DAugNet: Unsupervised, Multi-source, Multi-target, and Life-long Domain Adaptation for Semantic Segmentation of Satellite Images

Onur Tasar, Alain Giros, Yuliya Tarabalka, Pierre Alliez, Sébastien Clerc

To cite this version:

Onur Tasar, Alain Giros, Yuliya Tarabalka, Pierre Alliez, Sébastien Clerc. DAugNet: Unsupervised, Multi-source, Multi-target, and Life-long Domain Adaptation for Semantic Segmentation of Satellite Images. IEEE Transactions on Geoscience and Remote Sensing, 2020, 10.1109/TGRS.2020.3006161. hal-02893533

HAL Id: hal-02893533
https://hal.science/hal-02893533v1
Submitted on 8 Jul 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
DAugNet: Unsupervised, Multi-source, Multi-target, and Life-long Domain Adaptation for Semantic Segmentation of Satellite Images

Onur Tasar, Graduate Student member, IEEE, Alain Giros, Member, IEEE, Yuliya Tarabalka, Senior member, IEEE, Pierre Alliez, and Sébastien Clerc

Abstract—The domain adaptation of satellite images has recently gained an increasing attention to overcome the limited generalization abilities of machine learning models when segmenting large-scale satellite images. Most of the existing approaches seek for adapting the model from one domain to another. However, such single-source and single-target setting prevents the methods from being scalable solutions, since nowadays multiple source and target domains having different data distributions are usually available. Besides, the continuous proliferation of satellite images necessitates the classifiers to adapt to continuously increasing data. We propose a novel approach, coined DAugNet, for unsupervised, multi-source, multi-target, and life-long domain adaptation of satellite images. It consists of a classifier and a data augmentor. The data augmentor, which is a shallow network, is able to perform style transfer between multiple satellite images in an unsupervised manner, even when new data are added over the time. In each training iteration, it provides the classifier with diversified data, which makes the classifier robust to large data distribution difference between the domains. Our extensive experiments prove that DAugNet significantly better generalizes to new geographic locations than the existing approaches.

Index Terms—Domain adaptation, semantic segmentation, dense labeling, convolutional neural networks (CNNs), generative adversarial networks (GANs), multi-source adaption, multi-target adaption, life-long adaption.

I. INTRODUCTION

With the help of significant technological developments over the years, it has been possible to access to massive amounts of remote sensing data. For example, the constellations of Pléiades satellites are able to capture large amounts of images with high spatial resolution over the globe. The availability of such enormous data has opened the door to various applications and raised many challenges. Among these challenges, semantic segmentation or dense labeling of satellite images has become one of the most interesting and long-standing problems, since it is a crucial process for a wide range of applications in domains such as urban monitoring, urban management, agriculture, automatic mapping, and navigation.

In the last decade, convolutional neural networks (CNNs) have been one of the most promising tools for semantic segmentation. Particularly for dense labeling of remote sensing data, especially U-net [1] is a commonly used network architecture because of its great success in many benchmarks such as INRIA [2] and SpaceNet [3]. Although U-net and some other CNNs generate excellent maps when training data well represent test data, their performance significantly drops when there exists a large distribution shift between the training and the test data. In the literature, this issue is referred to as domain adaptation [4]. In the field of remote sensing, the domain shift usually stems from various atmospheric conditions and some differences in acquisition that change spectral characteristics of objects, intra-class variability, differences in the spectral characteristics of the sensor, or images with different types of spectral bands (e.g., some image might have red, green, and blue bands, whereas the others might consist of near infrared, red, and green channels.).

In remote sensing, we can refer to each satellite image as a domain. In a typical domain adaptation problem, the training and the test images are usually designated as source and target domains. The most naive approach to hinder the performance deficiency under a large distribution difference between the domains is to manually annotate some portion of the target domain to fine-tune [5] the already trained model on the source domain. However, such strategy requires human intervention in the process, and labeling even some portion of a satellite image is time-consuming. To increase the generalization capabilities of CNNs, another common approach is to perform online data augmentation with random color change [6]. For instance, gamma correction and random contrast change are widespread used in remote sensing [7]. However, these augmentation approaches are limited when it comes to adapting the model from one domain to another, when there exists a significant difference between data distributions. For example, with this simple augmentation approaches, it is not possible to adapt a model from one domain containing red, green, blue bands to another one composing of near-infrared, red, green channels. To overcome this limitation, one can use generative adversarial networks (GANs) [8] to generate a fake source domain with a similar data distribution to that of target domain. The fake data then can be used to train a classifier.

Unsupervised domain adaptation aims at adapting the classifier to unlabeled target domains. The common single-source and single-target adaptation setting limits the proposed ap-
proaches to small-scale segmentation problems. The recent
generation of satellites with a short revisit time generate in
routine massive amounts of remote sensing data. These data
usually contain multiple domains with largely different data
distributions and are relevant for many real-world applica-
tions. Therefore, to perform large-scale classification, an ideal
unsupervised domain adaptation approach should be capable
of learning from annotated multiple source domains and well
segmenting unlabeled multiple target domains even when the
data distributions of all the domains are different. In addition,
nowadays, the constellations of satellites collect images from
the entire earth everyday. As a consequence, in many cases,
one receives new annotated source domains and unlabeled
target domains over the time. Hence, it is also crucial for
the ideal approach to perform life-long adaptation over the
continuously increasing data.

By performing unsupervised style transfer between two
domains with GANs, it is possible to generate a fake source
domain whose data distribution is as similar as possible to the
distribution of a target domain [9], [10]. A possible solution for
unsupervised, multi-source, multi target, and life-long domain
adaptation problem would be learning to perform style transfer
between every domain pairs. If this task can be accomplished,
we can generate multiple fake domains from each domain,
where each fake domain is representative for a different
domain. We then can use the fake domains generated from
the source domains to train a classifier. Training a classifier
on such diversified data would allow the classifier to learn
from the data that are representative for both the source and
the target domains, instead of capturing the idiosyncrasies of
each source domain.

However, the problem of style transfer between progres-
sively growing multiple domains raises some challenges.
Firstly, all of the fake domains generated from each source
domain must be semantically consistent with the original
domain. For example, if the method replaces some objects
with others or adds artificial structures to the fake domains
in the generation process, the fake domains and the ground-
truth for the real domain would not match. Thus, we cannot
train a classifier. Secondly, when we perform style transfer
between domain pairs, the data distribution of the fake domain
generated from one domain must be similar to the distribution
of the other domain. Otherwise, the fake domain would not
be representative for the other domain. Thirdly, the number
of networks should not grow with the number of domains
to scale the method to many domains. For example, if the
method requires using a style encoder for each domain as in
MUNIT [11], DRIT [12], StarGAN [13], StarGAN v2 [14],
and StandardGAN [15], it would be limited to a certain
number of domains. In addition, it would be necessary to add
new style encoders every time when we receive new data.
Finally, once the training is completed, ability to perform style
transfer between the domains on the fly is desired. Otherwise,
it would be required to store all the fake domains on a disk.

In this article, we introduce a new approach dealing with the
aforementioned challenges. Our contributions are as follows:
1) We propose an approach that can perform style transfer
between multiple domains with only one encoder, one
decoder, and one discriminator unlike MUNIT [11],
DRIT [12], StarGAN [13], StarGAN v2 [14], and Stan-
dardGAN [15] that rely upon a different style encoder
for each domain. In addition, since the number of net-
woks is constant irrespective of the number of domains,
our approach can do style transfer across multiple do-
mains even when new domains are progressively added.
2) We introduce novel DAugNet consisting of a data aug-
mentor and a classifier. The data augmentor, which is
a shallow network, can stylize each domain as another
one. Due to its simple architecture, it allows online data
diversification rather than storing all the fake data on
a disk and loading them when training a classifier. In
each training iteration of DAugNet, the data augmentor
diversifies a batch of training patches by stylizing each
patch as a randomly selected domain before passing the
batch to the classifier.
3) We validate DAugNet on semantic segmentation of
Pléiades images captured over multiple cities from three
European countries. Our data comprise images with
either red, green, blue bands or near-infrared, red, green
bands. In three extensive experiments, we investigate
how DAugNet performs on single-source and single-
target, multi-source and multi-target, and life-long adap-
tation problems. We also conduct five ablation studies
to further analyze the properties of our solution.

II. RELATED WORK

a) Adapting the Inputs: These methods usually aim at
modifying the source domain to make its data distribution
similar to the distribution of the target domain. Then, a clas-
sifier is trained on the modified source domain and evaluated
on the original target domain. Histogram matching [16], [17]
and graph matching [18] are commonly used methods for
the above-mentioned modification. Another way to adapt the
source domain to the target domain is to use style transfer
or image-to-image translation (I2I) methods to generate a
fake source domain stylized as the target domain. For in-
stance, CyCADA [19] generates a fake source domain with
CycleGAN [20] and uses it to train a classifier. Similarly,
Benjdira et al. [21] use CycleGAN for I2I between aerial
images. Tasar et al. have recently proposed ColorMapGAN [9]
and SemI2I [10] to perform domain adaption between satel-
lite image pairs. Other state-of-the-art I2I approaches like
UNIT [22], MUNIT [11], DRIT [12] are also relevant for
domain adaptation.

b) Adapting the classifiers: The goal of these methods
is to adapt the classifier to an unlabeled target domain rather
than modifying the inputs. To achieve this task, a common
approach is simultaneously to learn from the source domain
supervisedly and to align the features extracted from both do-
mains with adversarial learning [23]–[25]. The same strategy
has also been applied to satellite images [26]. To adapt a
classifier to an unlabeled domain, the approaches based on self
learning [27], [28], expectation-maximization (EM) [29], [30],
multiple cascade classifiers [31], variants of support vector
machine (SVM) [32], centroid and covariance alignment [33],
and tensor alignment [34] have been proposed as well.
c) Data Standardization: The objective of the data standardization methods is to standardize each domain in the pre-processing step so that all the domains have similar data distributions. After the standardization, one can work only on the standardized data instead of the real data. One of the most widespread used standardization method is referred to as linear scaling to unit variance (a.k.a. z-score normalization) [35], which is computed by subtracting the mean value from the features and dividing by the standard deviation. Another common approach is histogram equalization [16]. The color constancy algorithms [36] such as gray-world [37] and gamut [38] aim at extracting the intrinsic properties on the objects that are affected by the illuminant. These algorithms can also be used for data standardization. However, in their problem definitions, these methods assume that there exist only one source domain. Hence, it may not be possible to use them for semantic segmentation.

In our case, on the other hand, there might be more than one source domain. However, such solution is sub-optimal as it is needed to apply an I2I approach multiple times. Specifically for multi-target adaptation problem, an information theoretic approach many times makes this strategy inconvenient. Zhao et al. have recently proposed StandardGAN [15] with the purpose of standardizing multiple domains with GANs. However, the approach requires using a style encoder for each domain, which limits the method to a certain number of domains. In some cases, the domain shift can be corrected by radiometric correction [39], [40] as well. Note that the data standardization methods can be used for multi-source and multi-target adaptation problems.

d) Multi-source Adaptation: One can consider doing I2I between each source and target domain pairs to make the data distribution of the source domains similar to the distribution of the target domain. However, the necessity of applying an I2I approach many times makes this strategy inconvenient. Zhao et al. [41] extends CyCADA [19] to multi-source single-target domain adaptation problem. However, due to its extensive computational requirements, it is not possible to train the method end-to-end on the current GPUs. The authors split the training pipeline into several stages, where the stages are executed consecutively and outputs of each stage are passed to the next stage as inputs. Redko et al. introduce an approach based on optimal transport [42]. Elshamli et al. propose a new method for local climate zone classification [43]. However, the method may not be appropriate for less coarse classes such as building and road. Several approaches have been proposed for multi-source domain adaptation in the context of classification [44]–[46]. However, these methods are not appropriate for semantic segmentation task, as it requires generating a precisely structured pixel-wise classification.

e) Multi-target Adaptation: As in multi-source adaptation problem, it is possible to solve multi-target adaptation problem by executing an I2I method to generate fake source domains with similar distributions to those of the target domains. These fake domains can then be used to train a classifier. However, such solution is sub-optimal as it is needed to apply an I2I approach multiple times. Specifically for multi-target adaptation problem, an information theoretic approach [47] and a model parameter adaptation framework [48] have been proposed. However, in their problem definitions, these methods assume that there exist only one source domain. In our case, on the other hand, there might be more than one source domain. In addition, these methods have been proposed for image categorization problem. Hence, it may not be possible to use them for semantic segmentation.

f) Life-long Adaptation: The problem of life-long, continual, incremental, or gradual adaptation has been tackled via GANs [49], meta-learning [50], or self training [51]. The main limitation of these works is their assumption that progressively added domains are a series of intermediate domains. Although these method are suitable for problems such as video processing and autonomous driving, they may not be applicable to remote sensing data, as the incoming domains might be completely different from each other. In the remote sensing literature, there are methods based on semi-supervised active learning [52]–[55], where an annotator labels some portion of the received domain to be included in the training data. However, we seek for an unsupervised method instead, as automation is the key for practical real-world applications.

III. Method

Our training pipeline consists of two stages. In the first stage, we learn how to stylize each domain like any other using only one encoder, one decoder, and one discriminator. Note that this stage is completely unsupervised. Therefore, we use both annotated source and unlabeled target domains during training. A subset of the networks used in this stage forms a data augmentor. In the second stage, we train DAugNet that is composed of the data augmentor and a classifier. In each training iteration of DAugNet, the data augmentor is frozen and diversifies a batch of patches sampled from source domains by stylizing each patch in the batch as a randomly selected domain. The diversified batch is then given to the classifier. The randomly selected domain can be chosen among both source and target domains. Hence, when training DAugNet, the classifier learns from the data that are representative for all of the source and the target domains. As a result, it is
Fig. 3. Style transfer between two domains. Before the training starts unique and style random codes $S_A$ and $S_B$ are randomly drawn from the uniform distribution. To generate fake A and fake B, we combine the embeddings of A and B with $S_B$ and $S_A$, respectively. L1 norms enforce semantic consistency between A and fake A, and B and fake B. The adversarial losses force fake A and fake B to look like B and A, respectively.

expected that DAugNet would perform better than a classifier trained only on real source domains.

In the rest of this section, we first describe how we concurrently switch the styles of two domains. We then detail how we perform style transfer between multiple domains. We finally explain the architecture of DAugNet and its overall training pipeline for multi-source, multi-target, and life-long domain adaptation.

A. Style Transfer Between Two Domains

To perform style transfer between two domains, we use one encoder, one decoder, and one multi-task discriminator. Throughout this sub-section, we refer to one of the domains as A and the other one as B for the sake of simplicity.

Before the training starts, we initially generate a unique and constant style code for each domain. We denote by $S_A = \{\gamma_A, \beta_A\}$ and $S_B = \{\gamma_B, \beta_B\}$ the style codes of A and B. We initialize each parameter of $S_A$ and $S_B$ (i.e., $\gamma$ and $\beta$) with 128 values drawn from the uniform distribution that ranges between 0 and 1. To generate a fake domain stylized as the other one, we first extract an embedding from the domain using the encoder. We then normalize the embedding via adaptive instance normalization (AdaIN) [56] that is defined as:

$$\text{AdaIN}(x, \gamma, \beta) = \gamma \left( \frac{x - \mu(x)}{\sigma(x)} \right) + \beta,$$

where $x$ denotes the final activation of the encoder from one domain, and $\gamma$ and $\beta$ correspond to the style code parameters of the other domain. We finally decode the normalized embedding via the decoder. As depicted in Fig. 4, the embedding extracted from one domain through the encoder consists of 128 channels, each of which is scaled and shifted by the style code parameters of the other domain with Eq. 1 before feeding the embedding to the decoder. Decoding the normalized embedding of A with $S_B$ and the embedding of B with $S_A$ results in generating fake A stylized as B and fake B with the style of A. StandardGAN [15] aims at learning the parameters of $S_A$ and $S_B$ from A and B by domain specific so-called style encoders. Such approach requires using a different style encoder for each domain. However, we have discovered that it is possible to combine the content of A with the style B (or vice versa) using randomly initialized and unique $S_A$ and $S_B$ parameters instead of trying to learn them.

After combining A with $S_B$ and B with $S_A$ to generate fake A and fake B, we need to ensure that the data distributions of A and fake B, and B and fake A are as similar as possible. To overcome this issue, one could use adversarial learning to force the distributions of fake A and fake B to be similar to those of B and A, respectively. To perform style transfer between two domains, the current state-of-the-art I2I approaches like SemiI2I [10], CycleGAN [20], UNIT [22] use two discriminators, where one discriminates between A and fake B, and the other one tries to distinguish B from fake A. However, as discussed in Sec. I, one of the challenges is to use as small number of networks as possible to easily extend the method to multi-domain adaptation problem. Hence, instead of including multiple discriminators, we propose to use only one multi-task discriminator. As shown in Fig. 4, our discriminator comprises several shared layers and multiple domain specific layers. Each domain specific layer tries to understand whether the given data are fake or belong to that domain. Let us also remark that each domain specific layer of our discriminator
outputs a two-dimensional matrix. Each element of this matrix determines whether different parts of the input image are real or fake. We then take average of all the elements to generate a scalar quantifying the realness of the input. Our generator is depicted in Fig. 3. Let us denote by $D_A$ and $D_B$ the discriminator’s outputs for $A$ and $B$, and by $G$ the generator. The adversarial loss for the discriminator is defined as:

$$L_{D_{adv}}(X, Y) = -\mathbb{E}_x[\log(D_X(x))] + \mathbb{E}_y[\log(1-D_X(G(y)))],$$

where $X$ and $Y$ denote two domains, and $x$ and $y$ correspond to patches sampled from these domains. The adversarial loss for the generator is described as:

$$L_{G_{adv}}(X, Y) = -\mathbb{E}_y[\log(D_X(G(y)))]$$

We compute the overall adversarial losses for the discriminator and the generator as:

$$L_{D_{adv}} = L_{D_{adv}}(A, B) + L_{D_{adv}}(B, A)$$

and

$$L_{G_{adv}} = L_{G_{adv}}(A, B) + L_{G_{adv}}(B, A).$$

Another challenge is to keep $A$ and fake $A$, and $B$ and fake $B$ semantically consistent. Otherwise fake $A$ and fake $B$ would not match with the ground-truth of $A$ and $B$, and could not be used to train a classifier. To enforce the semantic consistency, we define several constraints. Firstly, after we generate fake $A$ and fake $B$, we combine their contents with the style code parameters of a randomly selected domain, and decoding the normalized embedding. We then perform multi-domain style transfer by sampling a patch from randomly selected two domains among $N$ and minimizing Eqs. (9) and (10) in each training iteration. After a sufficiently long training process, we are able to generate fake $N - 1$ domains from each one. Our constraints explained in Sec. III-A enforce each fake domain to be representative for a different domain and to be semantically consistent with the original domain.

Once the training process of the initial unsupervised and multi-domain style transfer is completed, we move on to training DAugNet. As illustrated in Fig. 4, the main components of DAugNet is the data augmentor and the classifier. In this stage, the data augmentor is frozen, only the classifier is updated. The combination of the encoder and the decoder forms a data augmentor. Both the encoder and the decoder are shallow networks, where the former consists of six layers and the latter is composed of only two layers (see Fig. 1). In each training iteration of DAugNet, we randomly sample a batch of training patches from $N_s$ source domains as well as their corresponding ground-truth. With 0.9 probability, the data augmentor diversifies the batch by extracting its embedding via the encoder, scaling and shifting the embedding for each patch with the style code parameters of a randomly selected domain, and decoding the normalized embedding. We then use the diversified batch and the ground-truth for the original batch to update the classifier by minimizing the classification loss $L_{class}$ defined as:

$$L_{class} = \lambda_5 L_{sigmCE}(y, \hat{y}) + \lambda_6 L_{soft_IoU}(y, \hat{y}),$$

where $L_{sigmCE}$, $L_{soft_IoU}$, $y$, and $\hat{y}$ denote sigmoid cross entropy loss, soft IoU loss, class labels, and the predictions, respectively. Diversifying the original batch with the
TABLE I
THE DATA SET.

| City (Country) | Class percentages (%) | 256 × 256 patches |
|----------------|------------------------|-------------------|
|                | Building | Road | Tree | 768 | 1536 |
| Villach (AT)   | 9.26     | 10.63| 19.91| 749 |
| Sankt Pölten (AT) | 6.68 | 6.39 | 25.13| 1558 |
| Bad Ischl (AT) | 5.51     | 6.0  | 35.38| 457 |
| Salzburg Stadt (AT) | 9.44 | 8.69 | 23.88| 2496 |
| Liechtenitz (AT) | 7.00 | 7.37 | 16.78| 572 |
| Bourges (FR)   | 9.81      | 10.52| 14.83| 1188 |
| Lille (FR)     | 18.36     | 12.71| 15.40| 2181 |
| Béziers (FR)   | 19.09     | 17.62| 10.91| 407 |
| Albi (FR)      | 17.20     | 14.48| 15.19| 413 |
| Vaduz (LI)     | 3.57      | 4.30 | 33.69| 1612 |

proposed diversification shallow data augmentor has two advantages. Firstly, such diversification approach allows the classifier to learn from the data that are representative for all \( N \) domains. Therefore, for target domains, we can expect DAugNet to achieve a better performance than naively training a classifier on the real data. Secondly, the diversification can be carried out in an online fashion, rather than storing all the fake domains in the pre-processing step and loading them when training the classifier. In each iteration, we deactivate the data augmentor with 10% chance for the classifier to learn from the real data as well.

To better explain life-long, multi-source, and multi-task adaptation problem, let’s assume that we are provided additional \( M \) domains composing of \( M_s \) annotated source and \( M_t \) unlabelled target domains with different data distributions. The problem now becomes segmenting \( N_t + M_t \) domains. To solve this problem, we first generate unique style codes for newly added \( M_t \) domains. We load the style codes of the initial \( N_t \) domains and the pre-trained weights for the encoder, the decoder, and the classifier of DAugNet. We also initialize the parameters of the discriminator’s shared layers as well as its domain specific layers for the previous domains with the pre-trained weights. For the current \( M \) domains, we add additional \( M \) domain specific layers to the discriminator with random initialization. Then we conduct the initial training to perform style transfer between \( N_t + M_t \) domains. This time, selecting two random domains, we randomly choose one domain among the current \( M \) domains and another one from all \( M + N_t \) domains we have at hand. With this sampling strategy, we guarantee that at least one of the domains is selected from the newly added ones. We finally fine-tune DAugNet with the same diversification method as explained above. Let us remark that every time we receive a new domain, we do not increase the number of networks. We only add a new layer to the discriminator. Considering that the networks with hundreds of layers can be fit into the current GPUs as in ResNet-152 [59], it is feasible to add many layers to the discriminator.

IV. EXPERIMENTS

A. Data Set

Our data set consists of Pléiades images collected over five cities in Austria, four cities in France, and one city in Liechtenstein. We were given the annotations for building, road, and tree classes by the data providers. Table I reports city names and what percentage of the pixels belong to each class. The image from Leibnitz is composed of near-infrared, red, green bands, and its spatial resolution is 0.5 m. The rest of the cities comprise red, green, and blue channels, and their resolution is 1 m. We reduce the resolution of the image from Leibnitz to 1 m so as to make our data set homogenous in terms of spatial resolution. The images in our data set cover the total area of 663.02 km².

In the pre-processing step, we split all the images into 256 × 256 smaller patches with an overlap of 32 pixels. We use these patches in every stage of our training pipeline (i.e., both the multi-domain style transfer part and the process of training DAugNet). We indicate the number of patches belonging to each city in Table I.

B. City-to-city Adaptation

In our first experimental setup, we perform city-to-city adaptation between Villach and Bad Ischl. A close-up from each city is depicted in Figs. 5a and 6a. We first use Villach as the source city and Bad Ischl as the target. We then switch the source and the target cities. In both cases, we assume that the source image is annotated and the target image is unlabeled. We compare our method with thirteen different approaches. The compared methods can be categorized into four groups: naive, adapting the classifier, data standardization, and image-to-image translation.

The approach belonging to the first category corresponds to training a U-net [7] on the source image and segmenting the target city without doing any domain adaptation. As confirmed by Table II when there exist a large data distribution difference between the source and the target cities, U-net fails to generate high quality maps. Especially for Bad Ischl, the quantitative results are extremely poor.

We also apply AdaptSegNet [24], which is an approach belonging to the second group, to our data set. This network architecture simultaneously learns from the source city and aligns the features extracted from both cities to train a domain agnostic classifier. The authors use DeepLab v2 [60] as the classifier. The feature alignment can be executed in one or multiple layers. In Table II AdaptSegNet Single and AdaptSegNet Multi respectively correspond to the proposed method when the alignment is done in the last and in the last two layers before upsampling. However, the quantitative results prove that this method does not exhibit a good performance on domain adaptation of satellite images.

Among the methods based on data standardization, we compare our approach with gray-world algorithm [37], histogram equalization [16], and z-score normalization [35]. We first standardize images from both cities. We then train a U-net on the standardized source image and segment the standardized target data. Although these standardization methods enable U-net to better generalize, the improvement is unsatisfactory. As shown in Figs. 5 and 6 while the data distributions of the
### TABLE II

IoU scores for the first experiment (City-to-city adaptation).

| Method                     | Source: Bad Ischl ⇒ Target: Villach | Source: Villach ⇒ Target: Bad Ischl |
|----------------------------|--------------------------------------|--------------------------------------|
|                            | Building  | Road  | Tree  | Overall | Building  | Road  | Tree  | Overall |
| U-net [11]                 | naive     | 29.00 | 3.78  | 58.52   | 30.43     | 5.12  | 0.08  | 0.00    | 1.73    |
| AdaptSegNet Single [24]    | adapting  | 6.01  | 4.37  | 10.43   | 6.94      | 3.06  | 2.71  | 10.23   | 5.33    |
| AdaptSegNet Multi [24]     | classifier| 24.59 | 9.02  | 56.08   | 29.86     | 14.26 | 4.46  | 24.66   | 14.46   |
| Gray-world [37]            | data      | 11.98 | 25.66 | 60.91   | 32.85     | 34.30 | 29.08 | 58.12   | 40.50   |
| Hist. Equalization [16]    | standardization | 23.28 | 23.36 | 64.31   | 36.98     | 34.08 | 22.81 | 64.93   | 40.60   |
| Hist. Matching [16]        | image     | 17.86 | 20.53 | 61.64   | 33.34     | 23.20 | 34.22 | 52.00   | 36.47   |
| U-net trained on data [22] | image     | 8.20  | 20.18 | 64.81   | 27.73     | 2.27  | 0.13  | 0.00    | 0.80    |
| UNIT [22]                  | to        | 6.72  | 0.00  | 37.32   | 14.68     | 2.41  | 0.00  | 1.22    | 1.21    |
| MUNIT [11]                 | translation | 0.00  | 0.00  | 15.14   | 5.05      | 0.00  | 0.30  | 0.00    | 0.10    |
| CycleGAN [20]              | image     | 25.80 | 25.20 | 66.07   | 39.03     | 42.01 | 12.15 | 76.39   | 43.52   |
| ColorMapGAN [9]            | 42.28     | 40.25 | 65.28 | 49.27   | 52.00     | 42.29 | 47.28 | 47.19   |
| SemI2I [10]                | 40.18     | 32.66 | 66.83 | 46.56   | 49.32     | 41.06 | 70.18 | 53.52   |

Fig. 5. Bad Ischl and its modified versions to segment Villach. (a) Bad Ischl, (b) gray-world [37], (c) histogram equalization [16], (d) z-score normalization [35], (e) histogram matching [16], (f) UNIT [22], (g) MUNIT [11], (h) DRIT [12], (i) CycleGAN [20], (j) ColorMapGAN [9], (k) SemI2I [10], (l) DAugNet (ours).

Fig. 6. Villach and its modified versions to segment Bad Ischl. (a) Villach, (b) gray-world [37], (c) histogram equalization [16], (d) z-score normalization [35], (e) histogram matching [16], (f) UNIT [22], (g) MUNIT [11], (h) DRIT [12], (i) CycleGAN [20], (j) ColorMapGAN [9], (k) SemI2I [10], (l) DAugNet (ours).
images get close to each other with these approaches, we still observe some differences.

To evaluate image-to-image translation based methods, we first generate a target stylized fake source city. Afterwards, we train a U-net on the fake source city and evaluate it on the target city. We provide the results for histogram matching [16], UNIT [22], MUNIT [11], DRIT [12], CycleGAN [20], ColorMapGAN [9], and SemI2I [10]. As illustrated in Figs. 5 and 6, the fake images generated by UNIT [22], MUNIT [11], and DRIT [12] are semantically inconsistent with the real source images. Therefore, U-net learns from data, where image and the ground-truth do not match. As a consequence, the quantitative results for these approaches are poor. Histogram matching does not take into account the contextual information. For example, fake image generated by histogram matching from Bad Ischl contains a lot of buildings with violet or cyan rooftops. However, such buildings do not exist in Villach. Since the fake source image is not representative for the target image, the performance of this approach is unsatisfactory. CycleGAN generates blurry fake source images, which negatively affects the performance of U-net. ColorMapGAN and SemI2I are better performers than the other ones.

When we train the multi-domain style transfer part of our solution, we set $\lambda_1$, $\lambda_2$, $\lambda_3$, and $\lambda_4$ parameters in Eqs. 9 and 10 to 1, 10, 10, and 100, respectively. We have found these values empirically. We train our style transfer method for 25 epochs. The learning rate for the initial 15 epochs is 0.0001. We reduce it gradually in the rest of the epochs as:

\[
LR = 0.0001 \times \frac{\text{num\_epochs} - \text{epoch\_no}}{\text{num\_epochs} - \text{decay\_epoch}},
\]

where LR, num\_epochs, epoch\_no, and decay\_epoch denote the current learning rate, total number of epochs, current epoch no, and the epoch no where we start reducing the learning rate. As the optimizer, we use Adam algorithm [61]. We set the beta coefficients of Adam to 0.5 and 0.999, respectively. When we train DAugNet, our data augmentor generates only target stylized image batches, since there are only two cities in this experimental setup. Note that any network architecture can be used as the classifier of DAugNet. We compare DAugNet with the same standardization approaches explained in the previous experimental setup. Once all the data are standardized, we train a U-net on the standardized source images for 100 epochs. We compare our framework with the same standardization approaches explained in the previous experimental setup.

| Table III | reports the IoU scores for multi-source and multi-target experimental setup. Our first observation is that adding more source cities with different data distributions bridges the domain gap between the source and the target cities. Thus, the classifier better generalizes to new geographic locations. For instance, although we use Bad Ischl as the target city both in this experiment and in the previous one, the quantitative results of each method for this city are significantly higher in this setting. Secondly, in some cases, the performances of the standardization algorithms are unstable. All of these algorithms allow the classifier to exhibit a considerably better performance on Bad Ischl, whereas the quantitative results for Vaduz are either roughly on par with or even worse than naive U-net. As depicted in Fig. 7 our multi-domain style transfer method is able to stylize each city as another one. As mentioned earlier, diversifying a batch of image patches when training DAugNet with the help of the data augmentor enables DAugNet to learn from data that are not only representative for the source cities but also representative for the target cities. Hence, DAugNet achieves a better performance than the others. Fig 8 shows a close-up from each city, its ground-truth, and the predictions.

### D. Life-long Domain Adaptation

In the third experiment, we assume that in addition to the ones in the second experiment, we receive new annotated source images from Béziers, Salzburg Stadt, Albi, and unlabeled target data from Leibnitz. The image from Leibnitz consists of near-infrared, red, and green channels, whereas the other nine cities are composed of red, green, and blue spectral bands. This experimental setup has two goals. Firstly, we wish to verify whether our method can perform style transfer between ten images having different data distributions even when the type of spectral bands are different. Secondly, we want to observe the relevance of our solution for the life-long adaptation setting, where new source and target images are added. In this experimental setup, we have seven source and three target cities in total.

We first generate unique style codes for the new cities and train our style transfer method as described in Sec. III-B for 50 epochs. As in the first two experiments, we use Adam optimizer with the initial learning rate of 0.0001. We reduce the learning rate after the 25th epoch via Eq. 12. We train DAugNet as described in Sec. III-B for 100 epochs. We compare our framework with the same standardization approaches as the previous experiment. For each compared standardization method, we fine-tune the U-net trained in the second experiment for 100 epochs.
Fig. 7. Style transfer between the images used in the second experiment. The cells with red bounding boxes are real data. The rest of the cells represent the fake data generated by our multi-domain style transfer approach.

**TABLE III**

| Method | Bad Ischl | Vaduz |
|--------|-----------|-------|
|        | Building  | Road  | Tree  | Overall | Building  | Road  | Tree  | Overall |
| U-net [1] | 35.48 | 26.36 | 51.24 | 37.69 | 12.49 | 18.68 | 58.34 | 29.84 |
| U-net trained on data | Gray-world [37] | 50.30 | 43.00 | 70.41 | 54.57 | 19.45 | 19.39 | 18.85 | 19.29 |
|          | Histogram Equalization [16] | 42.32 | 38.21 | 75.16 | 51.90 | 19.32 | 20.64 | 68.50 | 36.15 |
|          | Z-score normalization [35] | 39.77 | 41.86 | 78.53 | 53.39 | 15.40 | 24.16 | 58.94 | 32.83 |
| DAugNet | 59.70 | 49.05 | 83.39 | 64.05 | 51.83 | 34.41 | 69.74 | 51.99 |
As confirmed by Table IV, because of the new three source cities, the IoU scores of all the methods for Bad Ischl and Vaduz are higher than in the second experiment. For example, the performance of U-net especially for tree class significantly improves. It is probably because trees in the new cities are more representative for the target cities. For instance, as can be seen in Fig. 9 the data distributions of trees in Salzburg stadt and Vaduz, and Albi and Bad Ischl seem close. Naive U-net slightly better detects trees in Vaduz than DAugNet. This slight performance difference probably stems from some artifacts added by our GAN architecture in the process of generating fake cities. However, DAugNet outperforms the compared approaches for all the other classes. When it comes to segmenting Leibnitz, the performances of the compared methods are extremely poor, mainly because the spectral bands of this image do not comprise red, green, and blue channels. On the other hand, as shown in Fig. 9 our style transfer method can easily convert red, green, blue images to near infrared, red, green ones. As a consequence, DAugNet significantly outperforms the others on Leibnitz, since it learns from data consisting of both red, green, blue and near infrared, red, green bands. Fig. 10 depicts a close-up from each city.
Fig. 10. Target images used in the third experiment, their ground-truth, and the predictions. Red, white, and green pixels represent building, road, and tree classes, respectively. The pixels in black do not belong to any class.

TABLE V
QUANTITATIVE RESULTS FOR THE FIRST ABLATION STUDY.

| City  | Result type | Building | Road | Tree | Overall |
|-------|-------------|----------|------|------|---------|
|       | IoUs for run 2 | 59.91 | 50.00 | 82.70 | 64.20 |
|       | IoUs for run 3 | 60.50 | 50.44 | 83.64 | 64.86 |
|       | standard dev.  | 0.34 | 0.58 | 0.40 | 0.35 |
| Bad Ischl | IoUs for run 2 | 51.65 | 35.42 | 62.56 | 49.88 |
|       | IoUs for run 3 | 51.72 | 32.76 | 66.98 | 50.49 |
|       | standard dev.  | 0.07 | 1.10 | 2.96 | 0.89 |
| Vaduz | IoUs for run 2 | 50.07 | 49.45 | 81.71 | 63.42 |
|       | IoUs for run 3 | 59.95 | 52.27 | 83.16 | 65.12 |
|       | standard dev.  | 0.36 | 1.43 | 0.74 | 0.70 |
|       | IoUs for run 2 | 46.03 | 37.31 | 64.80 | 50.73 |
|       | IoUs for run 3 | 50.07 | 34.14 | 65.86 | 48.68 |
|       | standard dev.  | 2.43 | 1.43 | 2.12 | 1.36 |

TABLE VI
QUANTITATIVE RESULTS FOR THE SECOND ABLATION STUDY.

| City  | Result type | Building | Road | Tree | Overall |
|-------|-------------|----------|------|------|---------|
|       | IoUs for run 2 | 59.10 | 49.45 | 81.71 | 63.42 |
|       | IoUs for run 3 | 59.95 | 52.27 | 83.16 | 65.12 |
|       | standard dev.  | 0.36 | 1.43 | 0.74 | 0.70 |
| Bad Ischl | IoUs for run 2 | 50.07 | 37.31 | 64.80 | 50.73 |
|       | IoUs for run 3 | 46.03 | 34.14 | 65.86 | 48.68 |
|       | standard dev.  | 2.43 | 1.43 | 2.12 | 1.36 |
| Vaduz | IoUs for run 2 | 50.07 | 49.45 | 81.71 | 63.42 |
|       | IoUs for run 3 | 59.95 | 52.27 | 83.16 | 65.12 |
|       | standard dev.  | 0.36 | 1.43 | 0.74 | 0.70 |

TABLE VII
DROP IN IOU SCORES WHEN EDGE LOSS IS DEACTIVATED.

| City  | Building | Road | Tree | Overall |
|-------|----------|------|------|---------|
| Bad Ischl | 2.04 | 2.25 | 2.59 | 2.30 |
| Vaduz | 5.88 | 2.00 | 14.81 | 7.56 |

used in this experiment, its ground-truth, and the predictions.

E. Ablation Studies

a) Effect of random diversification: As explained earlier, the data augmentor in DAugNet randomly diversifies the training batch before passing it to the classifier. In the first ablation study, we analyze the robustness of DAugNet to random diversification. We train DAugNet two more times from scratch on the source cities in the second experimental setup and segment Bad Ischl and Vaduz. Note that we do not train the initial multi-domain style transfer part of our pipeline again to evaluate the effect of only random diversification. Table V reports IoU scores of the two runs. In the table, we also indicate the standard deviation of IoUs in the second experiment and IoUs of these two runs. Because the standard deviations are quite small, we conclude that our framework is robust to random diversification.

b) Effect of random style codes and diversification: We repeat the whole training pipeline of our approach in the second experiment two times. Fig. II depicts the randomly initialized 128 γ and β values for Villach in the second experiment and in the two runs of this ablation study. In Table VI, we report IoU scores of these two runs and the standard deviations. Although the style codes (i.e., γ and β) of the cities are different in each run, the standard deviations for IoUs are very small. In conclusion, our solution is robust to random style codes and random diversification.

c) Effect of edge loss: We deactivate the edge loss by setting λ₄ in Eq. 9 to zero and repeat the second experiment to evaluate the effect of the edge loss. As can be seen in Fig. 12, when λ₄ is zero, our style transfer approach generates slightly blurred fake images, which causes DAugNet to exhibit a worse performance. The drop in IoU scores is reported in Table VII. Note that the other terms in Eq. 9 are mandatory; therefore, λ₁, λ₂, and λ₃ cannot be set to zero.

d) U-net vs. DAugNet training time comparison: Because the data augmentor needs to diversify the batch in each training iteration, the training time of DAugNet is longer than that of U-net. Table VIII reports the time required to train DAugNet and U-net on the source data in the second and in the third experiments on an Nvidia GeForce GTX 1080 Ti GPU. Due to its simple architecture, the data augmentor prolongs training time of the classifier for only 18 minutes in the second experiment, and for 28 minutes in the third
experiment. Table IX reports the time needed to train the multi-domain style transfer part of our framework in the second and in the third experiments.

e) Scalability of the proposed method: As mentioned in Sec. IV, significant technological advancements in satellite sensors and a large number of satellite missions have made massive volume of remote sensing data available. Therefore, it is of paramount importance to propose highly scalable methods that can efficiently process a lot of satellite images. To demonstrate that our method is capable of dealing with many satellite images, we perform style transfer between twenty images collected from different cities in Europe. Spectral bands of one image consists of near-infrared, red, green bands, whereas the other images comprise red, green, and blue channels. As depicted in Fig. 12, our style transfer method successfully stylizes each images like any other. This ablation study proves that our method is suitable for large-scale segmentation problems.

V. CONCLUDING REMARKS

The satellite missions launched in the last decade have enabled us to collect huge volume of remote sensing data over the whole earth every day. Because of various atmospheric conditions, sensor characteristics, and differences in acquisition, remote sensing images collected from separate geographic locations in distinct times tend to have largely different data distributions. The data distribution difference between source and target images prevents the machine learning models from generating precise maps. Moreover, scarcity of annotated data motivates us to propose methods that are robust to such distribution difference. In addition, it is of paramount importance to introduce new approaches that can adapt to continually growing data.

In this context, we presented a new approach for multi-source, multi-target, and life-long domain adaptation problem. In the pre-processing stage of our method, we learn how to perform style transfer between multiple source and target domains using only one encoder, one decoder, and one discriminator. A subset of the networks used in this stage constitutes a data augmentor. We then train the proposed DAugNet comprising the data augmentor and a classifier. In each training iteration, the data augmentor randomly diversifies the training batch before passing it to the classifier. As a consequence, the classifier is more robust to the data distribution difference between the images, since it learns from the data that are representative for all source and target cities.

The state-of-the-art methods in the literature either aim at solving single-source and single target adaptation problem or use multiple networks to deal with multi-source domain adaptation problem. However, because the number of networks in multi-domain style transfer stage (pre-processing step) of our method is constant regardless of the number of cities,
As future work, we plan to investigate between how many cities our approach can perform style transfer. We will also observe whether our solution can be used for domain adaptation of other types of remote sensing data such as aerial and Sentinel images.

REFERENCES

[1] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in International Conference on Medical image computing and computer-assisted intervention–MICCAI. Springer, 2015, pp. 234–241.
IEEE International Conference on Computer Vision, 2019, pp. 1406–1415.

B. Gholami, P. Sahu, O. Rudovic, K. Bouwmals, and V. Pavlovic, “Unsupervised multi-target domain adaptation: An information theoretic approach,” IEEE Transactions on Image Processing, vol. 29, pp. 3993–4002, 2020.

H. Yu, M. Hu, and S. Chen, “Multi-target unsupervised domain adaptation without exactly shared categories,” arXiv preprint arXiv:1809.08852, 2018.

M. Wulfmeier, A. Bewley, and I. Posner, “Incremental adversarial domain adaptation for continually changing environments,” in 2018 IEEE International conference on robotics and automation (ICRA), 2018, pp. 1–9.

Z. Zhang, S. Lathuilière, A. Pilzer, N. Sebe, E. Ricci, and J. Yang, “Online adaptation through meta-learning for stereo depth estimation,” arXiv preprint arXiv:1904.08462, 2019.

A. Kumar, T. Ma, and P. Liang, “Understanding self-training for gradual domain adaptation,” arXiv preprint arXiv:2002.11361, 2020.

C. Persello and L. Bruzzone, “Active learning for domain adaptation in the supervised classification of remote sensing images,” IEEE Transactions on Geoscience and Remote Sensing, vol. 50, no. 11, pp. 4468–4483, 2012.

G. Matasci, D. Tuia, and M. Kanevski, “Svm-based boosting of active learning strategies for efficient domain adaptation,” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 5, no. 5, pp. 1335–1343, 2012.

C. Deng, X. Liu, C. Li, and D. Tao, “Active multi-kernel domain adaptation for hyperspectral image classification,” Pattern Recognition, vol. 77, pp. 306–315, 2018.

S. Ghassemi, A. Fiandrotti, G. Francini, and E. Magli, “Learning and adapting robust features for satellite image segmentation on heterogeneous data sets,” IEEE Transactions on Geoscience and Remote Sensing, vol. 57, no. 9, pp. 6517–6529, 2019.

X. Huang and S. Belongie, “Arbitrary style transfer in real-time with adaptive instance normalization,” in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1501–1510.

R. M. Haralick, K. Shanmugam, and I. H. Dinstein, “Textural features for image classification,” IEEE Transactions on systems, man, and cybernetics, no. 6, pp. 610–621, 1973.

G. Máttyus, W. Luo, and R. Urtasun, “Deeproadmapper: Extracting road topology from aerial images,” in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 3438–3446.

K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” IEEE TPAMI, vol. 40, no. 4, pp. 834–848, 2018.

D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980, 2014.