ADVERSARIAL DOMAIN ADAPTATION FOR THE CLASSIFICATION OF AERIAL IMAGES AND HEIGHT DATA USING CONVOLUTIONAL NEURAL NETWORKS

Dennis Wittich*, Franz Rottensteiner

Institute of Photogrammetry and GeoInformation, Leibniz Universiät Hannover, Germany
- (wittich, rottensteiner)@ipi.uni-hannover.de

ICWG II/III: Pattern Analysis in Remote Sensing

KEY WORDS: Domain Adaptation, Segmentation, Classification, Fully Convolutional Networks

ABSTRACT:

Domain adaptation (DA) can drastically decrease the amount of training data needed to obtain good classification models by leveraging available data from a source domain for the classification of a new (target) domains. In this paper, we address *deep DA*, i.e. DA with deep convolutional neural networks (CNN), a problem that has not been addressed frequently in remote sensing. We present a new method for semi-supervised DA for the task of pixel-based classification by a CNN. After proposing an encoder-decoder-based fully convolutional neural network (FCN), we adapt a method for adversarial discriminative DA to be applicable to the pixel-based classification of remotely sensed data based on this network. It tries to learn a feature representation that is domain invariant; domain-invariance is measured by a classifier's incapability of predicting from which domain a sample was generated. We evaluate our FCN on the ISPRS labelling challenge, showing that it is close to the best-performing models. DA is evaluated on the basis of three domains. We compare different network configurations and perform the representation transfer at different layers of the network. We show that when using a proper layer for adaptation, our method achieves a positive transfer and thus an improved classification accuracy in the target domain for all evaluated combinations of source and target domains.

1. INTRODUCTION

The first step to generate maps from remotely sensed data is pixel-wise classification (or semantic segmentation) of these data. Deep learning based on convolutional neural networks (CNN) or, in the context of pixel-wise classification, fully convolutional neural networks (FCN) (Long et al., 2015a) is surpassing classical machine learning approaches. One of the keys to the success of CNN was the availability of large collections of annotated images (Krizhevsky et al., 2012). In remote sensing, there is only a limited amount of freely available data with annotations; see (Zhu et al., 2017) for a recent overview. The large variations of the appearance of objects, for instance due to seasonal effects, lighting conditions, geographical variability of objects and sensor properties, makes it impossible to apply classifiers trained on such existing data directly to new data without a significant drop of classification accuracy. Consequently, ground truth labels are usually generated by manual pixel-wise annotation based on subsets of the images to be classified, a very tedious and time-consuming task.

One strategy to mitigate or even avoid the efforts required for manual annotation of new training samples is transfer learning (TL) (Pan & Yang, 2010). In TL, one tries to use information from a *source domain*, in which training samples are supposed to be abundant, to solve a learning problem in a *target domain*, where only limited or no training data are available, in a better way. The data and the learning problems may differ between domains, but they must be related. TL is habitually applied in deep learning when networks that are pre-trained on a source domain dataset are re-trained to be applied to a target domain using a limited amount of new training samples (Yosinski et al., 2014). A specific setting of TL is *Domain Adaptation* (DA), where the domains only differ by the joint distribution of the features and the class labels. This corresponds to a situation where we have a set of training images labelled in the past (source domain) that we want to use to train a classifier for a new set of images (target domain) acquired with a sensor of the same type and with similar ground resolution. While the class structure remains unchanged, the objects may have a different appearance. We are mainly interested in methods for adapting the classifier to the target domain *without any new training samples*. Following (Tuia et al., 2016) we refer to this setting as *semi-supervised DA*. *Deep DA*, i.e. DA for deep learning, is a well-studied problem for tasks such as the prediction of a single label per image. However, there is much less work on pixel-wise classification (Wang & Deng, 2018), and there is hardly any work transferring these principles to remote sensing.

In this paper, we describe a new approach for DA for the pixel-wise classification of aerial imagery and derived data. First, we present an encoder-decoder FCN with skip connections based on U-Net (Ronneberger et al., 2015) and adapted to remote sensing data similarly to (Yang et al., 2019). We use separated encoder branches for the multispectral image and a digital surface model (DSM) to apply late fusion (Audeberg, 2018) and design the network so that removing the skip connections results only in a minor drop of quality. Using the Vaihingen dataset of (Wegner et al., 2017), we show that our FCN achieves results close to the state of the art. The main focus of the paper is on DA based on this FCN. We expand adversarial discriminative DA (ADDA) (Tzeng et al., 2017) for representation transfer to be applicable to pixel-wise classification. As it is unclear which layer of the network is optimal for representation transfer, we compare different variants, trying to achieve a domain-invariant feature representation at different layers of the network. Our DA method is evaluated using a dataset consisting of three domains. Our

* Corresponding author

experiments show that our FCN model achieves results close to the state of the art and that our DA approach achieves a positive transfer in all cases if an appropriate layer for representation transfer is used. We show that representation transfer before data fusion and removing the skip connections yield the best results.

2. STATE OF THE ART

We start with a brief introduction to DA based on (Tuia et al., 2016) to introduce our nomenclature. After that, we discuss the state of the art in semi-supervised DA in computer vision and remote sensing, focussing on the task of pixel-wise classification.

Following Tuia et al. (2016), we consider a source domain D^S and a target domain D^T, each associated with remotely sensed imagery. The domains are associated with the joint distributions $P^S(X,C)$ and $P^T(X,C)$, respectively, of the input variable X (associated with the image features) and the output variable C (associated with the class labels). In this paper, we assume the class structures (thus, C) and the feature space (X) to be identical. This setting is referred to as homogeneous DA in (Wang & Deng, 2018). The basic assumption of DA is that the joint distributions differ between domains, thus $P^S(X,C) \neq P^T(X,C)$. As $P(X,C) = P(C|X) \cdot P(X)$, this may be due to differences in the marginal distributions of the features, i.e. $P^S(X) \neq P^T(X)$, or due to differences in the posteriors, i.e. $P^S(C|X) \neq P^T(C|X)$. In any case, the differences between the distributions must not be too large. In D^S, we have a training data set T^S of n^S labelled training samples, each consisting of a tuple $(\mathbf{x}_i^S, \mathbf{c}_i^S)$ with $\mathbf{x}_i^S \in X$ and $\mathbf{c}_i^S \in C$ (note that in our application, $(\mathbf{x}_i^S, \mathbf{c}_i^S)$ corresponds to a labelled image patch, hence $\mathbf{c}_i^S$ is a vector with one class label per pixel in $\mathbf{x}_i^S$). In semi-supervised DA, the information available in D^T consists merely of a set U^T of n^T unlabelled samples (in our case: image patches) $\mathbf{x}_i^T \in X$. The task of DA is to use the data T^S and U^T to learn a classifier that predicts the unknown labels $\mathbf{c}_i^T$ in the target domain.

According to Tuia et al., (2016), DA can be based on instance transfer or on representation transfer. Instance transfer aims at adapting the classifier from the source to the target domain by using semi-labelled samples, i.e. target samples receiving their class labels from the current state of the classifier, e.g. (Bruzzone et al., 2008). On the other hand, representation transfer tries to find mappings from the feature spaces of both domains to a common representation space such that a shared classifier can be applied. In remote sensing, this is often done by finding a mapping that minimizes a statistical distance between the domains, e.g. the maximum-mean discrepancy (MMD) (Matasci et al., 2015). Although Tuia et al. (2016) list many DA methods for remote sensing, none of them is based on deep learning.

In computer vision, DA based on CNN (deep visual DA) is a very active field of research; see (Wang & Deng, 2018) for a recent overview. In this context, representation transfer is the most relevant approach for DA. According to Wang and Deng (2018), the main groups are *distance-based* and *adversarial approaches*. An example for a distance-based approach is (Long et al., 2015b). The authors train two different networks for mapping the features of both domains to a joint representation, using a shared network for classification of source and target samples. In addition to a classification loss for the source training samples, they introduce a loss that minimizes the MMD distance between several activation maps of the feature mapping networks from both domains to achieve a representation that is independent from the domain. In contrast, adversarial methods measure similarity of distributions by the capability of a domain classifier (the *discriminator*) to predict whether a sample is from the source or the target domain. The first example of such an approach is (Ganin et al., 2016). The network is also split into a feature extractor and a classifier, but there is only one feature extraction network. Beyond the classification loss for source samples, additional loss functions are related to the discriminator, which is fed the features generated for samples from both domains. They are designed to train the discriminator to predict the domain of a feature vector while at the same time pushing the extractor to produce features that cannot be distinguished by the discriminator, achieving a domain invariant representation. An alternative that is easier to train is adversarial discriminative DA (ADDA) (Tzeng et al., 2017), where separate feature extractors are trained for source and target domains. The source feature extractor and the classifier are learned independently from the target domain. After that, the target feature extractor is trained to produce a representation that fools the discriminator.

The examples cited so far solve the problem of predicting a single class label for an image. As noted by Wang and Deng (2018), there are relatively few papers addressing the problem of DA for the pixel-wise classification of images. An example is (Hoffman et al., 2016), adapting (Ganin et al., 2016) to the semantic segmentation of street scenes. A shared FCN is trained for the segmentation while domain-adversarial training is used to generate domain invariant features in the last layer of the encoder. Huang et al. (2018) adapt ADDA to semantic segmentation. Separate networks are used for the segmentation of source and target domain data. Multiple domain discriminators are used to match activation distributions at *different* layers of the source and target networks. The authors propose to use a L2-Distance based regularizer between source and target network to prevent a drift of the target networks parameters. Zhang et al. (2018) and Hoffman et al. (2017) expand domain adversarial representation transfer by adapting the visual appearance of images before passing them on to the feature extractor. This just compensates for differences in the marginal distributions $P^S(X)$ and $P^T(X)$ of the features.

Despite the recent developments in computer vision, we found only few papers that address deep domain adaptation in remote sensing. Beshmal et al. (2018) propose a method close to (Ganin et al., 2016) to create domain invariant representations for the classification of patches from aerial images, predicting one label per patch. They expand (Ganin et al., 2016) by using the reconstruction loss as regularizer such that the latent features hold enough information to reconstruct the input, but do not apply pixel-wise classification. To the best of our knowledge, the only publication specifically addressing DA for pixel-wise classification with CNNs in remote sensing is (Postadjian et al., 2018). However, the authors only address supervised DA and, thus, require annotated training samples in the target domain.

Our approach follows the concepts of Tzeng et al. (2017) and Huang et al. (2018), because we use separate feature encoders for source and target domain. We believe that using a shared feature encoder and joint training may lead to a deterioration of the source classifier if the domains are very different. The method most closely related to ours is (Huang et al., 2018). However, we use only one discriminator network because we argue that an adaptation of features close to the output layer of the network leads to an alignment to the label distribution, which can be harmful if the actual label distributions between source and target domains are very different. We thus concentrate on using a single discriminator network to match features and evaluate the representation transfer based on different intermediate layers of the network. Apart from that, we use another base network than

Huang et al. (2018) that is designed for using image and height data. We also use another regularization technique. In contrast to Zhang et al. (2018) and Hoffmann et al. (2017), we do not perform appearance adaptation, because it only matches the marginal distributions of the features. The scientific contributions of this paper are as follows:

- We adapt the principles of ADDA (Tzeng et al., 2017) to the pixel-based classification of aerial imagery and height data. To the best of our knowledge, this is the first application of this DA principle in remote sensing. Our experiments show that using this method in an appropriate setting leads to an increased classification accuracy after DA in all cases without any annotated training data samples in the target domain.
- We use a generalized formulation of representation transfer that allows us to carry out that transfer at arbitrary layers of the FCN. We compare different settings and show that transfer at the end of the decoder part of the network as in (Zhang et al., 2018) leads to suboptimal results.
- As a minor contribution, we propose an improved variant of the adapted U-Net structure of (Yang et al., 2019). This is mainly achieved by using zero-mean convolution for height data to make the model invariant to local terrain height changes and by changing the operations for down-sampling and up-sampling, which allows for creating a network without skip-connections that is better suited for DA while maintaining the classification accuracy of the original model.

3. ARCHITECTURE OF THE BASE FCN

In this section, we propose an FCN for the pixel-wise classification of multispectral orthophotos and DSMs.

3.1 Network Architecture

Like the U-Net (Ronneberger et al., 2015) variant in (Yang et al., 2019), our network is based on a multi-encoder decoder FCN with skip-connections (Figure 1). An input sample $\mathbf{x}_i \in X$ consists of a multispectral (MS) orthophoto (MSI_i) and a DSM (DSM_i) in the form of a height grid, both consisting of 640 x 640 pixels. The number of bands of the orthophoto may depend on the application, but in DA (Section 4), the images from different domains must have the same band configurations. The CNN delivers a label map $\mathbf{c}_i$ containing one class label $\mathbf{c}_{i,j} \in C$ for every pixel j of $\mathbf{x}_i$, where C is the label space. Both the image and height data are normalized as described in Section 5.1.

Like Yang et al. (2019), we have two separate encoder branches with different parameters, one for the MS image and one for the DSM. This differs from (Yang et al., 2019), where the second branch takes the DSM with two spectral bands as input. The encoder outputs of both branches are concatenated before the joint low-resolution representation is passed to the decoder part of the network. This corresponds to a *late fusion* of the MSI and the DSM. Yang et al. (2019) found this to achieve slightly better results than early fusion by applying a single encoder to a combination of the DSM and the MSI. It also allows for more flexibility in DA because we can choose to which branch of the network DA is applied. The decoder up-samples the low-resolution representation, resulting in a feature vector for each pixel to be classified. The last layer of the network is a softmax layer converting these feature vectors into class scores. Unlike Yang et al. (2019), we follow (Ronneberger et al., 2015) in using unpadded convolutions in all layers. Apart from the positive effect on accuracy (Ronneberger et al., 2015), we observe that unpadded convolutions decrease the required training time. In order to make the DSM encoder invariant to local terrain height changes, we apply zero-mean convolutions (Schlüter & Lehner, 2018), where the learned filters of the convolutional layers are reduced by their mean after each weight update. The normalization step for the n parameters $p_0, \ldots, p_n$ of filter f is

$$p_i^f = p_i^f - \frac{1}{n}\sum_{j=0}^{n} p_j^f . \quad (1)$$

In contrast to a normalization of the DSM based on the statistics of the dataset, zero-mean convolutions are also invariant to systematic terrain height changes inside one patch.

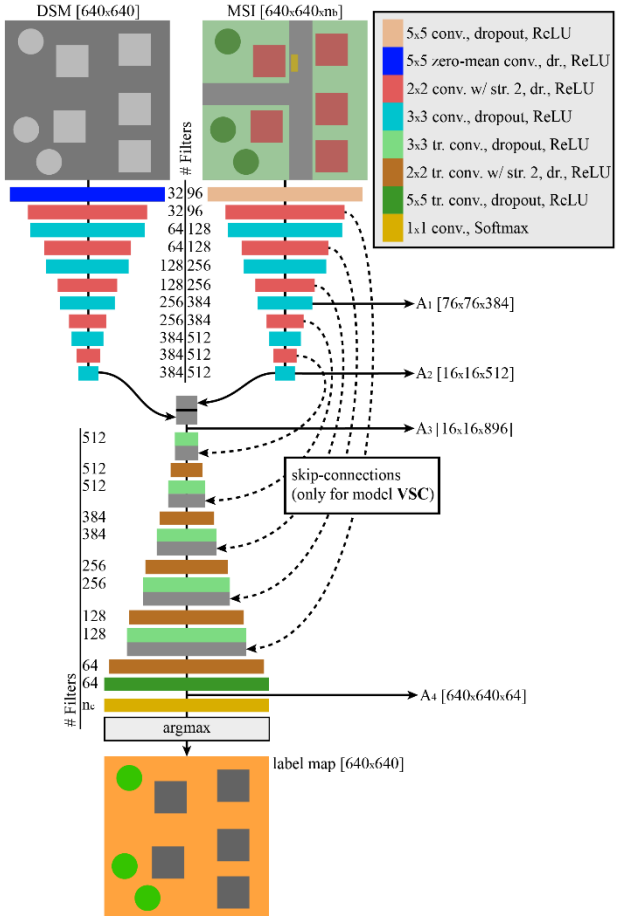

Figure 1: Proposed FCN structure. A_1 – A_4: layers used for DA in our experiments (cf. Sections 4 and 5).

Ronneberger et al. (2015) introduced skip connections between corresponding layers of the encoder and decoder, which was supposed to preserve object boundaries in a better way. Skip connections were also found to be beneficial for land cover classification in (Yang et al., 2019). Unlike Yang et al. (2019), we do not use skip connections from both encoders, but only from the colour band encoder to the decoder. This is motivated by the observation that DSMs, in particular if generated by image matching, are often inaccurate at the borders of elevated objects such as houses or trees. Preliminary experiments not reported in this paper have shown that an architecture with skip-connections only from the colour bands yields slightly better results than the same model with additional skip-connections from the height-encoder.

Our last modification compared to Ronneberger et al. (2015) and Yang et al. (2019) is related to the down-sampling and up-sampling layers of the network. We replace the pooling layers in the encoder by applying 2 x 2 convolutions with a stride of 2 along both spatial dimensions (Springenberg et al., 2015).

Similarly, the up-sampling operations in the decoder are replaced by 2 x 2 transposed convolutions (Noh et al., 2015) with a stride of 2. Preliminary experiments have shown that this strategy can preserve small details better than standard max-pooling and up-sampling by bilinear interpolation used in (Ronneberger et al., 2015) and (Yang et al., 2019). In our experiments, we investigate whether under these circumstances it is still required to use skip connections. This is relevant because we expect the skip connections to be detrimental to DA if representation transfer is applied in the decoder (cf. Section 4.1). We apply dropout (Srivastava, 2014) after each layer with a rate of 0.1 for regularization. We found this to yield better results than L1 or L2 regularization of the network parameters.

3.2 Training

Training requires patches of size 640 x 640 pixels for which the orthophoto, the DSM and the reference label map are available. We apply data augmentation to make the classifier more robust with respect to rotations, radiometric changes, and varying building heights, as described in Section 5. In the training process, we minimize the classification loss of Yang et al. (2019), an extension of the focal loss for binary segmentation (Lin, 2017), by stochastic gradient descent with a mini-batch size of 1. The ADAM optimizer (Kingma & Ba, 2015) is used with a learning rate of 0.0001 and parameters $b_1 = 0.95$, $b_2 = 0.999$. Training is stopped as soon as the training error has not decreased for 1000 iterations.

4. DEEP ADVERSARIAL DOMAIN ADAPTATION

In this section, we consider data from two domains, as described in Section 2. We follow the strategy of ADDA (Tzeng et al., 2017), but adapt it to be applicable for pixel-wise classification. We start with a brief revision of ADDA before presenting our extensions that also allow the representation transfer to occur at different layers of the FCN. After that, we discuss the architecture of the discriminator network required for representation transfer before describing the adversarial training procedure.

4.1 Review of ADDA

ADDA was developed for CNN predicting a single class label for an image (Tzeng et al., 2017). Formally, the CNN is split into a feature extraction part that produces a mapping $M(\mathbf{x}_i)$ of an image $\mathbf{x}_i$ into some feature space and a classifier $Cl(M(\mathbf{x}_i))$ that predicts a class label C_i for $\mathbf{x}_i$, thus $C_i = Cl(M(\mathbf{x}_i))$. The classifier consists of the last (output) layer of the network, while M corresponds to the rest of the network. In the presence of two domains, there are two mappings ($M^S(\mathbf{x}^S)$, $M^T(\mathbf{x}^T)$) for the source and target domains, respectively, and two classifiers (Cl^S, Cl^T). The strategy of ADDA is to learn M^T such that it produces outputs $M^T(\mathbf{x}^T)$ for target samples $\mathbf{x}^T$ that have a similar distribution as the outputs of $M^S(\mathbf{x}^S)$ for source samples. Consequently, the source classifier can be applied to the target representations, thus $Cl^S = Cl^T$. First, Tzeng et al. (2017) train the source mapping and the source classifier using the labelled source samples by standard CNN training. After that, the parameters of M^S are kept constant. The parameters of M^T are initialized by those of M^S and adapted in adversarial training. In this context, a discriminator $D(M)$ is used as a kind of similarity measure for distributions. The discriminator takes a feature vector produced by mapping M and predicts whether it was generated from a source sample by M^S or from a target sample by M^T. D is trained to differentiate source from target samples as good as possible, while at the same time the parameters of M^T are adapted so that this task becomes as difficult as possible for D. After training, the class label C_i^T of a target sample $\mathbf{x}_i^T$ can be determined by $C_i^T = Cl^S(M^T(\mathbf{x}_i^T))$.

4.2 Proposed concept for domain adaptation

In our case, the output of the CNN is not a single class label, but a label map $\boldsymbol{c}$ having the same number of pixels as the input $\boldsymbol{x}$ (cf. Section 3). Directly applying the principles of Tzeng et al. (2017) to such a CNN implies that the representation to be adapted is the one delivered by the last layer before the classifier, consisting of one feature vector per pixel. The discriminator would also have to deliver a binary label map of the same dimensions, discriminating whether such a feature vector was generated from the source domain or the target domain. Such a strategy is followed by Zhang et al. (2018). We argue that this may not necessarily be the best option. In principle, the output of any layer of the network could serve as the intermediate representation to be adapted, and we expect the selection of an appropriate layer to have a heavy impact on the results. It might not make much sense to select the early feature maps, because we expect these features to be highly correlated with the input and, consequently, not abstract enough to be adaptable to the other domain. On the other hand, we expect layers near the output layer to be more correlated to the labels, which may be bad for adaptations in case the label distributions $P(C)$ are very different.

To gain flexibility for selecting the layer of the network at which transfer is to occur, we decompose the feature mapping of the network into two parts M_1 and M_2. We assume the mapping M_1 to be domain specific and to produce an intermediate representation $\mathbf{r}$ that will be adapted. This intermediate representation is $\mathbf{r}_i^S = M_1^S(\mathbf{x}_i^S)$ for a source sample and $\mathbf{r}_i^T = M_1^T(\mathbf{x}_i^T)$ for a target sample. The representation $\mathbf{r}$ is fed into the mapping M_2 of the network, whose output is classified by the classifier Cl. Note that M_1 and M_2 need not correspond to the encoder and decoder parts of the network described in Section 3; the output of an arbitrary intermediate layer can be selected for providing the representation to be adapted, while the remaining layers before the classifiers correspond to M_2. In the extreme case, we can select the layer to be adapted to be the last layer before the classifier. This is the strategy used in (Zhang et al., 2018) and can be accommodated by selecting M_2 to be an identical mapping. In our experiments, we will compare different variants for M_1 and M_2 (cf. layers A_1-A_4 in Figure 1) to find out whether DA is best carried out near the transition from the encoder to the decoder or at the end of the decoder. Figure 2 shows the concept of our DA method.

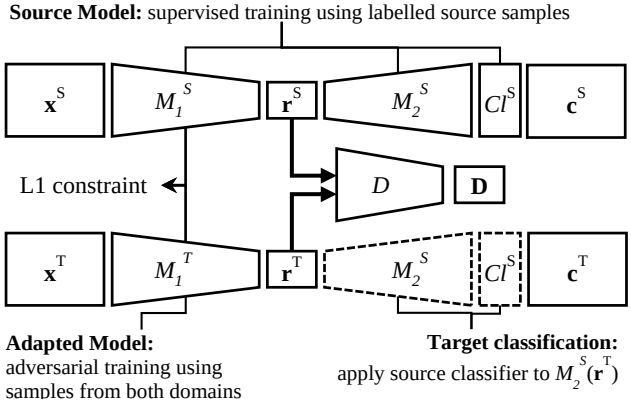

Figure 2. Concept of adversarial domain adaptation.

Like in ADDA, we start by training the source mappings M_1^S, M_2^S as well as the source classifier Cl^S on source domain training

samples as described in section 3.2, which results in a network predicting a label map $\mathbf{c}_i^S$ from a source sample (MSI + DSM) $\mathbf{x}_i^S$ according to $\mathbf{c}_i^S = Cl^S(M_2^S(M_1^S(\mathbf{x}_i^S)))$. Adversarial training using a discriminator D is applied to learn $M_1^T(\mathbf{x}^T)$ so that its output cannot be differentiated from $M_1^S(\mathbf{x}^S)$ by D. In this context, we apply an additional regularization to prevent a weight drift (cf. Section 4.3). Following the ADDA principles, we do not adapt the remaining parts of the network. Thus, after adversarial training of M_1^T, a target sample can be classified by applying M_2^S and Cl^S to the output of $M_1^T(\mathbf{x}^T)$. For a target sample $\mathbf{x}_i^T$, the label map $\mathbf{c}_i^T$ is predicted according to $\mathbf{c}_i^T = Cl^S(M_2^S(M_1^T(\mathbf{x}_i^T)))$.

4.3 Discriminator architecture and adversarial training

The discriminator $D(\mathbf{r})$ takes a feature map $\mathbf{r}$ generated either by M_1^T or by M_1^S and produces the probabilistic confidence map $\mathbf{D}$ of a binary classifier. Each pixel $d_{rc}(\mathbf{r}_{rc})$ of $\mathbf{D}$ at position (r, c) contains the posterior for the feature vector of the corresponding cell $\mathbf{r}_{rc}$ of $\mathbf{r}$ to have been generated from a source sample. The posterior for $\mathbf{r}_{rc}$ to have been generated from a target sample is $1-d_{rc}(\mathbf{r}_{rc})$. Thus, we propose to view each activation in a specific position in the feature map individually and convert it into one posterior. In this way, the discriminator has to learn local decisions based on the support window of $\mathbf{r}_{rc}$. We argue that this is a more difficult task than just taking one such decision based on the entire feature map $\mathbf{r}$, because in the latter case, the discriminator might just learn to differentiate different types of scenes (e.g. suburban from densely built-up). We expect this to mitigate the impact of different class distributions $P(C)$ on DA. The discriminator consists of four convolutional layers with a depth of 512 and leaky ReLU non-linearity with a slope of 0.2. We use zero-mean filters for all convolutions, which we observed to deliver more stable results than standard filters. In order to accomplish an individual classification of each vector $\mathbf{r}_{rc}$, we only use 1 x 1 convolutions. The final layer is another 1 x 1 convolution with depth of 1 and sigmoid activation to produce the probabilistic output. Note that this corresponds to applying a multilayer perceptron to each vector $\mathbf{r}_{rc}$ individually.

We follow the principles of adversarial training of (Goodfellow et al., 2014). The training consists of alternatingly updating the discriminator network by minimizing the discriminator loss

$$\mathcal{L}_D = -\sum_{r,c} \log\left(d_{rc}(\mathbf{r}_{rc}^S)\right) - \log\left(1 - d_{rc}(\mathbf{r}_{rc}^T)\right) \quad (2)$$

and updating M_1^T by minimizing the loss

$$\mathcal{L}_G = -\sum_{r,c} \log\left(d_{rc}(\mathbf{r}_{rc}^T)\right) + \lambda \cdot \frac{1}{n_M} \sum_{i=0}^{n_i} \left\| \boldsymbol{\theta}^S - \boldsymbol{\theta}^T \right\|_1, \quad (3)$$

In (3), the first term corresponds to the GAN loss recommended by Goodfellow et al. (2014) to prevent the discriminator from an early saturation. The second term, weighted by a parameter λ, is a regularization loss aiming at minimizing the mean L1 distance of the parameters of the source and target mappings. The parameters of M_1^S and M_1^T are denoted by $\boldsymbol{\theta}^S$ and $\boldsymbol{\theta}^T$, respectively, while n_M is the number of parameters of M_1 (identical for S and T). This regularization loss is designed to prevent a drift of the parameters of M_1^T and keep them close to those of the source network M_1^S. We choose the L1 distance because we observed more stable results in comparison to using the L2 distance as a similarity metric in preliminary experiments.

In the DA phase, the training is done by alternately updating the target segmentation network and the discriminator using stochastic gradient descent with a mini-batch size of 1. Again, the ADAM optimizer is used for both networks with a learning rate of 0.0001 and parameters $b_1 = 0.5$, $b_2 = 0.999$. We use a fixed number of training epochs (40), noting that it is difficult to define a stopping criterion without labelled target samples.

5. EXPERIMENTS

5.1 Datasets and Test setup

We use two datasets in our experiments. The first one is the Vaihingen dataset of the ISPRS labelling challenge, consisting of 33 patches of annotated multispectral and height data (Wegner et al., 2017). For each patch, a multispectral orthophoto consisting of three bands (NIR, red, green) and a DSM is provided, both at a ground sampling distance (GSD) of 9 cm. The average patch size is about 2000 x 2000 pixels. The reference contains the six classes (*impervious surface*, *building*, *low vegetation*, *tree*, *car clutter*). This dataset is only used in Section 5.2 to compare two variants of our FCN model (Section 3) to other approaches. Following the protocol of the benchmark, we use 16 patches for training and 17 for evaluation. To be consistent with the evaluation on the benchmark website, we report the overall accuracies (OA) and F1 scores determined without considering pixels near object boundaries in the reference, i.e., based on the eroded reference provided by the benchmark organizers.

For the evaluation of our DA approach, we use the 3City dataset provided by (Vogt et al., 2018). It consists of aerial images of three German cities, referred to as C1, C2, and C3. For each city, it consists of a grid of 3 x 3 adjacent tiles with a total extent of about 10,000 x 10,000 pixels at a GSD of 20 cm. For each tile, a four-channel multispectral orthophoto (NIR, red, green, blue) and a DSM is provided. To make the input data consistent with the Vaihingen dataset, we do not use the blue channel in the experiments. All pixels of the dataset were manually labelled as belonging to one of the three classes *tree*, *building*, *ground*. In the context of DA, we consider the three cities in the dataset as three different domains. In all experiments, we use the outer ring of eight tiles of each city for training and DA and the central tiles for the evaluation. We report the OA achieved in the central tiles in all experiments. As we found it difficult to define a stopping criterion for adversarial training in DA (cf. Section 4.3), in all experiments involving DA we report the average OA evaluated for the last 15 (of 40) iterations in adversarial training as well as the corresponding standard deviations. We report on two sets of experiments. In Section 5.3, we compare several variants of DA that differ by FCN architecture, the layer of the FCN chosen for adapting the domains (cf. Section 4.2) and by the architecture of the discriminator (cf. Section 4.3). In this context, we also compare the results of DA to a naive approach just applying the classifier trained on source data to the target domain without adaptation. The difference in OA between this naive approach and a DA-based approach is a measure for positive transfer, i.e. the degree to which DA improves the classification accuracy. We report results for all possible pairs of source and target domains involving two cities. This will indicate the performance for different degrees of similarity of the domains: while the domains C1 and C3 are similar, C2 is rather different, class *building* covering about twice the area it covers in the other test sites.

We apply a band-wise normalization to each domain in each dataset. The normalization for a band b is done by subtracting the mean μ_b and dividing by the standard deviation σ_b. As the DSMs contain metric values that have the same scale, the heights are normalized with a fixed standard deviation $\sigma_{DSM} = 5\,m$. Data

augmentation was applied by randomly cropping patches of 640x640 pixels from the tiles used for training. These patches were also randomly flipped along one axis and rotated by k x 90° with $k \in \{0, 1, 2, 3\}$ being drawn randomly. Additional random rotations did not improve the results. To make the model more robust against different data distributions, we applied a band-specific random scale $s \sim N(1.0, 0.1)$ and a random shift $b \sim N(0.0, 0.1)$ to each sample. The weight of the regularization loss for DA was set to $\lambda = 2.0$ in all experiments.

For testing we apply a sliding window evaluation with a stride of 200 pixels. Starting from the upper left corner of the test patch, a window of 640 x 640 pixels is fed to the network to produce pixel-wise class scores. Due to using a stride that is smaller than the window classified by the FCN, most pixels of a test patch will be classified multiple times. In order to obtain a unique prediction, we sum the class scores for every pixel and assign the pixel to the class having the maximum combined score.

5.2 Evaluation of the FCN model for classification

In this section, we compare two variants of our FCN network to other architectures using the Vaihingen data: model **VSC** has skip connections while model **VB** has not. The evaluation results based on the eroded reference are given in Table 3.

Model	Overall Accuracy [%]	F1 Score [%]				
		Imp. sur.	Build.	Low veg.	Tree	Car
VSC	89.6	92.3	94.5	82.5	88.4	77.7
VB	89.3	91.9	94.4	82.1	88.1	74.9

Table 3. Results for Vaihingen benchmark.

The results show that our model without skip connections (**VB**) performs only 0.3% worse in terms of OA than **VSC** (with skip connections). This indicates that the performance does not depend heavily on these connections. Nevertheless, besides leading to a slightly better accuracy, they decrease the training time of the model significantly. We observe that **VB** has difficulties in reconstructing precise details like building corners than **VSC**, but its predictions are less noisy than those from **VSC**. Having followed the protocol of the ISPRS labelling challenge, we can compare our results to those achieved by other methods. Currently, the benchmark website (Wegner et al., 2017) lists the best OA as 91.6%. Thus, we perform slightly worse (2%) than the best approach.

5.3 Evaluation of DA

5.3.1 Classifying target images using a source classifier: This experiment serves as a baseline for all DA variants. We train the models **VSC** and **VB** using one domain and apply these classifiers to the other domains without adaptation. This is repeated three times, each time using another domain for training. Table 4 shows the resulting OA for each model, training domain (TR) and evaluation domain (EV). The accuracies achieved when training and testing on the same domain are printed in bold font. To summarize the performance of the models on the same and on another domain, the corresponding mean OA is provided.

Model		**EV**, OA [%]			Mean OA [%]	
	TR	C1	C2	C3	Same D.	Other D.
VSC	C1	**92.4**	77.7	86.2	91.7	83.1
	C2	86.3	**91.1**	83.8		
	C3	90.3	74.5	**91.5**		
VB	C1	**91.7**	83.2	86.9	91.1	84.6
	C2	86.3	**91.1**	83.1		
	C3	88.1	80.2	**90.6**		

Table 4. OA of applying a classifier to other domains DA.

Again, **VSC** achieves slightly better results when trained and evaluated on the same domain (bold numbers in Table 4). However, when training and evaluation domains are different, **VB** achieves a higher mean accuracy, which indicates that **VSC** has a stronger tendency to overfit to the training domain than **VB**. In any case, we note a considerable drop in OA when applying a classifier to another domain without adaptation. For **VSC**, it is in the order of 5% in most cases, but it can reach about 10% when the data are dissimilar; cf. the results for classifiers trained using datasets C1 or C3 when applied to C2 (column C2 in Table 4).

5.3.2 Comparing different DA variants: Here we test four variants of DA differing by the layer of the network at which the adaptation occurs, i.e. using different definitions of the mappings M_1 and M_2 (cf. Section 4.2). In all cases, the feature mappings M_1^S, M_2^S and the classifier Cl^S for the source domain are those already determined in the training described in Section 5.3.1. For all source domains, DA is applied using the two other cities as target domains. The layers at which the adaptation occurs are marked as A_1-A_4 in Figure 1. In all cases, M_1 consists of all layers before and including A_i, while the remaining layers constitute M_2.

In variant **V1**, representing an early matching of representations, the adaptation occurs at layer A_1 of the colour branch of the encoder (cf. Figure 1). In variant **V2**, the adaptation is based on the results of the last convolutional layer of the colour branch of the encoder (layer A_2 in Figure 1). It corresponds an adaptation of the representations generated by the encoder before fusion. In contrast, variant **V3** adapts the encoder output after fusing the results of the DSM and colour branches (layer A_3 in Figure 1). Finally, variant **V4** corresponds to late matching, with adaptation occurring in the last layer of the decoder (layer A_4 in Figure 1).

The results achieved for the four variants using the models **VB** and **VSC** are shown in Table 5. In these tables, numbers in bold font mark cases of a positive transfer (i.e., the OA for the specific pair of source and target domains is better than the one reported for the baseline in Table 4). Numbers in bold and italic font show a neutral DA result, while numbers in standard font indicate a negative transfer. SD and TD are the source and target domains, respectively. There are no numbers on the main diagonals, because in this case SD and TD are identical (Table 4).

Analysing Table 5, it is evident that variant **V1** is the only one achieving a positive transfer in all combinations of source and target domains and for both models. Although the improvement of the OA due to DA is larger for **VSC**, final mean OA is slightly higher for model **VB**. On average, we can gain 2.3% and 1.0% in OA for **VSC** and **VB**, respectively. In both cases, the largest improvement (4-5%) is observed for the difficult case (column C2) using the model **VSC**. Variant **V2** can achieve a positive transfer in 9 of 12 cases, with a slightly smaller improvement compared to **V1**. There is still a small average improvement of OA. Variant **V3** results in a negative transfer in the majority of cases (9 out of 12). The reasons for this are unclear and require further investigations. Although we consider this setup as not suitable for a stable DA, we still want to point out that the adaptation still worked slightly better for model **VB**, where at least 2 of 6 adaptations were successful and the final mean OA is higher by 1.9% than for **VSC**. Finally, our results for variant **V4** show that, not unexpectedly, the last layer is not suited well for adaptation, achieving a considerable negative transfer. Interestingly, model **VB** results in a positive transfer from C1 to C3 and vice versa, which might be due to the fact that the label distributions of these domains are similar. We consider variant **V1** with model **VB** as the best method, yielding a positive transfer in all tested cases.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-IV-2-W7-197-2019 | © Authors 2019. CC BY 4.0 License.

Model		**TD**, OA after adaptation [%] mean and std. dev. of epochs 25-40			Mean OA [%]	Variant
	SD	C1	C2	C3		
VSC	C1	---	**81.2 ± 0.003**	**86.2 ± 0.005**	**85.4**	**V1** (early matching)
	C2	***86.3 ± 0.009***	---	**85.4 ± 0.006**		
	C3	**91.0 ± 0.002**	**82.1 ± 0.011**	---		
VB	C1	---	**83.3 ± 0.005**	**87.0 ± 0.005**	**85.6**	
	C2	**87.1 ± 0.002**	---	**84.7 ± 0.003**		
	C3	**88.7 ± 0.001**	**82.8 ± 0.002**	---		
VSC	C1	---	**80.5 ± 0.006**	85.3 ± 0.016	**84.2**	**V2** (before fusion)
	C2	***86.3 ± 0.005***	---	**85.2 ± 0.008**		
	C3	**90.8 ± 0.004**	**76.8 ± 0.009**	---		
VB	C1	---	81.9 ± 0.022	85.6 ± 0.002	**85.0**	
	C2	**87.4 ± 0.002**	---	**84.1 ± 0.005**		
	C3	**88.7 ± 0.003**	**82.0 ± 0.008**	---		
VSC	C1	---	77.1 ± 0.008	85.1 ± 0.011	81.5	**V3** (after fusion)
	C2	81.9 ± 0.053	---	76.3 ± 0.140		
	C3	88.1 ± 0.014	**80.6 ± 0.026**	---		
VB	C1	---	**84.6 ± 0.019**	86.1 ± 0.005	83.4	
	C2	84.0 ± 0.050	---	77.2 ± 0.062		
	C3	**89.7 ± 0.008**	78.8 ± 0.011	---		
VSC	C1	---	**78.9 ± 0.002**	**86.4 ± 0.001**	74.7	**V4** (late matching)
	C2	74.3 ± 0.022	---	47.9 ± 0.119		
	C3	86.7 ± 0.004	74.1 ± 0.004	---		
VB	C1	---	83.0 ± 0.002	**87.0 ± 0.001**	80.9	
	C2	83.4 ± 0.020	---	73.8 ± 0.036		
	C3	**88.2 ± 0.001**	70.1 ± 0.022	---		

Table 5. Mean OA and standard deviation on the target domain after DA by variants **V1-V4**.

5.3.3 Comparing different discriminator architectures: In this section, we report on additional experiments highlighting some properties of the proposed discriminator architecture, again using the models **VB** and **VSC**. In the first experiment, we want to validate the positive effects of using zero-mean convolutions in the discriminator. This experiment is based on the best DA variant according to Section 5.3.2, variant **V1** (early matching). We replace all zero-mean convolutions in the discriminator (cf. Section 4.3) with regular convolutions. The results for models **VB** and **VSC** are shown in Table 6. We can achieve a positive transfer only in 8 of 12 cases and the mean accuracies for both models are lower than in variant **V1**. We take this as an indication for the importance of using the zero-mean convolutions.

Model		**TD**, OA after adaptation [%] mean and std. dev. of epochs 25-40			Mean OA [%]
	SD	C1	C2	C3	
VSC	C1	---	**79.5 ± 0.019**	85.8 ± 0.004	83.9
	C2	85.8 ± 0.017	---	**85.2 ± 0.005**	
	C3	**90.5 ± 0.004**	**76.5 ± 0.011**	---	
VB	C1	---	74.8 ± 0.105	85.7 ± 0.009	83.7
	C2	**87.3 ± 0.002**	---	**84.1 ± 0.006**	
	C3	**88.5 ± 0.006**	**81.8 ± 0.022**	---	

Table 6. Mean OA and standard deviation on the target domain after DA by variant **V1** w/o zero-mean convolutions.

In a last set of experiments, we adapt the discriminator from (Zhang et al., 2018) which uses four dilated 3 x 3 convolutions with dilation rates 1,2,3 and 4 and 128 filters each. The resulting activations are concatenated and a 1 x 1 convolution with depth 1 and sigmoid activation is applied. As proposed in (Zhang et al., 2018) we apply this discriminator to activation map A_4 (variant **V4b**). Table 7 shows the results for models **VB** and **VSC**. The results using this discriminator are even worse than those for **V4**. We analyse the transfer from C2 to C3 to show possible reasons why this approach does not work here. For that purpose, we compare the results of **V4b** to those of **V1**, where the transfer was successful. As stated in Section 4.2, the last feature maps are highly correlated with the labels, which are very different in C2 and C3: while the percentage of *tree* pixels is similar, in C2, 36.7% of the pixels correspond to *building* and 42.2% to *ground*. The corresponding numbers for C3 are 18.2% and 67.0%, resp. In this setting of DA, the target network has to adjust to the label distribution in the source domain in order to fool the discriminator. Consequently, after DA in variant **V4b**, the percentages of *building* and *ground* pixels in the target domain (C3) are much closer to those of the source domain (29.7% and 56.3%, resp.). This contrasts with 20.6% and 66.3%, resp., in variant **V1**, which is much closer to the true label distribution in the target domain and, thus, allows for positive transfer.

Model		**TD**, OA after adaptation [%] mean and std. dev. of epochs 25-40			Mean OA [%]
	SD	C1	C2	C3	
VSC	C1	---	**78.7 ± 0.005**	63.9 ± 0.050	71.8
	C2	67.7 ± 0.059	---	58.5 ± 0.008	
	C3	88.8 ± 0.007	73.2 ± 0.016	---	
VB	C1	---	72.3 ± 0.021	**87.2 ± 0.005**	76.1
	C2	80.8 ± 0.020	---	67.1 ± 0.064	
	C3	81.1 ± 0.008	68.4 ± 0.019	---	

Table 7. Mean OA and standard deviation on the target domain after DA by variant **V4b**.

6. CONCLUSION

In this paper, first we presented a FCN that is invariant to shifts in the height model by design and showed that neglecting the skip-connections in our model only leads to only a small drop of classification quality, while in general achieving results close to the state of the art in a benchmark. Our main contribution is the transfer of a method for deep DA to the task of pixel-wise classification of aerial imagery and derived data. We tested different variants of the representation transfer and found that DA performed best when applied to the middle layer of the colour branch of the encoder network. In this variant, we could achieve a positive transfer for all combinations of source and target domains. We could also show that neglecting the skip-connections results in a better OA after DA and that the success of representation transfer based on the last feature map is heavily influenced by the label distribution, resulting in poor DA performance if the true label distributions differ.

Future research should analyse whether the inclusion of multiple domain discriminators can improve the results, in particular for difficult cases (large differences between domains). The reasons why an adaptation of the height data is detrimental to the results also need to be analysed. Furthermore, we have not fine-tuned our hyper-parameters using a validation dataset; this could have a positive impact on the results. Finally, additional tests involving larger datasets and more domains are required to analyse the behaviour of our DA approach in more detail.

ACKNOWLEDGEMENTS

This work was partially funded by the Federal Ministry of Education and Research, Germany (Bundesministerium für Bildung und Forschung, Förderkennzeichen 01IS17076). The Vaihingen dataset was provided by the German Society for Photogrammetry, Remote Sensing and Geoinformation (DGPF) (Cramer, 2010): http://www.ifp.uni-stuttgart.de/dgpf/DKEP-Allg.html. The 3City dataset is an extract from the geospatial data of the Lower Saxony survey and cadastre administration, (c) 2013 (LGLN), the reference was provided by (Vogt et al., 2018).

REFERENCES

Yang, C., Rottensteiner, F, Heipke, C., 2019. Towards better classification of land cover and land use based on convolutional neural networks. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. XLII-2/W13.

Audeberg, N., Saux, B. L., Lefevre, S., 2018. Beyond RGB: Very high resolution urban remote sensing with multimodal deep networks. *ISPRS Journal of Photogrammetry and Remote Sensing*,140, 20-32.

Bashmal, L., Bazi, Y., AlHichri, H., AlRahhal, M. M., Ammour, N., Alajlan, N., 2018. Siamese-GAN: Learning invariant representations for aerial vehicle image categorization. *Remote Sensing*, 10(2), 351.

Bruzzone, L., Chi, M., Marconcini, M., 2006: A novel transductive SVM for semisupervised classification of remote-sensing images. *IEEE Transactions on Geoscience & Remote Sens.*, 44(11), 3363-3373.

Cramer, M., 2010. The DGPF test on digital aerial camera evaluation – overview and test design. *Photogrammetrie Fernerkundung Geoinformation*, 2(2010), 73–82.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Lorachelle, H., Laviolette, F., Lempitsky, V., 2016: Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1), 2096-2030.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets. *Advances in Neural Information Processing Systems (NIPS)*, 2672-2680.

Hoffmann J. Wang, D., Yu, F., Darrell, T., 2016. FCNs in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv:1612.02649.*

Hoffmann, J., Tzeng, E., Park, T., Zhu, J., Isola, P., Saenko, K., Efros, A., Darrell, T., 2017. CyCADA: Cycle-consistent adversarial domain adaptation. *Proceedings of the 35th International Conference on Machine Learning*, 1989-1998.

Huang, H., Huang, Q., Krähenbühl, P., 2018. Domain transfer through deep activation matching. *European Conference on Computer Vision (ECCV).* 590-605.

Kingma, D. P., Ba, J. L., 2015. Adam: A method for stochastic optimization. *International Conference on Learning Representations.*

Krizhevsky, A., Sutskever, I., Hinton, G. E., 2012. ImageNet classification with deep convolutional neural networks. *Advances in Neural Information ProcessingSystems(NIPS)*, 1097-1105.

Lin, T., Goyal, P., Girshick, R., He, K., Dollár, P., 2017. Focal loss for dense object detection. *Proceedings of the IEEE international conference on computer vision (ICCV)*, 2980-2988.

Long, J., Shelhamer, E., Darrell, T., 2015a. Fully convolutional networks for Semantic Segmentation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3431-3440.

Long, M., Cao, Y., Wang, J., Jordan, M. I., 2015b: Learning transferable features with deep adaptation networks. *Proceedings 32nd International Conference on Machine Learning*,37,97-105.

Matasci, G., Volpi, M., Kanevski, M., Bruzzone, L., Tuia, D., 2015. Semisupervised transfer component analysis for domain adaptation in remote sensing image classification. *IEEE Transactions on Geoscience & Remote Sens.*, 53(7), 3550-3564.

Noh, H., Hong, S., Han, B., 2015. Learning Deconvolution Network for Semantic Segmentation. *IEEE International Conference on Computer Vision (ICCV)*, 1520-1528.

Pan, S. J., Yang, Q., 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10), 1345-1359.

Postadjian, T., Le Bris, A., Sahbi, H., Mallet, C., 2018. Domain adaptation for large scale classification of very high resolution satellite images with deep convolutional neural networks. *International Geoscience and Remote Sensing Symposium*, 3623-3626.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 9351,234-241.

Schlüter, J., Lehner, B., 2018. Zero-mean convolutions for level-invariant singing voice detection. *19th International Society for Music Information Retrieval Conference*, 23-27

Springenberg, J., Dosovitskiy, A., Brox, T., Riedmiller, M., 2015. Striving for simplicity: The all convolutional net. *International Conference on Learning Representations workshops.*

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15, 1929-1958.

Tuia, D., Persello, C., Bruzzone, L., 2016. Domain adaptation for the classification of remote sensing data: an overview of recent advances. *IEEE Geoscience and Remote Sensing Magazine*, 4(2), 41-57.

Tzeng, E., Hoffman, J., Saenko, K., Darrell, T., 2017. Adversarial Discriminative Domain Adaptation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7167-7176.

Ulyanov, D., Vedaldi, A., Lempitsky, V., 2016. Instance Normalization: The Missing Ingredient for Fast Stylization. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Vogt, K., Paul, A., Ostermann, J., Rottensteiner, F., Heipke, C., 2018. Unsupervised source selection for domain adaptation. *Photogrammetric Engineering & Remote Sens. 84(5)*, 249–261.

Wang, M., Deng, W., 2018. Deep visual domain adaptation: a survey. *Neurocomputing*, 312, 135-153.

Wegner et al., 2017. The ISPRS 2D semantic labelling contest. http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html (accessed 8/4/2019).

Yosinski, J., Clune, J., Bengio, Y., Lipson, H., 2014. How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems 27 (NIPS'14)*, 2, 3320-3328.

Zhang, Y., Qiu, Z., Yao, T., Liu, D., Mei, T., 2018. Fully Convolutional Adaptation Networks for Semantic Segmentation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6810-6818.

Zhu, X.X., Tuia, D., Mou, L., Xia, G.-S., Zhang, L., Xu, F., Fraundorfer, F., 2017. Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine*, 5(4), 8-36.