SemI2I: Semantically Consistent Image-to-Image Translation for Domain Adaptation of Remote Sensing Data

Onur Tasar, S L Happy, Yuliya Tarabalka, Pierre Alliez

SEMI2I: SEMANTICALLY CONSISTENT IMAGE-TO-IMAGE TRANSLATION FOR DOMAIN ADAPTATION OF REMOTE SENSING DATA

Onur Tasar[1]*, S L Happy*[2]*, Yuliya Tarabalka*[3]*, Pierre Alliez*[1]

[1]Université Côte d'Azur, Inria, TITANE team; [2]Inria, STARS team; [3]LuxCarta Technology
Email: onur.tasar@inria.fr

ABSTRACT

Although convolutional neural networks have been proven to be an effective tool to generate high quality maps from remote sensing images, their performance significantly deteriorates when there exists a large domain shift between training and test data. To address this issue, we propose a new data augmentation approach that transfers the style of test data to training data using generative adversarial networks. Our semantic segmentation framework consists in first training a U-net from the real training data and then fine-tuning it on the test stylized fake training data generated by the proposed approach. Our experimental results prove that our framework outperforms the existing domain adaptation methods.

Index Terms— Domain adaptation, data augmentation, semantic segmentation, dense labeling, image-to-image translation, generative adversarial networks, GANs.

1. INTRODUCTION

Over the years, U-net [1] and its variants [2] have become one of the most commonly used network architecture for semantic segmentation problem because of their continuous success in various benchmarks. However, the difference between the data distributions of training and test images caused by atmospheric effects, position of the sun, intra-class variability, etc. makes the U-net likely to fail to generate good maps. To overcome this issue, the traditional data augmentation methods such as gamma correction, random contrast change [3, 4], histogram matching [5], color constancy algorithms [6] are widely used. However, adopting such augmentation techniques is usually insufficient to generalize the model to unseen data, especially when the domain shift between training and test data is highly large.

Better data augmentation strategy would be to generate semantically consistent and test stylized fake training data. To accomplish this task, especially in the field of computer vision, various image-to-image translation (I2I) approaches have already been proposed [7, 8, 9, 10]. The biggest challenge for the style transfer problem is to generate test stylized fake training image that semantically matches the real training image, so that one can use the fake training image and the ground-truth for the real training image to train or to fine-tune a model. The recent I2I approaches [7, 8, 9, 10] fail to keep the real and the fake training images semantically the same. For instance, some of them replace trees by buildings, buildings by roads, etc., whereas some of them create totally artificial objects in the fake training image, which are not available in the real training data. Thus, the fake training images generated by these approaches and the ground-truth for the real training data do not correspond. Hence, training a model on such image and ground-truth pair causes the model to yield a poor performance. Another approach is to adapt the model to test data [11], which is a more difficult task, rather than modifying the data.

To overcome the limitations described above, Tasar *et al.* have recently proposed ColorMapGAN [12], which learns to change the color distribution of training data without doing any structural change. Although the method works very well, it has some limitations. Firstly, it does not support style transfer when the images in both domains have different number of channels. Secondly, it learns to map the colors of the source domain linearly, which may not be strong enough in some cases. Finally, since it transforms each color separately, its output is usually slightly noisy. In this work, we propose semantically consistent image-to-image translation (SemI2I) method, which overcomes all the limitations of ColorMapGAN. By following the same segmentation strategy explained in [12], we use the fake training images and the original ground-truth to fine-tune the U-net trained on real data.

2. METHODOLOGY

Let us assume that both domains are denoted by A and B, respectively. The essential goal of SemI2I is to generate fake A with the style of B, and fake B with the style of A. The design of SemI2I is illustrated in Fig. 1. The main components and the stages of SemI2I are as follows:

Style transfer. The crucial components for the style transfer are adaptive instance normalization (adaIN [13]) and adversarial losses. To generate B stylized fake A and A stylized fake B, we first encode both A and B. In the encoded embeddings, we compute mean μ and standard deviation σ for each feature channel. The embedding itself has the content information, whereas μ and σ carry the style information. We combine the content of A and the style of B, and the content of B and the style of A by using adaIN. After this operation,

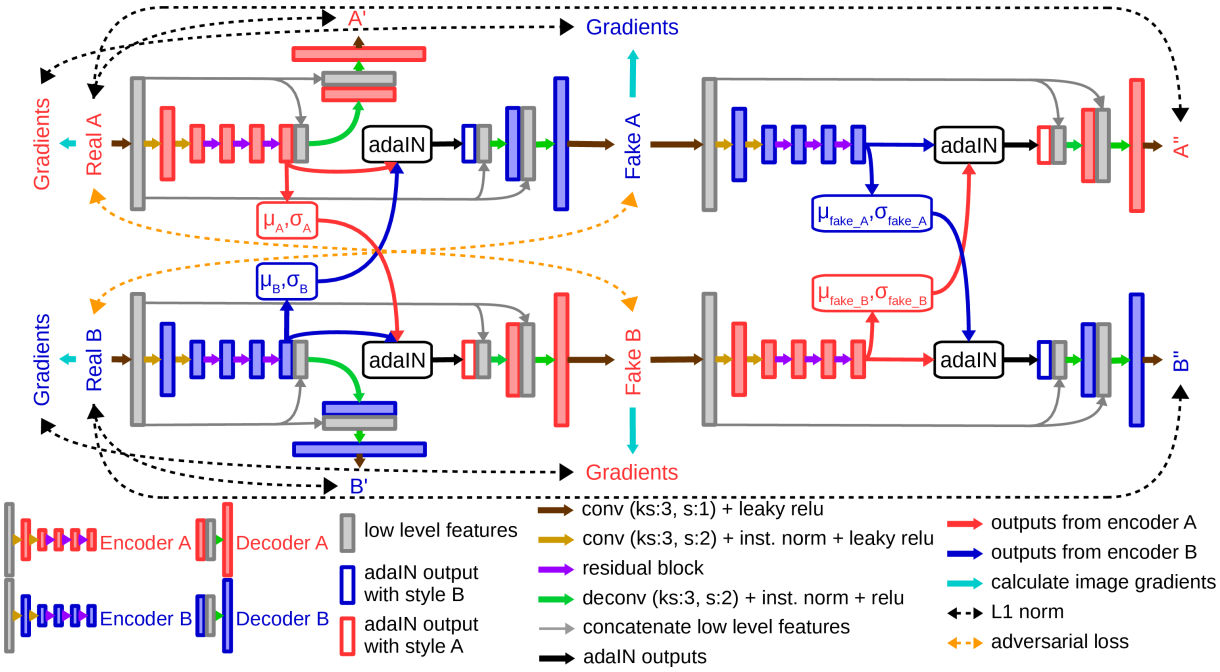


Fig. 1. SemI2I. ks and s stand for kernel size and strides. μ and σ are the mean and standard deviation values for each feature channel of the embeddings. Since SemI2I performs I2I, one can consider A as training and B as test data, or vise versa.

since we switch the styles of A and B, we obtain fake A by the decoder B and fake B by the decoder A. Moreover, to make B and fake A, and A and fake B visually as similar as possible, we minimize the adversarial losses between them. The combination of an encoder and a decoder forms a generator. For the discriminator, we use the same architecture as in CycleGAN [7]. As can be verified from Fig. 1, there are 6 generators and 2 discriminators in SemI2I.

Semantic consistency. By following the same strategy described above, we switch the styles of fake A and fake B to obtain A$''$ and B$''$ (cross reconstruction). In an ideal transformation A and A$''$, and B and B$''$ have to be exactly the same. In addition, when we encode A with encoder A and decode the embedding with decoder A, we must reconstruct itself. We call this operation self reconstruction. The self reconstruction also applies to B. Besides, we compute the image gradients from A and fake A by sobel filter. We force the difference between the gradients calculated from A and fake A to be as small as possible to help them to be semantically consistent. Again, the same rule applies to B. Finally, we have visually verified that the filters in the first convolution layer of each encoder learns the low level features such as edges. We re-size and concatenate these low level features from each encoder with the input to each deconvolution layer in the corresponding decoder (see gray arrows in Fig. 1). Such concatenation allows the decoder to have a footprint of the objects in the real data; therefore, it guides the decoder to place the right objects in the correct locations.

Training. We now formulate the losses to train SemI2I. We compute the cross reconstruction loss $\mathcal{L}_{cross}$, which is the sum of L1 norm between A and A$''$ and L1 norm between B and B$''$. Similarly, the self reconstruction loss $\mathcal{L}_{self}$ is computed by summing L1 norms between A and A$'$, and B and B$'$. By adding L1 norms between the image gradients calculated from A and fake A, and the gradients from B and fake B, we compute the gradient loss $\mathcal{L}_{grad}$. Finally, we use the loss functions in LSGAN [14] to compute the adversarial losses. The adversarial loss for the generators $\mathcal{L}_{adv\_G}$ is the sum of the LSGAN generator losses between A and fake B, and B and fake A. Similarly, the adversarial loss for the discriminators $\mathcal{L}_{adv\_D}$ is computed by adding the LSGAN discriminator losses between A and fake B, and B and fake A. The overall generator loss $\mathcal{L}_G$ that is minimized in the training stage is computed as:

$$\mathcal{L}_G = \lambda_1 \mathcal{L}_{cross} + \lambda_2 \mathcal{L}_{self} + \lambda_3 \mathcal{L}_{grad} + \lambda_4 \mathcal{L}_{adv\_G}, \quad (1)$$

where λ_1, λ_2, λ_3, and λ_4 are used to adjust the relative importance of each loss. The discriminator loss is defined as :

$$\mathcal{L}_D = \lambda_4 \mathcal{L}_{adv\_D}. \quad (2)$$

To train SemI2I, we simultaneously minimize $\mathcal{L}_G$ and $\mathcal{L}_D$.

Test. In the test stage, to generate B stylized fake A, we need the encoder A and the decoder B (see the top left generator in Fig. 1 that generates fake A). However, as can be seen in Fig. 1, before feeding the embedding encoded by the encoder A from real A to the decoder B, it needs to be normalized by adaIN using μ_B and σ_B calculated from the embedding

of B. When generating fake A, one may think of randomly sampling a patch from B, encoding it by the encoder B, computing its μ and σ values, and using them to normalize the embedding of A. However, this is not a good idea, because depending on which patch is sampled from B, we may end up generating fake A having a different style in each run. We have exactly the same problem when generating fake B. We want SemI2I to generate exactly the same fake data every time when we test it. To do this, we estimate the global μ and σ values for the embeddings of both domains via Alg. 1 during the training. In Alg. 1, d_rate is a parameter ranging between 0 and 1. Note that this parameter needs to be set to a value that is close to 1 (e.g., 0.95), so that the current patches would not change the global mean and standard deviation values too much. When generating fake A, we use μ_{gB} and σ_{gB}. Similarly, we use μ_{gA} and σ_{gA} while generating fake B.

Algorithm 1: Estimation of the global μ and σ values for the embeddings of both domains. $*$ corresponds to a domain (either A or B).

output: global mean μ_{g*} and std. σ_{g*} for the embed.
$\mu_{g*} \leftarrow 0$, $\sigma_{g*} \leftarrow 0$
d_rate $\leftarrow 0.95$; `// decay rate`
foreach *training iteration* **do**
  sample a patch, compute μ_* and σ_* of its emb.
  $\mu_{g*} \leftarrow \text{d\_rate} \times \mu_{g*} + (1 - \text{d\_rate}) \times \mu_*$
  $\sigma_{g*} \leftarrow \text{d\_rate} \times \sigma_{g*} + (1 - \text{d\_rate}) \times \sigma_*$
end

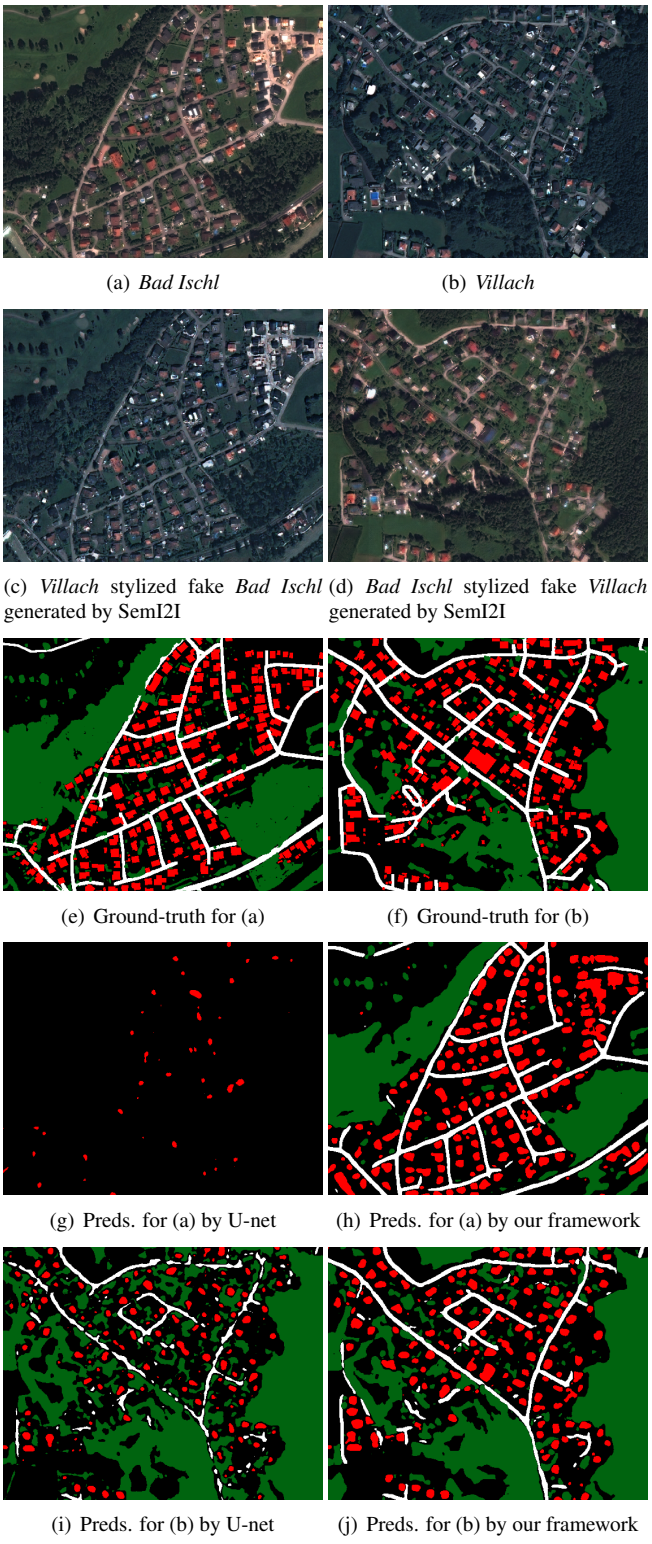
3. EXPERIMENTS AND CONCLUSION

We use Pléiades images collected from *Bad Ischl* and *Villach* in *Austria*, covering 27.71 km^2 and 43.59 km^2, respectively. The images contain RGB color channels, and their resolution is 1 m. The annotations for *building*, *road*, and *tree* classes have been manually prepared. We perform city-to-city domain adaptation. In the first experiment, we use *Bad Ischl* as training and *Villach* as test data. In the second experiment, we switch the training and the test data.

We split the cities into 256×256 patches with an overlap of 32 pixels. In each training iteration, we randomly sample only 1 patch from each domain. We train SemI2I for 25 epochs, where the number of iterations in each epoch is the minimum of the number of patches from each domain. We optimize SemI2I with Adam optimizer. We set the learning rate to 0.001 in the first 15 epochs, and compute it in the rest of the epochs as:

$$\text{LR} = 0.001 \times \frac{\text{num\_epochs} - \text{epoch\_no}}{\text{num\_epochs} - \text{decay\_epoch}}, \qquad (3)$$

where LR, num_epochs, epoch_no are the current learning rate, the total number of epochs, and current epoch no. decay_epoch stands for the epoch, which is set to 15 in our experiments, where the learning rate is started to be reduced.



(a) *Bad Ischl* (b) *Villach*

(c) *Villach* stylized fake *Bad Ischl* generated by SemI2I (d) *Bad Ischl* stylized fake *Villach* generated by SemI2I

(e) Ground-truth for (a) (f) Ground-truth for (b)

(g) Preds. for (a) by U-net (h) Preds. for (a) by our framework

(i) Preds. for (b) by U-net (j) Preds. for (b) by our framework

Fig. 2. Training and test data used in both experiments, their ground-truth, test stylized fake training images generated by SemI2I, and the predictions generated by U-net and our framework. *Red, white,* and *green* colors represent *building, road*, and *tree* classes, respectively.

Table 1. IoU scores.

Method		Training: Bad Ischl, Test: Villach				Training: Villach, Test: Bad Ischl			
		building	**road**	**tree**	**Overall**	**building**	**road**	**tree**	**Overall**
U-net		23.61	0.91	40.53	21.68	5.84	0.24	0.50	2.19
AdaptSegNet Single[11]		6.01	4.37	10.43	6.94	3.06	2.71	10.23	5.33
AdaptSegNet Multi[11]		24.59	9.02	56.08	29.86	14.26	4.46	24.66	14.46
Our proposed framework with	CycleGAN[7]	43.03	28.96	**68.86**	46.95	43.62	38.69	71.68	51.33
	UNIT[8]	30.86	15.84	63.00	36.57	19.29	36.83	35.57	30.56
	MUNIT[9]	0.02	1.38	47.23	16.21	6.20	0.13	0.05	2.13
	DRIT[10]	0.01	3.96	8.72	4.23	0.00	10.19	0.01	3.40
	Gray world[6]	25.19	26.43	56.15	35.92	29.55	24.80	46.41	33.58
	Histogram matching[5]	24.95	29.34	61.59	38.63	6.45	0.92	1.28	2.88
	ColorMapGAN[12]	**48.47**	37.82	58.92	48.40	49.16	41.75	59.84	50.25
	SemI2I (ours)	47.79	**38.22**	68.76	**51.59**	**53.38**	**47.59**	**79.17**	**60.05**

We set $\lambda_1, \lambda_2, \lambda_3$ and λ_4 in Eqs. 1 and 2 to 10, 10, 10, and 1, respectively. We have found these values empirically. To generate maps, we first train a U-net for 8,000 iterations on the real training data. We then fine-tune it using the fake image patches generated by SemI2I and the original ground-truth for 2,500 iterations. In each training iteration of both the initial training and the fine-tuning steps, we sample a mini-batch of 32 patches. We optimize U-net in both steps via Adam optimizer with the learning rate of 0.0001.

Among the I2I approaches, we compare SemI2I with CycleGAN [7], UNIT [8], MUNIT [9], DRIT [10], histogram matching [5], and gray world algorithm [6]. To make a fair comparison, we replace the fake images used in the fine-tuning step by the fake images generated by these methods. We also provide the results for the traditional U-net [1] without doing domain adaptation and AdaptSegNet [11], which aims at adapting the model instead of modifying the data.

Fig. 2 depicts the close-ups from the city pair used in the experiments, test stylized fake training images generated by SemI2I, the ground-truth, and the predictions by U-net and our framework. As can be seen, our method performs the style transfer perfectly, and real and the fake images are semantically consistent. The figure shows that U-net performs extremely poorly, when there is a large shift between the data distributions of training and test images. The improvement of our framework against U-net can clearly be observed. The quantitative results for our segmentation framework and for the other methods are reported in Table 1, which demonstrates the superior performance of our methodology.

In this work, we presented our novel SemI2I approach, which is a new data augmentation method. We compared our segmentation framework with no less than 10 different methods and showed that it exhibits a much better performance than the existing domain adaptation approaches. In the future, we plan to tackle multiple cities-to-multiple cities adaptation problem instead of performing city-to-city adaptation.

4. REFERENCES

[1] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015.

[2] A. Khalel, O. Tasar, G. Charpiat, and Y. Tarabalka, "Multi-task deep learning for satellite image pansharpening and segmentation," in *IGARSS*, 2019.

[3] O. Tasar, Y. Tarabalka, and P. Alliez, "Incremental learning for semantic segmentation of large-scale remote sensing data," *IEEE JSTARS*, 2019.

[4] O. Tasar, Y. Tarabalka, and P. Alliez, "Continual learning for dense labeling of satellite images," in *IGARSS*, 2019.

[5] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition)*, 2006.

[6] G. Buchsbaum, "A spatial processor model for object colour perception," *Journal of the Franklin institute*, 1980.

[7] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *CVPR*, 2017.

[8] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *NIPS*, 2017.

[9] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *ECCV*, 2018.

[10] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. K. Singh, and M.-H. Yang, "Diverse image-to-image translation via disentangled representations," in *ECCV*, 2018.

[11] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker, "Learning to adapt structured output space for semantic segmentation," in *CVPR*, 2018.

[12] O. Tasar, S L Happy, Y. Tarabalka, and P. Alliez, "ColorMapGAN: Unsupervised domain adaptation for semantic segmentation using color mapping generative adversarial networks," *arXiV*, 2019.

[13] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *ICCV*, 2017.

[14] X. Mao, Q. Li, H. Xie, Raymond Y.K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *ICCV*, 2017.