methods, the latent features from F_{dec} are high dimensional and therefore, we assume that it is beneficial to take into consideration the correspondence between the features at the level of pixels.

Method	Train	Dataset	Evaluation resolution	Lower is better				Higher is better		
				Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
BTS [31]	S	K(I+D)	1241×376	0.059	0.245	2.756	0.096	0.956	0.993	0.998
AdaBins [3]	S	K(I+D)	1241×376	0.058	0.190	2.360	0.088	0.964	0.995	0.999
Monodepth2 [21]	SS	K(I) + video + stereo	1024×320	0.106	0.806	4.630	0.193	0.876	0.958	0.980
Monodepth2 [21]	SS	K(I) + video	1024×320	0.115	0.882	4.701	0.190	0.879	0.961	0.982
Johnston [29]	SS	K(I) + video	640×192	0.106	0.861	4.699	0.185	0.889	0.962	0.982
T^2Net [52]	U(DA)	vK(I+D)+K(I)	960×288	0.179	1.620	6.108	0.257	0.754	0.902	0.962
CrDoCo [12]	U(DA)	vK(I+D)+K(I)	960×288	0.232	2.204	6.733	0.291	0.739	0.883	0.942
CrDoCo* [12]	U(DA)	vK(I+D)+K(I)	960×288	0.174	1.439	5.701	0.241	0.770	0.914	0.967
Ours	U(DA)	vK(I+D)+K(I)	960×288	**0.168**	**1.228**	**5.498**	**0.235**	**0.771**	**0.921**	**0.973**

Table 1: Quantitative results on KITTI [19]. Methods, which use only synthetic image-depth pairs and real depth maps as training datasets, are marked in gray. For training, S: supervised, SS: self-supervised, U: unsupervised, and DA: domain adaptation. For dataset, K and vK represent KITTI and virtual KITTI, I and D indicate the use of RGB images and depth maps, stereo and video indicate the use of stereo and video information. The model with * is trained with the same hyper-parameters as our model.

Method	Train	Lower is better			
		Abs Rel	Sq Rel	RMSE	RMSE log
Zhou [53]	SS	0.383	5.321	10.470	0.478
DDVO [44]	SS	0.387	4.720	8.090	0.204
Monodepth2 [21]	SS	0.322	3.589	7.417	0.163
Johnston [29]	SS	0.297	2.902	7.013	0.158
T2Net [52]	U(DA)	0.337	4.767	8.735	0.128
CrDoCo [12]	U(DA)	0.606	14.221	13.92	0.209
CrDoCo* [12]	U(DA)	0.330	4.295	8.011	0.127
Ours	U(DA)	**0.309**	**3.567**	**7.401**	**0.119**

Table 2: Quantitative results on Make3D [2].

Based on this assumption, we define a symmetrized loss as

$$L_{cossim} = \frac{1}{2}(cossim(p_S, sg(z_{S\to T}))) + \frac{1}{2}(cossim(p_{S\to T}, sg(z_S))), \quad (1)$$

where sg is a stop gradient operation [10] and $cossim(a, b)$ is a pixel-wise negative cosine similarity function defined as

$$cossim(A, B) = \frac{1}{N}\sum_{i\in N}\left(-\frac{a_i}{\|a_i\|_2}\cdot\frac{b_i}{\|b_i\|_2}\right), \quad (2)$$

where a_i and b_i are feature vectors of the size $C \times 1 \times 1$ with the spatial index i of feature maps A and B (*i.e.* $a_i \in A$ and $b_i \in B$), and N is the total number of the spatial indices. Overall, we minimize L_{cossim} in a self-supervised manner in the SSRL stage, *i.e.* $L_{SSRL} = L_{cossim}$.

5. Experiments and results

5.1. Network architecture

We use CycleGAN [54] as our bidirectional image translation network $G_{S\to T}$, $G_{T\to S}$, D_S, and D_T. For our task network F, we adopt an encoder-decoder based architecture using EffcientNet-B5 [42] pre-trained on ImageNet [13] as the encoder. For our Siamese network, we follow [25, 10] to leverage a projector and a predictor. Specifically, our projector S_{pro} has 3 convolutional layers with 1×1 kernels, ReLU activation and Batch Normalization [28] after each layer except the last layer. Since the output feature maps from F_{dec} in the SSRL stage are the size of $C(=12) \times W \times H$, we set the channel dimension of each layer in the projector to C. Also, the predictor has 2 convolutional layers with 1×1 kernels, ReLU and Batch Normalization [28] applied only after the first layer. In our predictor S_{pre}, the input and output channel dimensions are set to C whereas the hidden layer's channel dimension is set to C/α, where α is a scaling factor. Here, we set $\alpha = 3$ in our experiments. Please see Table 5 for our ablation studies on α and implementation code for more details.

5.2. Datasets

In our experiments, we use KITTI [19] as the target domain dataset and virtual KITTI (vKITTI) [16] as the source domain dataset. KITTI is an outdoor scene dataset captured using a moving vehicle with a resolution of around 1241×376. We use a subset of 22,600 images for training as specified by Eigen *et al*. [14]. vKITTI provides 21,260 synthetic image-depth pairs generated from different virtual urban worlds. The maximum sensed depth in KITTI is on the order of 80m while vKITTI has more precise depth values to a maximum of 655.3m. Consistent with previous work [52], we remove 'fog' and 'rain' images, and clip the maximum depth in vKITTI to match that of KITTI, *i.e*. 80m.

For input resolution, we use relatively larger input size that is comparable with the state-of-the-art supervised methods (*e.g*. 704×352 cropped from 1241×376 for training and 1241×376 for testing) [31, 3] to see the current performance gap between UDA methods and supervised coun-

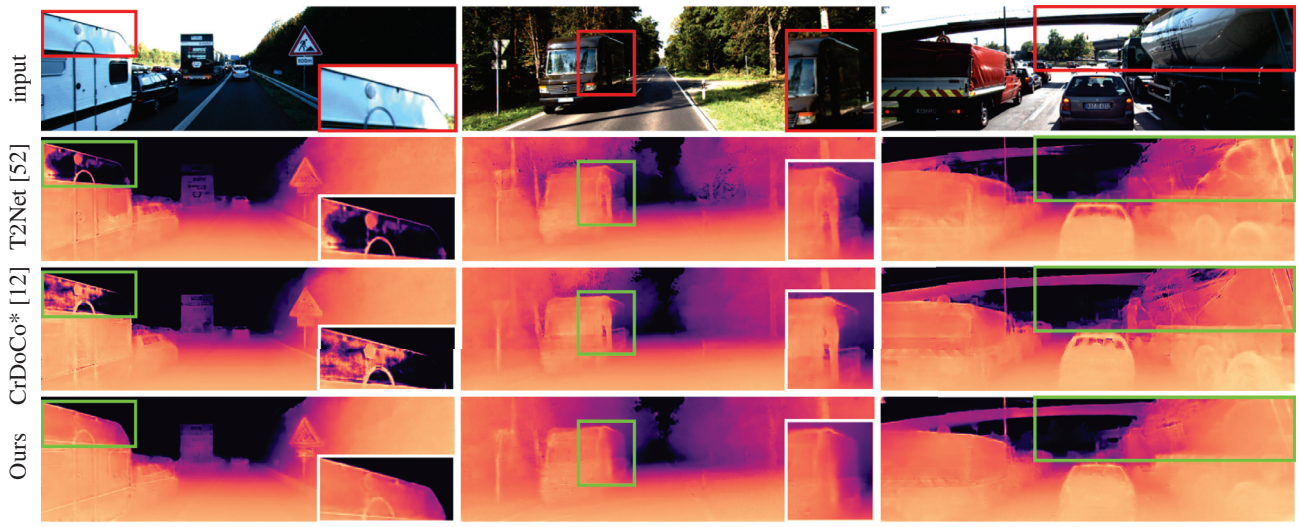

Figure 3: Qualitative results on KITTI [19]. Additional qualitative results are available in the supplementary material.

terparts. Specifically, images are resized to 960×288 in our framework. This is the same aspect ratio, *i.e.* 10:3, used in the previous state-of-the-art UDA method (640×192) [52].

At test time, we upsample the predictions (960×288) to match the ground truth resolution (1241×376) and apply the Garg cropping [18]. The reported results are based on the range of 1-80m for KITTI. Additionally, to show generalization performance on different scenes, we follow previous works for depth estimation [52, 53, 44, 21, 29] to test our model on the Make3D [2] outdoor scene dataset without re-training on the Make3D. For the evaluation on Make3D, we follow the same testing protocol and the evaluation criteria as in [21].

We note that in contrast to [52], we could not conduct an experiment on indoor scenes using SUNCG [40] because SUNCG [40] is no longer publicly available. Therefore, we follow other depth estimation works, such as [20, 21, 50], to conduct extensive experiments on outdoor scenes using KITTI, the most used benchmark dataset for depth estimation, as well as vKITTI and Make3D.

5.3. Implementation details

We trained our framework using PyTorch on 8 NVIDIA A100 GPUs. During optimization, we set most of the the relative weights of the different loss functions based on previous works: λ_{cycle} from [54], and λ_{task}, λ_{smooth}, $\lambda_{identity}$ (or α_r in [52]) from [52]. We also conduct an ablation study on λ_{crdoco} as we modify the task network. More specifically, we set the hyper-parameters as $\lambda_{cycle} = 10$, $\lambda_{identity} = 100$, $\lambda_{task} = 100$, $\lambda_{smooth} = 0.1$, $\lambda_{crdoco} = 1$.

Our models are trained for 20 epochs (style transfer stage), 100 epochs (SSRL stage) and 30 epochs (depth estimation stage). We set the batch size of 16, 128, 16 for each stage and the learning rate of $lr \times BatchSize/16$ (linear scaling [24]), with a base $lr = 0.0004$ for the task network F and Siamese network S, and $lr = 0.0002$ for the image translation network $G_{S \to T}$, $G_{T \to S}$, D_S, and D_T. Also, we applied a linearly decaying rate for the last half epochs in each stage. The training takes around 40 hours for the style transfer stage, 15 hours for the SSRL stage, and 40 hours for the depth estimation stage. For the depth estimation stage, we train our model twice and report the mean scores of all metrics. The implementation code and model weights will be made available.

5.4. Comparisons

We use the standard metrics reported in [14] for evaluation. As comparison methods, we adopt the state-of-the-art UDA methods, T2Net [52] and CrDoCo [12]. We trained their models with their released code with the same resolution size as our model, *i.e.* 960×288, for a fair comparison.

In addition, as mentioned in Section 3.2, we follow the recent standard to utilize ImageNet [13] Initialization and design our encoder-decoder based task network with EfficientNetB5 [42] pre-trained on ImageNet [13] as an encoder. For a fair comparison, therefore, we implement the comparison methods, T2Net [52] and CrDoCo [12], using our task network pre-trained on ImageNet [13].

Furthermore, unlike T2Net [52] that prepares hyper-parameters specifically for outdoor scene depth estimation, CrDoCo [12] did not conduct experiments on outdoor scenes and thus, their original hyper-parameters may not be suitable for outdoor scenes. This observation can be backed up with our experiment as shown in Table 1 and a recent work [51], which indicates that CrDoCo [12] performs worse than T2Net [52] in depth estimation tasks. In our work, we believe that using non-optimal hyper-parameters for comparison methods is less meaningful because we aim

Method	Lower is better				Higher is better		
	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
w/o SSRL	0.171	1.423	5.781	0.239	0.776	0.916	0.968
3rd last layer	0.173	1.376	5.570	0.239	0.768	0.914	0.968
2nd last layer	0.170	1.318	5.558	0.236	**0.779**	**0.924**	0.972
last layer	**0.168**	**1.228**	**5.498**	**0.235**	0.771	0.921	**0.973**

Table 3: Ablation studies with different layers of F_{dec} for the SSRL stage on KITTI.

Method	Lower is better				Higher is better		
	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
global	0.194	1.625	5.890	0.248	0.750	0.909	0.966
pixel-wise	**0.168**	**1.228**	**5.498**	**0.235**	**0.771**	**0.921**	**0.973**

Table 4: Ablation studies with L_{cossim} on KITTI.

to not only achieve the state-of-the-art but also show the current exact performance gap between the UDA methods and state-of-the-art supervised counterparts. Therefore, we also report the result of CrDoCo [12] using the same hyper-parameters as our model to show the full performance of the previous state-of-the-art method for a fairer comparison. We denote this model as CrDoCo* in Table 1 and 2.

Please note that we follow previous works [50, 52] to include current SOTA depth estimation methods other than UDA as reference in Table 1 and 2.

5.5. Results

We present the results of depth estimation on KITTI [19] in Table 1. Our method outperforms previous state-of-the-art UDA methods [52, 12] achieving constant improvement in all metrics (*e.g*. 14.7% on Sq Rel and 3.6% on RMSE) and noticeably brings the performance closer to the fully supervised methods. See Fig. 3 for a qualitative comparison. Our method is able to handle bright reflections and complex structures much better than other UDA methods, and thus, infer depths more reliably. Additional qualitative results are available in the supplementary material.

We provide the results on Make3D [2] in Table 2. Again, our method performs better than all UDA methods on all metrics *e.g*. by 16.9% on Sq Rel. It is also noticeable that our method performs on par with the self-supervised counterparts that utilize extra information such as stereo or ego-motion [21, 29, 22] and achieve the best result on $\log_{10}$ among all of UDA and self-supervised methods.

5.6. Ablation studies

We provide ablation studies on KITTI [19] in Table 3, 4, 5, and 6. Additional ablation studies are available in the supplementary material.

Layer of F_{dec} for SSRL. We study different layers of F_{dec} for the SSRL stage, as summarized in Table 3. We first highlight that our framework even without our proposed SSRL stage ('w/o SSRL') achieves state-of-the-art performance against current SOTA methods in Table 1. In addition, we observe that using the representations for SSRL from the 'last layer' or the '2nd to last layer' before the final output layer yields better performance than 'w/o SSRL'. Note that the '2nd to last layer' model outperforms 'w/o SSRL' on all metrics whereas the 'last layer' model shows great improvement on almost all metrics with a slight drop in the $\delta < 1.25$ metric. For our experiments, we adopt 'last layer' as mentioned in Section 4 because we consider the improvement in Sq Rel (13.7%) and RMSE (4.9%) more significant than the slight decrease in $\delta < 1.25$ (0.6%).

Cosine similarity loss L_{cossim}. We study the effect of the cosine similarity loss L_{cossim} in Table 4. Specifically, we compare a global negative cosine similarity function, in which the correspondence is built between the whole feature maps, with the pixel-wise negative cosine similarity function as described in Section 4. The result indicates that 'pixel-wise' performs better than 'global' with significant margins for our high dimensional features, contributing to our state-of-the-art performance. This supports our assumption as described in Section 4.2 that when the latent features are high dimensional, it is beneficial to account for the correspondence between the features at the level of pixels.

Scaling factor α for the predictor S_{pre}. We explore the effect of different values of the scaling factor α that decides the number of the hidden layers' channel dimensions of S_{pre} as shown in Table 5. The result shows that relatively larger α (*i.e*. the lower number of the channel dimensions) performs better than $\alpha = 1$ (*i.e*. the same number of the channel dimensions as inputs). Thus, we observe the same tendency as indicated in [10] that 'auto-encoder'-like structures can be effective for the predictor to digest information. Based on this result, we adopt $\alpha = 3$ for all experiments.

Method	Lower is better				Higher is better		
	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
$\alpha = 1$	0.185	1.596	5.964	0.249	0.757	0.910	0.968
$\alpha = 2$	0.172	1.389	5.613	0.238	**0.771**	0.915	0.969
$\alpha = 3$	**0.168**	**1.228**	**5.498**	**0.235**	**0.771**	**0.921**	**0.973**
$\alpha = 4$	0.170	1.306	5.528	0.236	0.768	0.918	0.971

Table 5: Ablation studies with the different values of α in S_{pre} on KITTI.

Method	Lower is better				Higher is better		
	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Task only w synthetic data	0.176	1.752	6.209	0.252	0.766	0.907	0.962
Task only w real data	0.100	0.631	4.190	0.165	0.886	0.968	0.988
Ours	**0.168**	**1.228**	**5.498**	**0.235**	0.771	**0.921**	**0.973**
(a) w SSRL and DE combined, w L_{cossim}	0.175	1.477	5.712	0.240	0.769	0.913	0.967
(b) w SSRL and DE combined, w S and L_{cossim}	0.171	1.302	5.520	0.238	**0.773**	0.915	0.969
(c) w local features for S in SSRL	0.189	1.621	5.841	0.250	0.753	0.906	0.965
(d) w local features for S_{pre} in SSRL	0.174	1.463	5.799	0.239	0.769	0.914	0.969

Table 6: Ablation studies with variations of our framework on KITTI. Task: task network, S: Siamese network, SSRL: self-supervised representation learning stage, DE: depth estimation stage.

Variations of our framework. We study different variations of our framework, as summarized in Table 6.

Firstly, we train only the task network F using either the synthetic or real dataset. It is worth noting that T2Net [52] (Table 1) did not show better performance against the task network trained on only the synthetic dataset ('Task only w synthetic data'). We suspect that this is because T2Net [52] is based on an end-to-end training with an image translation network and a task network. That is, at the beginning of the training, their translation network produces images of 'bad quality' (*e.g*. blurry images) that lead to inappropriate gradients for the encoder of the task network pre-traiend on ImangeNet. Meanwhile, our framework and CrDoCo* [12] use a separate style transfer stage to pre-train the image translation network to produce images with 'good' quality in advance. Thus, these methods are able to make full use of the ImageNet initialization.

Secondly, we implement our framework with the SSRL and depth estimation stages combined to see if the separate SSRL stage is necessary. Specifically, we (a) apply only L_{cossim} in the depth estimation stage or (b) utilize both the Siamese network S and L_{cossim} in the depth estimation stage. Although the latter model shows promising results, introducing the separate SSRL stage performs better. It is also worth noting that (a) performs worse than 'w/o SSRL' in Table 3. We believe that this is because L_{cossim} acts in a similar way as the cross-domain consistency loss L_{crdoco} [12], resulting in too strong prediction level consistency. This observation can be backed up by our ablation study on L_{crdoco} in Table 8 in the supplementary material, which indicates that strong prediction level consistency deteriorates the performance.

Thirdly, we explore ways to input feature maps into S instead of using global feature maps as inputs to S as described in Section 4. One way is to divide the feature maps into several local feature blocks, which are forwarded to S. Specifically, (c) we separate M of the size $12 \times 960 \times 288$ into 30 blocks of the size $12 \times 96 \times 96$, and input the blocks into S. In another way, (d) we perform the same operation to the output of S_{pro} and apply them to only S_{pre}. As shown in Table 6, however, using global features yields the best performance; therefore, we adopt it for all experiments.

6. Conclusion

We introduced a novel framework for unsupervised domain adaptation for monocular depth estimation. We propose using a bidirectional image translation and a Siamese network to learn representations that are invariant across real and synthetic domains in a self-supervised manner. Our extensive results demonstrate that our method brings decisive improvements both quantitatively and qualitatively on two popular datasets, KITTI and Make3D, *e.g*. by 14.7% for KITTI and 16.9% for Make3D on square relative error (Sq Rel). Limitation: our method requires an image-to-image translation network to provide two style augmented views of the same scene. In future work, we would like to explore modifications that do not require paired style augmented images. We would also like to investigate how our framework generalizes to other dense prediction tasks such as semantic segmentation.

References

[1] Ibraheem Alhashim and Peter Wonka. High quality monocular depth estimation via transfer learning. *arXiv e-prints*, 2018.

[2] Andrew Y. Ng. Ashutosh Saxena, Min Sun. Make3d: Learning 3d scene structure from a single still image. *IEEE Transactions of Pattern Analysis and Machine Intelligence (PAMI)*, 30(5):824–840, 2009.

[3] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins, 2020.

[4] Sai Bi, Kalyan Sunkavalli, Federico Perazzi, Eli Shechtman, Vladimir G. Kim, and Ravi Ramamoorthi. Deep cg2real: Synthetic-to-real translation via image disentanglement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[5] Minghao Chen, Hongyang Xue, and Deng Cai. Domain adaptation for semantic segmentation with maximum squares loss. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2090–2099, 2019.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[7] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 22243–22255. Curran Associates, Inc., 2020.

[8] Wuyang Chen, Zhiding Yu, SD Mello, Sifei Liu, Jose M Alvarez, Zhangyang Wang, and Anima Anandkumar. Contrastive syn-to-real generalization. 2021.

[9] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[10] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

[11] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised visual transformers. *arXiv preprint arXiv:2104.02057*, 2021.

[12] Yun-Chun Chen, Yen-Yu Lin, Ming-Hsuan Yang, and Jia-Bin Huang. Crdoco: Pixel-level domain transfer with cross-domain consistency. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[14] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[15] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2002–2011, 2018.

[16] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4340–4349, 2016.

[17] Yukang Gan, Xiangyu Xu, Wenxiu Sun, and Liang Lin. Monocular depth estimation with affinity, vertical pooling, and label enhancement. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 224–239, 2018.

[18] Ravi Garg, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European conference on computer vision*, pages 740–756. Springer, 2016.

[19] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[20] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017.

[21] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3828–3838, 2019.

[22] Juan Luis GonzalezBello and Munchurl Kim. Forget about the lidar: Self-supervised depth estimators with med probability volumes. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12626–12637. Curran Associates, Inc., 2020.

[23] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

[24] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[25] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21271–21284. Curran Associates, Inc., 2020.

[26] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

[27] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[28] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[29] Adrian Johnston and Gustavo Carneiro. Self-supervised monocular trained depth estimation using self-attention and discrete disparity volume. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4756–4765, 2020.

[30] Jogendra Nath Kundu, Phani Krishna Uppala, Anuj Pahuja, and R Venkatesh Babu. Adadepth: Unsupervised content congruent adaptation for depth estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2656–2665, 2018.

[31] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*, 2019.

[32] Guangrui Li, Guoliang Kang, Wu Liu, Yunchao Wei, and Yi Yang. Content-consistent matching for domain adaptive semantic segmentation. In *European Conference on Computer Vision*, pages 440–456. Springer, 2020.

[33] Qing Lian, Fengmao Lv, Lixin Duan, and Boqing Gong. Constructing self-motivated pyramid curriculums for cross-domain semantic segmentation: A non-adversarial approach. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6758–6767, 2019.

[34] Weizhe Liu, David Ferstl, Samuel Schulter, Lukas Zebedin, Pascal Fua, and Christian Leistner. Domain adaptation for semantic segmentation via patch-wise contrastive learning, 2021.

[35] Adrian Lopez-Rodriguez and Krystian Mikolajczyk. Desc: Domain adaptation for depth estimation via semantic consistency. In *British Machine Vision Conference (BMVC)*, 2020.

[36] Fengmao Lv, Tao Liang, Xiang Chen, and Guosheng Lin. Cross-domain semantic segmentation via domain-invariant interactive relation transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4334–4343, 2020.

[37] Andrea Pilzer, Dan Xu, Mihai Marian Puscas, Elisa Ricci, and Nicu Sebe. Unsupervised adversarial depth estimation using cycled generative networks. *CoRR*, abs/1807.10915, 2018.

[38] Koutilya PNVR, Hao Zhou, and David Jacobs. Sharingan: Combining synthetic and real data for unsupervised geometry estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13974–13983, 2020.

[39] Swami Sankaranarayanan, Yogesh Balaji, Arpit Jain, Ser Nam Lim, and Rama Chellappa. Learning from synthetic data: Addressing domain shift for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3752–3761, 2018.

[40] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. *Proceedings of 30th IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[41] M Naseer Subhani and Mohsen Ali. Learning from scale-invariant examples for domain adaptation in semantic segmentation. *arXiv preprint arXiv:2007.14449*, 2020.

[42] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

[43] Alessio Tonioni, Matteo Poggi, Stefano Mattoccia, and Luigi di Stefano. Unsupervised domain adaptation for depth prediction from images. *CoRR*, abs/1909.03943, 2019.

[44] Chaoyang Wang, José Miguel Buenaposada, Rui Zhu, and Simon Lucey. Learning depth from monocular videos using direct methods. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2022–2030, 2018.

[45] Zuxuan Wu, Xintong Han, Yen-Liang Lin, Mustafa Gokhan Uzunbas, Tom Goldstein, Ser Nam Lim, and Larry S Davis. Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 518–534, 2018.

[46] Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. 2021.

[47] Yanchao Yang and Stefano Soatto. Fda: Fourier domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4085–4095, 2020.

[48] Wei Yin, Yifan Liu, Chunhua Shen, and Youliang Yan. Enforcing geometric constraints of virtual normal for depth prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5684–5693, 2019.

[49] Zhenyu Zhang, Stephane Lathuiliere, Elisa Ricci, Nicu Sebe, Yan Yan, and Jian Yang. Online depth learning against forgetting in monocular videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[50] Shanshan Zhao, Huan Fu, Mingming Gong, and Dacheng Tao. Geometry-aware symmetric domain adaptation for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9788–9798, 2019.

[51] Yunhan Zhao, Shu Kong, Daeyun Shin, and Charless Fowlkes. Domain decluttering: Simplifying images to mitigate synthetic-real domain shift and improve depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3330–3340, 2020.

[52] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. T2net: Synthetic-to-realistic translation for solving single-image depth estimation tasks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 767–783, 2018.

[53] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1851–1858, 2017.

[54] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

[55] Xinge Zhu, Hui Zhou, Ceyuan Yang, Jianping Shi, and Dahua Lin. Penalizing top performers: Conservative loss for semantic segmentation adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 568–583, 2018.

[56] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Domain adaptation for semantic segmentation via class-balanced self-training. *arXiv preprint arXiv:1810.07911*, 2018.