remote sensing

MDPI

Article

Siamese-GAN: Learning Invariant Representations for Aerial Vehicle Image Categorization

Laila Bashmal [1], Yakoub Bazi [1,*], Haikel AlHichri [1], Mohamad M. AlRahhal [2], Nassim Ammour [1] and Naif Alajlan [1]

[1] Computer Engineering Department, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia; lailabashmal@outlook.com (L.B.); hhichri@ksu.edu.sa (H.A.); nammour@ksu.edu.sa (N.A.); najlan@ksu.edu.sa (N.A.)

[2] Information Science Department, College of Applied Computer Science, King Saud University, Riyadh 11543, Saudi Arabia; mmalrahhal@ksu.edu.sa

* Correspondence: ybazi@ksu.edu.sa; Tel.: +966-1014696297

Received: 17 January 2018; Accepted: 22 February 2018; Published: 24 February 2018

Abstract: In this paper, we present a new algorithm for cross-domain classification in aerial vehicle images based on generative adversarial networks (GANs). The proposed method, called Siamese-GAN, learns invariant feature representations for both labeled and unlabeled images coming from two different domains. To this end, we train in an adversarial manner a Siamese encoder–decoder architecture coupled with a discriminator network. The encoder–decoder network has the task of matching the distributions of both domains in a shared space regularized by the reconstruction ability, while the discriminator seeks to distinguish between them. After this phase, we feed the resulting encoded labeled and unlabeled features to another network composed of two fully-connected layers for training and classification, respectively. Experiments on several cross-domain datasets composed of extremely high resolution (EHR) images acquired by manned/unmanned aerial vehicles (MAV/UAV) over the cities of Vaihingen, Toronto, Potsdam, and Trento are reported and discussed.

Keywords: manned/unmanned aerial vehicles (MAV/UAV); extremely high resolution (EHR) images; distribution mismatch; generative adversarial networks (GANs); Siamese encoder–decoder

1. Introduction

The rapid development of remote sensing imaging technologies has allowed us to obtain heterogonous images of the Earth's surface with high spatial and temporal resolution. The rich and complex structural information conveyed by these types of imagery has opened the door for the development of advanced methodologies for processing and analysis. Among these methodologies, scene-level classification has attracted much research from the remote sensing community in recent years. The task of scene classification is to automatically assign an image to a set of predefined semantic categories. This task is particularly challenging as it requires the definition of high-level features for representing the image content to assign it to a specific category.

Among the proposed solutions, one can find approaches based on handcrafted features, which refer to image attributes that are manually designed such as scale-invariant feature transform (SIFT) [1], local binary pattern (LBP) [2], and bag of visual words (BOVW) model. In the BOWV model, each image is represented as a histogram of visual word frequencies, and then a visual word codebook is generated by partitioning an image into dense regions and applying k-means clustering. The conventional (BoW) was mainly designed for document classification. Therefore, when it is applied to images it describes the local information using the local descriptors but ignores the spatial information in the image. For such purposes, improved models have been proposed to utilize spatial information of images. For instance, a pyramid-of-spatial-relations (PSR) model was developed in [3] to capture both the absolute and relative spatial relationships of local features leading to rotation invariance representation for land use scene images. Zhu et al. [4] improved the (BOVW) model by combining the local and global features of high spatial resolution (HSR) images. They considered the shape-based invariant texture index (SITI) as the global texture feature, the mean and standard deviation values as the local spectral feature, and the (SIFT) feature as the structural feature. Another work [5] proposed a local–global fusion strategy, which used BoVW and spatial pyramid matching (SPM) to generate local features, and multiscale completed (CLBP) to extract global features. In [6], the authors proposed a concentric circle-based spatial- and rotation-invariant representation strategy to describe the spatial information of visual words and a concentric circle-structured multi-scale (BoVW) method using multiple features. This model incorporates rotation-invariant spatial layout information into the original BOVW model to enhance scene classification results.

Feature learning-based approaches provide an alternative way to automatically learn discriminative feature representation from images. There have been many studies that attempt to address the scene classification problem by using feature learning techniques. In [7], Cheriyadat proposed unsupervised feature learning strategy for aerial scene classification that uses sparse coding to generate a new image representation from low-level features. In [8], Mekhalfi et al. presented a framework that represents an image through an ensemble of compressive sensing and a multi-feature framework. They considered different types of features, namely histogram of oriented gradients, co-occurrence of adjacent local binary patterns and gradient local autocorrelations. The authors of [9] proposed a multi-feature fusion technique that describes images by three feature vectors: spectral, textural, and SIFT vectors, which are separately extracted and quantized by *K-means* clustering. The latent semantic allocations of the three features are captured separately by probabilistic topic model and then fused into the final semantic allocation vector. In [10], Cheng et al. introduced a classification method based on pre-trained part detectors. They used one-layer sparse coding to discover midlevel features from the partlets-based low-level features. In [11], the authors proposed a two-layer framework for unsupervised feature learning. The framework can extract both simple and complex structural features of the image via a hierarchical convolutional scheme. *K-means* clustering is used to train the features extractor and then *K-nearest neighbors* is performed for classification. Hu et al. [12] proposed unsupervised feature learning algorithm, which learns on the low-level features via *K-means* clustering. The feature representation of the image is generated by building a (BOW) model of the encoded low-level features. Finally, in [13], the authors proposed a Dirichlet-derived multiple topic model to fuse four types of heterogeneous features including global, local, continuous, and discrete features.

Recently, deep learning methods have been shown to be more efficient than traditional methods in many applications such as audio recognition [14] face recognition [15] medical image analysis [16] and image classification [17]. Deep learning methods are based on multiple processing layers used to learn a good feature representation automatically from the input data. Different from shallow architectures, features in deep learning are learned in a hierarchical manner [18]. There are several variants of deep learning architecture, e.g., deep belief networks (DBNs) [19] stacked auto-encoders (SAEs) [20] and convolutional neural networks (CNNs) [21].

Deep networks can be designed and trained from scratch for a specific problem domain. For example, Luus et al. [22] proposed a multiscale input strategy for supervised multispectral land use classification. They proved that single deep CNN can be trained with multiscale views to obtain improved classification accuracy compared to using multiple views at one scale only. In [23], the authors proposed a feature selection method based on (DBN), the network is used to achieve feature abstraction by minimizing the feature reconstruction error, where features with relatively small reconstruction errors were taken as the discriminative features. Wu et al. [24] developed a model that stacks multicolumn autoencoders and Fisher vector pooling layer to learn abstract hierarchical semantic features. Zhang et al. [25] proposed a gradient-boosting random convolutional network framework that can effectively classify aerial images by combining many deep neural networks.

In some applications, including remote sensing, it is not feasible to train a new neural network from scratch, as this usually requires a considerable amount of labeled data and high computational costs. One possible solution is to use existing pre-trained networks such as GoogLeNet [26], AlexNet [27], or CaffeNet [28], and perform fine-tuning of its parameters using the data of interest. Several studies have used this technique to improve the network training process. Scott et al. [29] investigated the use of deep CNN for the classification of high-resolution remote sensing imagery. They developed two techniques based on data augmentation and transfer learning by fine-tuning from pre-trained models, namely CaffeNet, GoogLeNet, and ResNet. Another work [30] evaluated and analyzed three strategies using CNN for scene classification, including fully-trained CNN, fine-tuned CNN, and pre-trained CNN used as feature extractors. The results showed that fine-tuning tends to be the best-performing strategy. In [31], Marmanis et al. proposed a two-stage framework for earth observation classification. In the first stage, an initial set of representations is extracted by using a pre-trained CNN, namely ImageNet. Then, the obtained representations are fed to a supervised CNN for further learning. Hu et al. [32] proposed two scenarios for generating image representations. In the first scenario, the activation vectors are extracted directly from the fully connected layers and considered as global features. In the second scenario, dense features are extracted from the last convolutional layer and then encoded into a global feature. Then the features are fed into a support vector machine (SVM) classifier to obtain the class label. In [33], the authors used pre-trained (CNN) to generate an initial feature representation of the images. The output of the last fully connected layer is fed into a sparse autoencoder for learning a new representation. After this stage, two different scenarios are proposed for the classification system. Adding a softmax layer on the top of the encoding layer and fine-tune the resulting network, or train an autoencoder for each class and classify the test image based on the reconstruction error. In another work [34], used features extracted from CNNs pre-trained on ImageNet. They combined two types of features: The high-level features extracted from the last fully connected layer, and the low and mid-level features extracted from the intermediate convolutional layers. Weng et al. [35] proposed a framework that combines pre-trained CNNs and extreme learning machine. The CNN's fully connected layers are removed to make the rest parts of the network work as features extractor, while the extreme learning machine is used as a classifier. Chaib et al. [36] used VGG-Net model to extract features from VHR images. They used the outputs of the first and second fully connected layer of the network and combined them using discriminant correlation analysis to construct the final representation of the image scene.

From the above analysis, it appears that most of these methods were designed for a single domain classification task (assuming the training and testing images are from the same domain). Figure 1 shows a typical situation in the case of UAV platform acquiring extremely high resolution images (EHR) over a specific area. However, in many real-world applications, the training images used to learn a model may have different distributions from the images used for testing. This problem arises when dealing with data acquired over different locations of the Earth's surface and with different platforms, as shown in Figure 2. We recall that this aspect is not obvious in the currently available scene datasets as the training and testing data are generated randomly during evaluation. To highlight this undesirable effect, the authors of [37] have shown that the methods based on pre-trained CNNs may produce low accuracies when benchmarked with cross-domain datasets. As a remedial action, they have proposed compensating for the distribution mismatch by adding additional regularization terms to the objective function of the neural network besides the standard cross-entropy loss.

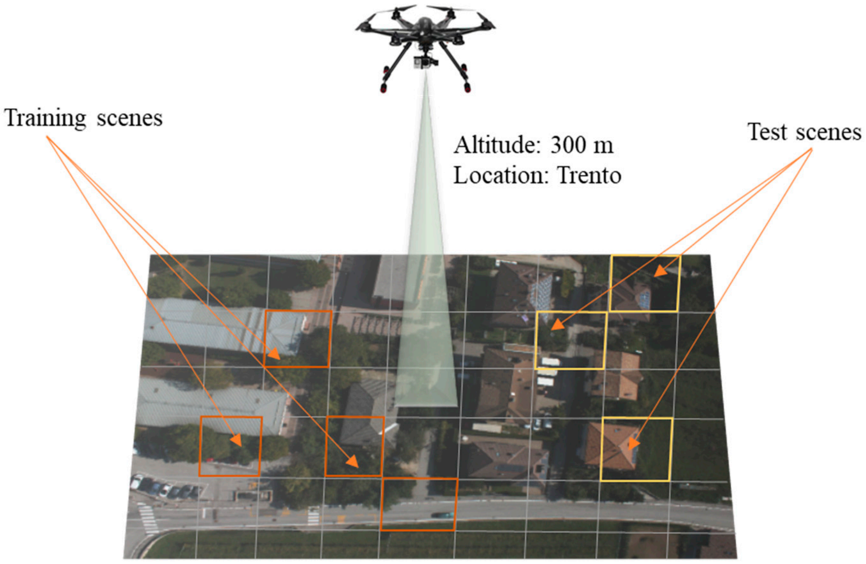


Figure 1. Standard supervised classification: training and test scenes are extracted from the same domain.

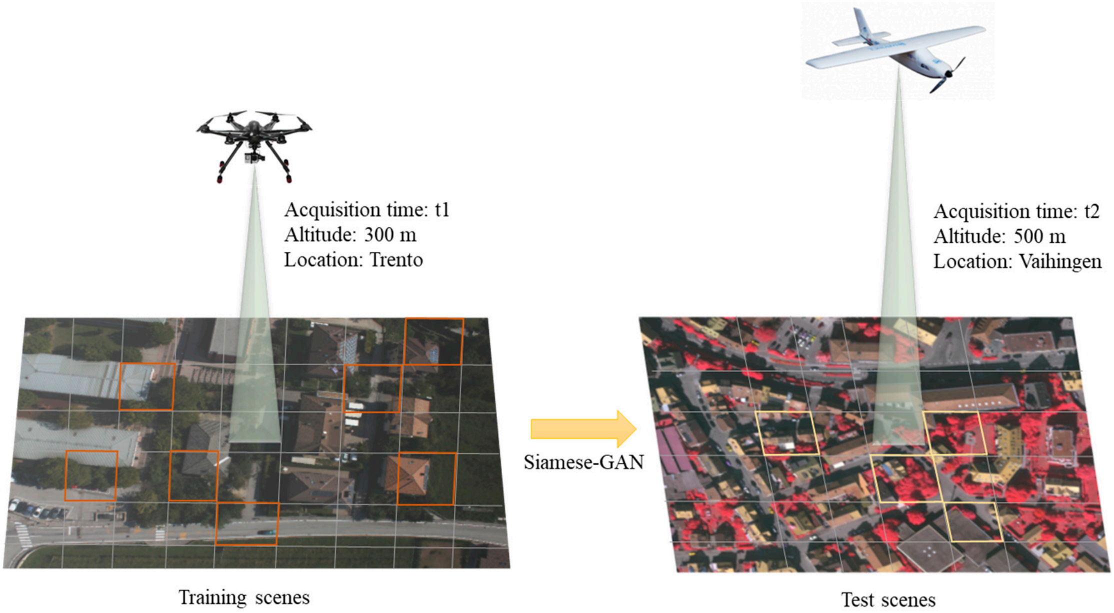


Figure 2. Cross-domain classification: use training samples from a previous domain to classify data coming from a new domain.

In this work, we propose a new domain adaptation approach to automatically handle such scenarios (Figure 2). Our objective is to learn invariant high-level feature representations for both training and testing data coming from two different domains referred here for convenience as labeled source and unlabeled target data. The method, termed Siamese-GAN, trains jointly in an adversarial manner a Siamese encoder–decoder network coupled with another network acting as a discriminator. The encoder-decoder network has the task to match the distributions of both domains in a shared space regularized by the reconstruction ability, while the discriminator seeks to distinguish between them. At the end of the optimization process, we feed the resulting encoded labeled source and unlabeled target features into an additional network for training and classification, respectively.

The major contribution of this work can be summarized as follows: (1) Introduce GANs as promising solution for the analysis of remote sensing data. (2) Overcome the data-shift problem for cross-domain classification by proposing an efficient method named Siamese-GAN. (4) Validate the method on several cross-domain datasets acquired over different locations of the earth surface and with different MAV/UAV platforms. (4) Present a comparative study against some related methods proposed in the literature of remote sensing and computer vision.

The paper is organized as follows. Section 2 reviews GANs. Section 3 describes the proposed Siamese-GAN method. Section 4 presents the results obtained for several benchmark cross-domain datasets. Section 5 analyzes the sensitivity of the method and presents comparisons with state-of-the-art methods. Finally, Section 6 concludes the paper.

2. Generative Adversarial Networks (GANs)

GANs have emerged as a novel approach for training deep generative models. The original GAN that was mainly proposed for image generation consists of two neural networks: the generator G and the discriminator D. The networks are trained in opposition to one another through a two-player minimax game. The generator network learns to create fake data that should come from the same distribution as the real data, while the discriminator network attempts to differentiate between the real and the fake data created by the generator. During each training cycle, the generator takes a random noise vector as an input and creates a synthetic image, the discriminator is presented with a real or generated image and tries to classify it as either "real" or "fake". Ideally, the two networks compete during the training process until the Nash equilibrium is reached. The GANs' objective function is given by:

$$\min_{G}\max_{D} V(D,G) = \mathbb{E}_{\boldsymbol{X}\backsim p_{data}\ (\boldsymbol{X})}[\log D(X)] + \mathbb{E}_{\boldsymbol{z}\backsim\ p_{z}(\boldsymbol{z})}[\log(1-D(G(z)))], \tag{1}$$

where X represents the real image from the true data distribution p_{data}, z represents the noise vector sampled from distribution p_z, and $G(z)$ represents the generated image. The generator G is learned by maximizing $D(G(z))$, while D is trained by minimizing $D(G(z))$.

Since the appearance of GANs in 2014, many extensions have been proposed to its architecture. For instance, Deep Convolutional GANs (DCGANs) [38] were designed to allow the network to generate data with similar internal structure as training data, improving the quality of the generated images, and Conditional GANs [39] add an additional conditioning variable to both the generator and the discriminator. Based on the previous architectures the concept of GANs has been adopted to solve many computer visions related tasks such as image generation [40,41], image super-resolution [42], unsupervised learning [43], semi-supervised learning [44], and image painting and colorization [45,46].

In the context of domain adaptation, some works have recently been introduced to the literature of computer vision. For instance, Ganin et al. [47] presented a domain-adversarial neural network method, which combines a deep feature extractor module with two classifiers for class-label and domain prediction, respectively. The network is trained by minimizing the label prediction loss for source data, and the domain classification loss for both source and target data via a gradient reversal layer. Liu and Tuzel [48] introduced an architecture that couples two or more GANs, each corresponding to one image domain. The two generators share the weights of the first layers that decode high-level features to learn the joint distribution of the images in the two domains, while the discriminators share the weights of the last layers. The authors of [49] proposed an architecture based on a CNN that is first trained with labeled source images. Then train in an adversarial manner a generator and a discriminator on source and target data. The domain adaptation is achieved by mapping the target data into the source domain using the trained generator. Then the mapped target data are classified using the CNN trained previously on the source data. In another work [50], the authors proposed an adversarial training for unsupervised pixel-based domain adaptation to make synthetic images more realistic. The generator in this model uses the source images as input instead of the noise vector.

The adaptation is achieved by transforming the source pixels directly to the target space, and the synthetic images help to maximize the accuracy of the classifier.

In the context of remote sensing, Lin et al. [43] used GANs for unsupervised scene classification. The model consists of a generator that learns to produce additional training images similar to the real data, and a discriminator that works as a feature extractor, which learns better representations of the images using the data provided by the generator. In another work, He et al. [44] proposed a semi-supervised method for the classification of hyperspectral images. Spectral–spatial features are extracted from the unlabeled images and are used to train a GAN model.

3. Proposed Methodology

In this work, we assume that we have only one source domain and one target domain. We are given a set of labeled images $Tr^{(s)} = \left\{\boldsymbol{I}_i^{(s)}, y_i\right\}_{i=1}^{n_s}$ from the source domain, where $y_i \in \{1, 2, \ldots, K\}$ is the corresponding class label and K is the number of classes. Additionally, we are given another set of unlabeled images $Ts^{(t)} = \left\{\boldsymbol{I}_j^{(t)}\right\}_{j=1}^{n_t}$ from the target domain. Our objective is to learn an invariant representation for both source and target domains by minimizing the mismatch of data distribution between the two domains. To this end, we propose a method based on the GANs theory, as shown in Figure 3. Detailed descriptions of the different blocks composing this network, in addition to the optimization process, are presented in next subsections.

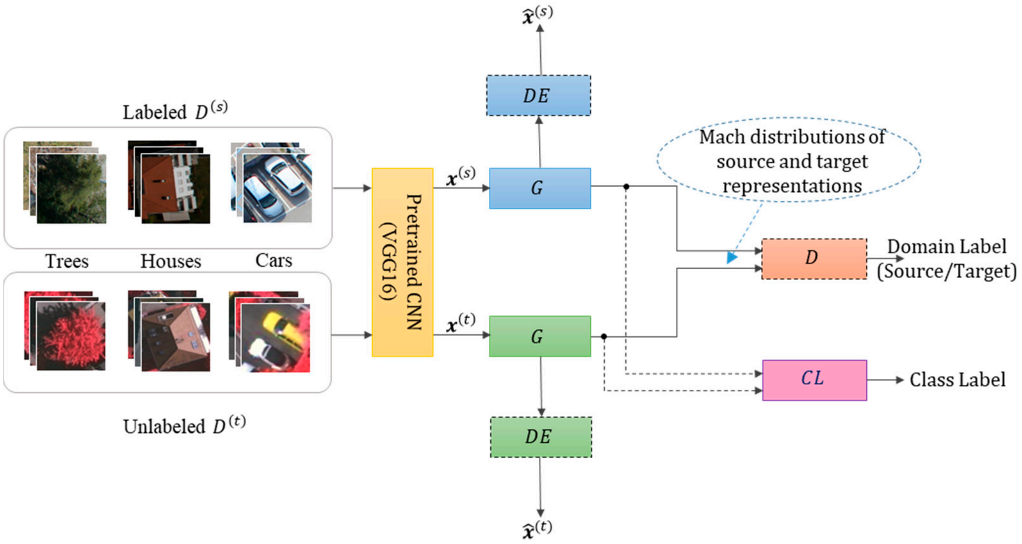


Figure 3. Proposed Siamese-GAN method.

3.1. Feature Extraction

We use the VGG16 network, which is a 16-layer network proposed by the VGG team in the ILSVRC 2014 competition [38]. This network is mainly composed of 13 convolutional layers, five pooling layers, and three fully connected layers. The network was trained on 1.2 million RGB images of 224 × 224 pixel size belonging to 1000 classes related to general images such as beaches, dogs, cats, cars, shopping carts, minivans, etc.

For feature extraction, we feed the labeled and unlabeled images to this pre-trained CNN and take the output of the activation function of the first fully connected layer. This results in high-level features of dimension 4096 as shown in Figure 4 We recall that other feature extractions or combinations at different levels of the network could be considered as well.

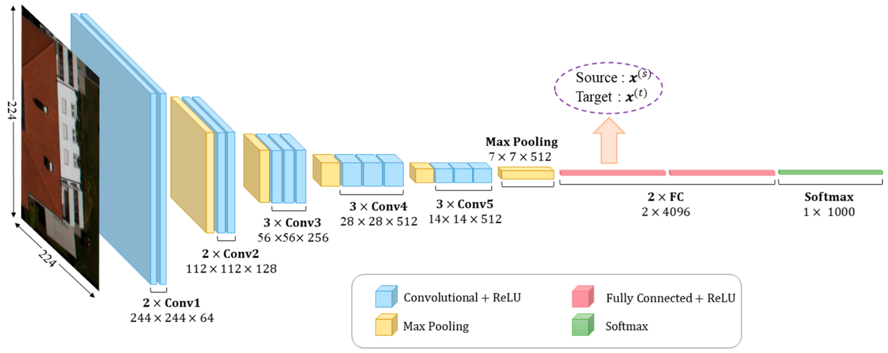


Figure 4. Feature extraction using a VGG16 pre-trained CNN.

3.2. Siamese-GAN Architecture

Figure 5 depicts the architecture of the different networks composing Siamese-GAN. First, we have a Siamese encoder–decoder network, where $G(\boldsymbol{W}_G)$ denotes the encoder part and $DE(\boldsymbol{W}_{DE})$ the decoder part. Then we have a discriminator denoted by $D(\boldsymbol{W}_D)$ and a classifier denoted by $CL(\boldsymbol{W}_{CL})$. Here the weights $\boldsymbol{W}_G$, $\boldsymbol{W}_{DE}$, $\boldsymbol{W}_D$ *and* $\boldsymbol{W}_{CL}$ refer to the learnable parameters associated with each component. The encoder G aims to match the source and target data samples into an embedded space, while the discriminator D tries to separate between the two domains. The decoders DE serve to constrain the mapping spaces to those allowing a good reconstruction of the original source and target samples. The classifier CL has the task of classifying the mapped target data samples after being learned on the mapped source data.

In detail, the encoder G receives feature vectors of dimension $d = 4096$ and maps them to features of dimension 128. This network consists of three dense layers, each followed by batch Normalization and leaky linear rectified unit (Leaky ReLU) activation function, except the last layer that uses a sigmoid activation function. The Leaky ReLU is similar to the standard rectified linear unit (ReLU), but with a small slope α in the negative region. In the experiments, we set this slope to 0.2. The output features obtained from the encoder are fed into the decoder that takes an input of dimension 128 and tries to reconstruct the original feature vector. The decoder also employs batch Normalization and Leaky ReLU for all layers except for the last layer, which uses sigmoid activation.

The discriminator receives as input a feature vector of dimension 128 from the encoder and outputs the domain prediction through binary classification. The output of the encoder is also passed to the classifier for multiclass classification through its softmax regression layer. For these networks, we consider also the dropout regularization technique to reduce overfitting. This technique randomly deactivates some neurons during the training phase, with a probability usually set to 0.5.

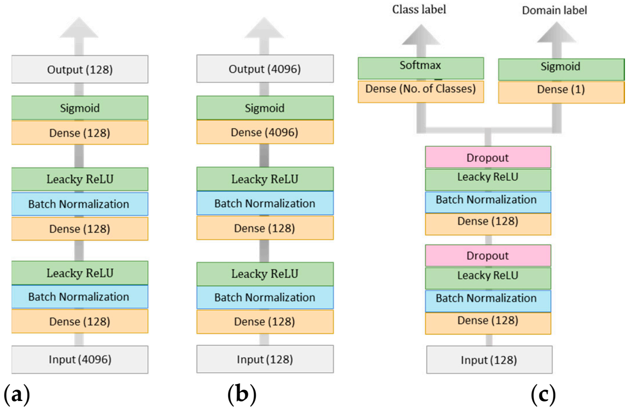


Figure 5. Architecture of the (**a**) encoder G, (**b**) decoder DE, (**c**) discriminator D, and classifier CL.

3.3. Network Optimization

Let us consider $Tr^{(s)} = \left\{\boldsymbol{x}_i^{(s)}, y_i\right\}_{i=1}^{n_s}$ and $Ts^{(t)} = \left\{\boldsymbol{x}_j^{(t)}\right\}_{j=1}^{n_t}$ the set of source and target features obtained from the pre-trained VGG16 network. To learn the parameters of the discriminator and the Siamese encoder sub-networks, we propose minimizing the following adversarial losses:

$$\mathcal{L}_D\left(D\left(\boldsymbol{x}^{(s)},\boldsymbol{x}^{(t)},\boldsymbol{W}_D\right)\right) = \mathbb{E}\left[\log D\left(G\left(\boldsymbol{x}^{(s)}\right)\right)\right] + \mathbb{E}\left[\log\left(1 - D\left(G\left(\boldsymbol{x}^{(t)}\right)\right)\right)\right] \tag{2}$$

$$\mathcal{L}_G\left(G\left(\boldsymbol{x}^{(s)},\boldsymbol{x}^{(t)},\boldsymbol{W}_G,\boldsymbol{W}_{DE}\right)\right) = \mathbb{E}\left[G\left(\boldsymbol{x}^{(s)}\right)\right] - \mathbb{E}\left[G\left(\boldsymbol{x}^{(t)}\right)\right]_2^2 + \lambda\mathbb{E}\left[\left(\boldsymbol{x}^{(s)} - \hat{\boldsymbol{x}}^{(s)}\right)^2\right] + \lambda\mathbb{E}\left[\left(\boldsymbol{x}^{(t)} - \hat{\boldsymbol{x}}^{(t)}\right)^2\right]. \tag{3}$$

The loss $\mathcal{L}_D\left(D\left(\boldsymbol{x}^{(s)},\boldsymbol{x}^{(t)},\boldsymbol{W}_D\right)\right)$ is the standard binary cross-entropy loss used by the original GANs for the discriminator. However, here the discriminator tries to distinguish between the source and target features received from the output of the Siamese encoder. On the other side, the loss of the Siamese encoder $\mathcal{L}_G\left(G\left(\boldsymbol{x}^{(s)},\boldsymbol{x}^{(t)},\boldsymbol{W}_G,\boldsymbol{W}_{DE}\right)\right)$ is composed of three terms. The first term seeks to match the distributions of the source and target data in order to confuse the discriminator. It can be expressed as follows:

$$\left\|\mathbb{E}\left[G\left(\boldsymbol{x}^{(s)}\right)\right] - \mathbb{E}\left[G\left(\boldsymbol{x}^{(t)}\right)\right]\right\|_2^2 = \left\|\frac{1}{n_s}\sum_{i=1}^{n_s} G\left(x_i^{(s)}\right) - \frac{1}{n_t}\sum_{j=1}^{n_t} G\left(x_i^{(t)}\right)\right\|_2^2. \tag{4}$$

The second and third terms represent the reconstruction error of the source and target data, respectively. They are expressed as follows:

$$\begin{cases} \mathbb{E}\left[\left(\boldsymbol{x}^{(s)} - \hat{\boldsymbol{x}}^{(s)}\right)^2\right] = \frac{1}{n_s}\sum_{i=1}^{n_s}\left(\boldsymbol{x}_i^{(s)} - \hat{\boldsymbol{x}}_i^{(s)}\right)^2 \\ \mathbb{E}\left[\left(\boldsymbol{x}^{(t)} - \hat{\boldsymbol{x}}^{(t)}\right)^2\right] = \frac{1}{n_t}\sum_{i=1}^{n_t}\left(\boldsymbol{x}_i^{(t)} - \hat{\boldsymbol{x}}_i^{(t)}\right)^2 \end{cases}. \tag{5}$$

These two losses are introduced for regularization purposes. That is to constrain the mapping spaces to those that allow a good reconstruction of the original features. In the experiments, we show that this regularization is crucial to obtain significant improvements in terms of classification accuracy. At the end of the adaptation process, we learn the parameters $\boldsymbol{W}_{CL}$ of the sub-network CL on the encoded labeled source data $G\left(\boldsymbol{x}^{(s)}\right)$ to discriminate between the different K classes by minimizing the multiclass cross-entropy loss $\mathcal{L}_{CL}\left(G(\boldsymbol{x}^{(s)}),\ \boldsymbol{W}_{CL}\right)$:

$$\mathcal{L}_{CL}\left(G(\boldsymbol{x}^{(s)}),\boldsymbol{W}_{CL}\right) = -\frac{1}{n_s}\sum_{i=1}^{n_s}\sum_{k=1}^{K} 1(y_i = k)logP\left(y_i = k\middle|G\left(x_i^{(s)}\right),\boldsymbol{W}_{CL}\right), \tag{6}$$

where $1(\cdot)$ is an indicator function that takes 1 if statement true otherwise it takes 0 and $P\left(y_i = k|G\left(x_i^{(s)}\right),\boldsymbol{W}_{CL}\right)$ is the probability output vector provided by the softmax regression layer placed on the top of the network CL.

To optimize the above loss functions, we use the backpropagation algorithm and the adaptive moment estimation (Adam) method for updating the parameters. The Adam method is an extension to the classical stochastic gradient descent (SGD) method. While SCD maintains a single learning rate for all weights during the training process, the Adam method computes individual adaptive learning rates for different parameters from estimates of first- and second-order moments of the gradients, which makes it very efficient.

In the following, we provide the main steps for training Siamese-GAN with its nominal parameters:

Algorithm. Siamese-GAN.

```
Input:   Source images: $Tr^{(s)} = \left\{ \boldsymbol{I}_i^{(s)}, y_i \right\}_{i=1}^{n_s}$ and target images: $Ts^{(t)} = \left\{ \boldsymbol{I}_i^{(t)} \right\}_{i=1}^{n_t}$
Output: Target class labels

1:  Set Network parameters:
      • $\lambda = 1$
      • Mini-batch size: $b = 100$
      • Adam parameters: learning rate: 0.0001, exponential decay rate for the first and second moments
        $\beta_1 = 0.9$, $\beta_2 = 0.999$ and epsilon = $1e^{-8}$
2:  Obtain pre-trained CNN features: $\boldsymbol{x}^{(s)} = VGG16\left(\boldsymbol{I}^{(s)}\right)$ and $\boldsymbol{x}^{(t)} = VGG16\left(\boldsymbol{I}^{(t)}\right)$
3:  Set the number of mini-batches: $n_b = n_s/b$
4:  for $epoch = 1 : \ num\_epoch$
      4.1  Shuffle randomly the source labeled samples and organize them into $n_b$ groups each of size $b$
      4.2  for $k = 1 : \ n_b$
             • Pick minibatch $k$ from the source data: $\boldsymbol{x}_k^{(s)} = \left\{ \boldsymbol{x}_i^{(s)} \right\}_{i=1+(k-1)n_b}^{kn_b}$
             • Pick randomly another minibatch of size $b$ from the target data $\boldsymbol{x}_{rand}^{(t)}$
             • Compute the encoded source and target features: $\mathrm{G}\left(\boldsymbol{x}_k^{(s)}\right)$ and $\mathrm{G}\left(\boldsymbol{x}_{rand}^{(t)}\right)$
             • Update the parameters $\boldsymbol{W}_D$ of the discriminator $D$ by minimizing the loss defined in (2) by
               training on $\mathrm{G}\left(\boldsymbol{x}_k^{(s)}\right)$ and $\mathrm{G}\left(\boldsymbol{x}_{rand}^{(t)}\right)$
             • Pick randomly new mini-batches $\boldsymbol{x}_{rand}^{(s)}$ and $\boldsymbol{x}_{rand}^{(t)}$ each of size $b$ from both source and
               target data
             • Update the parameters $\boldsymbol{W}_G$ and $\boldsymbol{W}_{DE}$ of the Siamese encoder by minimizing the loss
               defined in (3) by training on $\boldsymbol{x}_{rand}^{(s)}$ and $\boldsymbol{x}_{rand}^{(t)}$
5:  Feed the complete source $\boldsymbol{x}^{(s)}$ and target data $\boldsymbol{x}^{(t)}$ to the trained Siamese encoder to generate the final
    encoded data.
6:  Train the sub-network $CL$ on the encoded source data $\mathrm{G}\left(\boldsymbol{x}^{(s)}\right)$ data by minimizing the loss function
    defined in (5).
7:  Classify the encoded target data $\mathrm{G}\left(\boldsymbol{x}^{(t)}\right)$.
```

4. Experimental Results

4.1. Datasets Used for Creating the Cross-Domain Datasets

To evaluate the performance of the proposed method, we use four aerial datasets acquired with different sensors and altitudes and over diverse locations over the earth surface to build several benchmark cross-domain scenarios. Originally, these datasets were proposed for semantic segmentation and multilabel classification. Here, we tailor them to the context of cross-domain classification.

The first dataset was captured over Vaihingen city in Germany using Leica ALS50 system at an altitude of 500 m above ground level in July and August 2008. The resulting images are characterized by a spatial resolution of 9 cm. Each image is represented by three channels: near infrared (NIR), red (R), and green (G) channels. The dataset consists of three sub-regions: the inner city, the high riser and the residential area. The first area is situated in the center of the city, and is characterized by dense and complex historic buildings along with roads and trees. The second area consists of a few high-rise residential buildings surrounded by trees. The third area is a purely residential area with small detached houses and many surrounding trees.

The second dataset was taken over the central district of the city of Toronto in Canada by the Microsoft Vexcel's UltraCam-D camera and the Optech's airborne laser scanner (ALTM-ORION M) at an altitude of 650 m in February 2009. This dataset is located in a commercial zone that has representative scene characteristics of a modern mega city, containing buildings with a wide range of shape complexity in addition to trees and other urban objects. The resulting images have a ground resolution of 15 cm and RGB spectral channels.

The third dataset was acquired over the city of Potsdam using an airborne sensor. This dataset consists of RGB images with a ground resolution of 5 cm. Typically, this dataset contains several land cover classes such as buildings, vegetation, trees, cars, impervious surfaces, and other objects classified as background.

Finally, the Trento dataset consists of UAV images acquired over the city of Trento in Italy, on October 2011. These images were captured using a Canon EOS 550D camera with an 18 megapixels CMOS APS-C sensor. The dataset provides images with a ground resolution of approximately 2 cm and RGB spectral channels.

4.2. Cross-Domain Datasets Description

From the above four datasets, we build several cross-domain scenes by identifying the most common classes through visual inspection. For Toronto and Vaihingen, we identify nine common classes labeled as trees, grass, buildings, cars, roads, bare soil, water, solar panels, and train tracks. For the Trento and Potsdam datasets, we identified only eight classes, as the images for water and train track classes are unavailable for the first and second one, respectively. Table 1 summarizes the number of images per class extracted for each dataset, while Figure 6 shows some samples (cropped from the original images) normalized to the size 224 × 224 pixels. In the experiments, we refer to the resulting 12 transfer scenarios as source→target. For example, for the scenario Toronto→Vaihingen we have nine classes with 120 images per class. The total number of labeled source images and unlabeled target images used for learning is equal for both to 1080.

Table 1. Cross-domain scenarios built from Toronto, Trento, Vaihingen, and Potsdam datasets.

Class	**Number of Images per Dataset of Size: 224 × 224 Pixels**			
	Toronto	**Trento**	**Vaihingen**	**Potsdam**
Trees	120	120	120	120
Grass	120	120	120	120
Houses	120	120	120	120
Bare soil	120	120	120	120
Roads	120	120	120	120
Cars	120	120	120	120
Water	120	-	120	120
Solar Panels	120	120	120	120
Train Tracks	120	120	120	-
Total	1080	960	1080	960

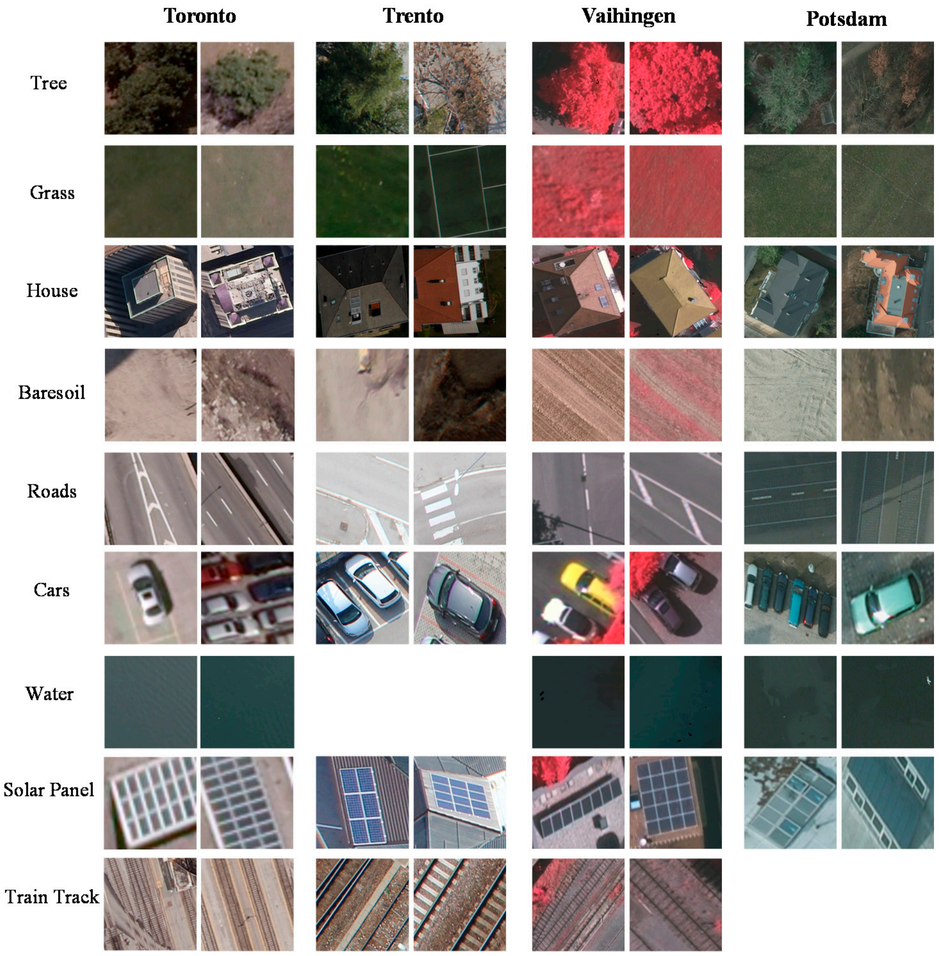


Figure 6. Sample EHR images used in the experiments.

4.3. Experimental Setup

We implement the Siamese-GAN method in a Keras environment, which is a high-level neural network application programming interface written in Python. For training the related subnetworks, we fix the mini-batch size to 100 samples. Additionally, we set the learning rate of the Adam optimization method to 0.0001. Regarding the exponential decay rates for the moment estimates and epsilon, we use the following default values 0.9 and 0.999 and $1e^{-8}$, respectively.

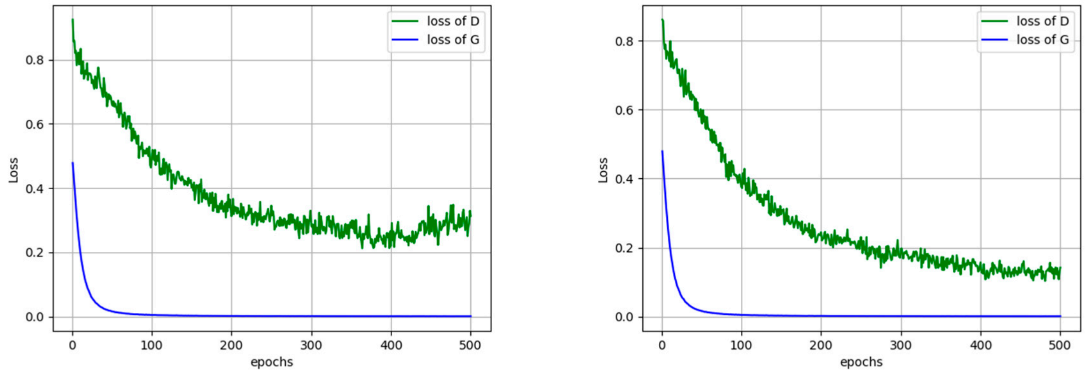
In the first set of experiments, we present the results by fixing the regularization parameter of the reconstruction loss to $\lambda = 1$. Next, we provide a detailed sensitivity analysis of Siamese-GAN with respect to this parameter, besides other features related to the network architecture. Finally, we compare our results to several state-of-the-art methods. For performance evaluation, we present the results on the unlabeled target images using per-class accuracy through confusion matrices, the overall accuracy (OA), which is the ratio of the number of correctly classified samples to the total number of the tested samples, and the average accuracy (AA) for each method, which represents the sum of the OA obtained for all scenarios divided by 12 (i.e., AA = OA/12). The experiments are performed on a MacBook Pro laptop (processor Intel Core i7 with a speed of 2.9 GHz, and 8 GB of memory).

4.4. Results

In this first set of experiments, we analyze the performance of our proposed method compared to the standard off-the-shelf classifiers solution. To this end, we first run the experiments by feeding the features extracted from VGG16 directly to an additional NN. This extra network has a similar architecture to the one shown in Figure 5c. Table 2 shows the classification accuracies for the 12 cross-domain scenarios. The lowest accuracy is obtained for Toronto→Vaihingen with an OA of 64.72%, while Potsdam→Trento shows the best result with an OA of 80.24%. Over the 12 scenarios, this solution yields an AA of 70.82%. We repeat these experiments using a linear multiclass SVM classifier with one-versus-one training strategy. We search for the best value of the regularization parameter according to a 3-fold cross-validation procedure in the range $[10^{-3}\ 10^{3}]$. In this case, the scenario Vaihingen→Potsdam shows relatively the lowest OA accuracy with 61.35%, while the best result is obtained for the scenario Potsdam→Trento with an OA of 86.55%. The average classification accuracy across the 12 scenarios is equal to 70.23%, which is very close to result obtained by the NN method.

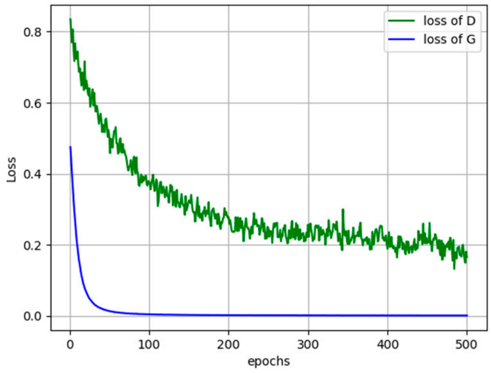
Next, we run the Siamese-GAN method as explained in Section 3.3. In Figure 7, we show the evolution of the Siamese encoder and discriminator losses. We recall that the Siamese encoder–decoder aims the match the distributions of both source and target while the discriminator seeks to discriminate them. The results reported in Table 2 show clearly that it improves greatly the AA accuracy for all scenarios from 70.81% to 90.34%, which corresponds to an increase of around 19%. For certain scenario like Trento→Vaihingen, it improves the OA by 28.85%. To understand better the behavior of the network, we show in Figure 8 the data distributions before and after adaptation for three typical scenarios, which are Potsdam→Vaihingen, Toronto→Vaihingen, and Trento→Toronto, respectively. This figure shows that the shift between the source and target distributions is obvious before adaptation, which explains the low performance obtained by off-the-shelf classifier solution. However, this discrepancy is greatly reduced by Siamese-GAN, while keeping the discrimination ability between the different classes.

In Figures 9–11 we report the confusion matrices before and after adaptation. For example for Potsdam→Vaihingen, the accuracies of classifying some classes with (NN) such as Water and House were already high before adaptation (96% and 97%), and have been increased to 100% with adaptation. For classes with low accuracies such as Grass, more than 60% of the images were misclassified as either Roads, Cars or Bare soil. The result has been improved with adaptation from 29% to 98%, which is equal to 69% gain in accuracy. Additionally, the confusion between Roads and Bare soil has been reduced, resulting in an increase from 68% to 94%. For Trento→Toronto, before adaptation 65% of Trees samples were misclassified as Bare soil and the accuracy has increased after adaptation from 33% to 60%. On the other hand, the confusion between Grass and Bare soil classes has been resolved with adaptation, and the classification accuracy of the Grass class increases from 43% to 100%. For Toronto→Vaihingen, the accuracy of Grass samples has been greatly increased from 0% to 92% with adaptation. However, the Roads class accuracy dropped from 73% to 43%.



(**a**) (**b**)

Figure 7. *Cont.*

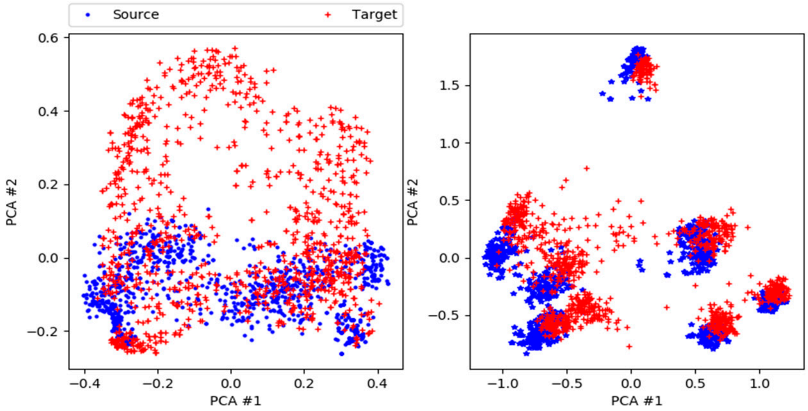


(**c**)

Figure 7. The adversarial losses of Siamese-GAN for the scenarios: (**a**) Potsdam→Vaihingen, (**b**) Trento→Toronto, and (**c**) Toronto→Vaihingen.

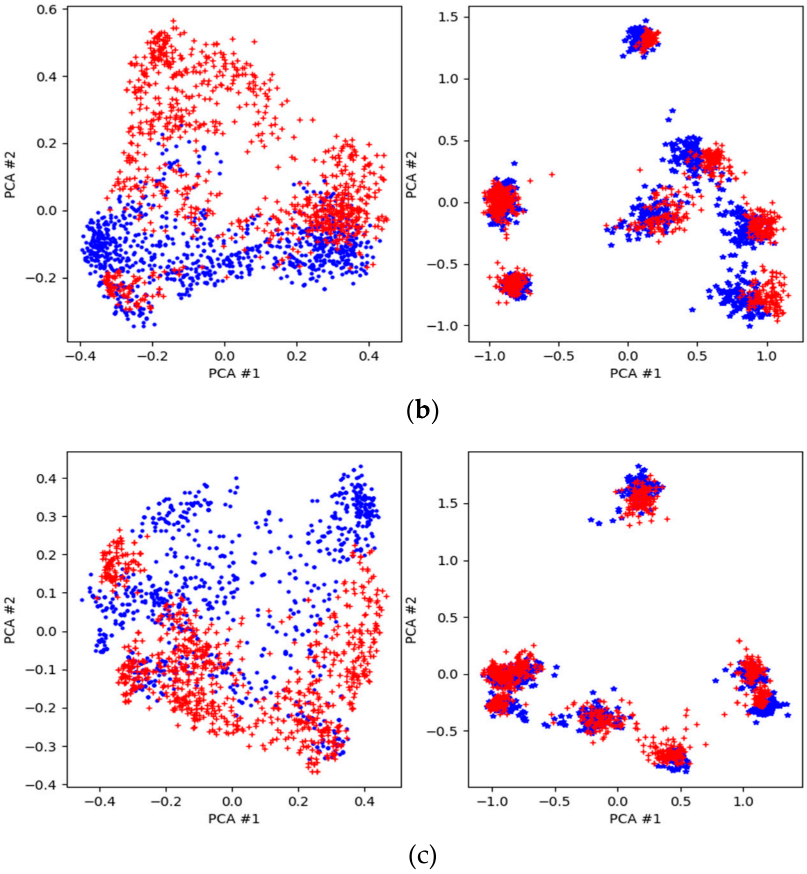
Table 2. Results are expressed in terms of OA [%] and AA [%] over the 12 scenarios.

Datasets	SVM	NN	Siamese-GAN
Toronto→Vaihingen	63.89	64.72	82.69
Toronto→Potsdam	68.96	69.17	84.27
Toronto→Trento	68.65	70.94	91.46
Vaihingen→Toronto	65.64	67.41	88.98
Vaihingen→Potsdam	61.35	65.10	88.33
Vaihingen→Trento	61.88	71.77	91.46
Potsdam→Toronto	72.19	70.83	92.71
Potsdam→Vaihingen	84.48	78.75	98.44
Potsdam→Trento	86.55	80.24	87.62
Trento→Toronto	68.23	70.21	91.56
Trento→Vaihingen	67.40	69.90	98.75
Trento→Potsdam	73.57	70.83	87.86
AA [%]	70.23	70.82	90.34



(**a**)

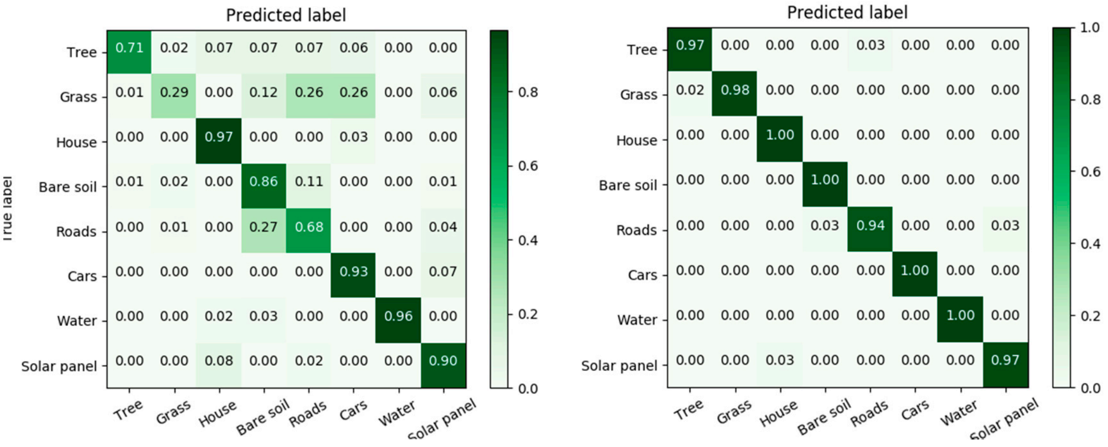
Figure 8. *Cont.*



(**b**)

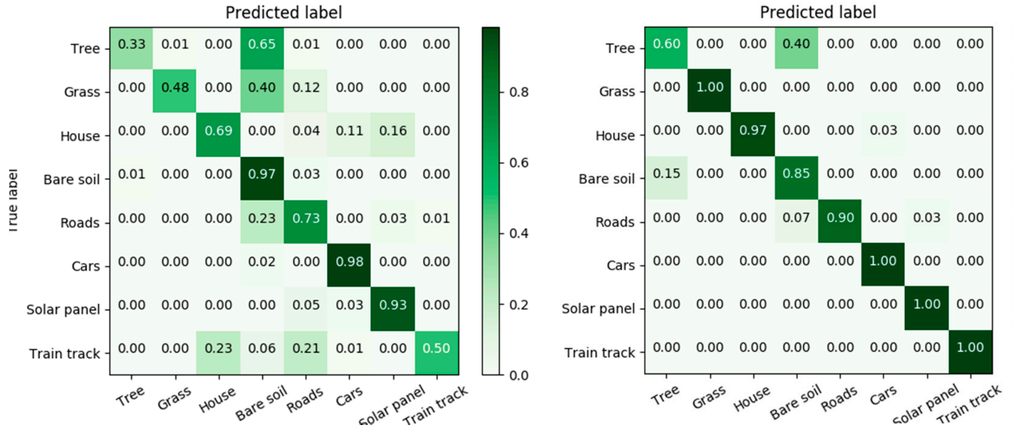
(c)

Figure 8. PCA for the transfers: (**a**) Potsdam→Vaihingen; (**b**) Trento→Toronto; (**c**) Toronto→Vaihingen. First column: before adaptation; second column: after adaptation.



(**a**) (**b**)

Figure 9. Confusion matrices for Potsdam→Vaihingen: (**a**) NN; (**b**) Siamese-GANs.



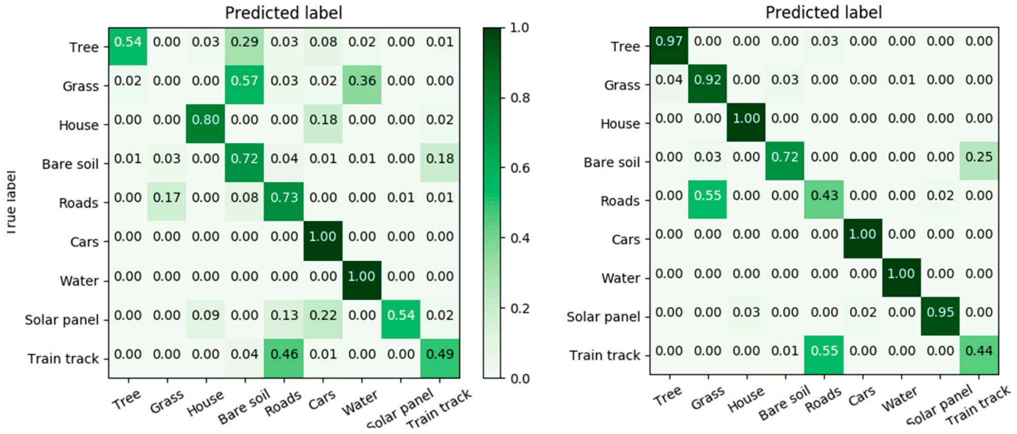
	Tree	Grass	House	Bare soil	Roads	Cars	Solar panel	Train track
Tree	0.33	0.01	0.00	0.65	0.01	0.00	0.00	0.00
Grass	0.00	0.48	0.00	0.40	0.12	0.00	0.00	0.00
House	0.00	0.00	0.69	0.00	0.04	0.11	0.16	0.00
Bare soil	0.01	0.00	0.00	0.97	0.03	0.00	0.00	0.00
Roads	0.00	0.00	0.00	0.23	0.73	0.00	0.03	0.01
Cars	0.00	0.00	0.00	0.02	0.00	0.98	0.00	0.00
Solar panel	0.00	0.00	0.00	0.00	0.05	0.03	0.93	0.00
Train track	0.00	0.00	0.23	0.06	0.21	0.01	0.00	0.50

(**a**)

	Tree	Grass	House	Bare soil	Roads	Cars	Solar panel	Train track
Tree	0.60	0.00	0.00	0.40	0.00	0.00	0.00	0.00
Grass	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
House	0.00	0.00	0.97	0.00	0.00	0.03	0.00	0.00
Bare soil	0.15	0.00	0.00	0.85	0.00	0.00	0.00	0.00
Roads	0.00	0.00	0.00	0.07	0.90	0.00	0.03	0.00
Cars	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
Solar panel	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
Train track	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00

(**b**)

Figure 10. Confusion matrices for Trento→Toronto: (**a**) NN; (**b**) Siamese-GANs.



	Tree	Grass	House	Bare soil	Roads	Cars	Water	Solar panel	Train track
Tree	0.54	0.00	0.03	0.29	0.03	0.08	0.02	0.00	0.01
Grass	0.02	0.00	0.00	0.57	0.03	0.02	0.36	0.00	0.00
House	0.00	0.00	0.80	0.00	0.00	0.18	0.00	0.00	0.02
Bare soil	0.01	0.03	0.00	0.72	0.04	0.01	0.01	0.00	0.18
Roads	0.00	0.17	0.00	0.08	0.73	0.00	0.00	0.01	0.01
Cars	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
Water	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
Solar panel	0.00	0.00	0.09	0.00	0.13	0.22	0.00	0.54	0.02
Train track	0.00	0.00	0.00	0.04	0.46	0.01	0.00	0.00	0.49

(**a**)

	Tree	Grass	House	Bare soil	Roads	Cars	Water	Solar panel	Train track
Tree	0.97	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00
Grass	0.04	0.92	0.00	0.03	0.00	0.00	0.01	0.00	0.00
House	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
Bare soil	0.00	0.03	0.00	0.72	0.00	0.00	0.00	0.00	0.25
Roads	0.00	0.55	0.00	0.00	0.43	0.00	0.00	0.02	0.00
Cars	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
Water	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
Solar panel	0.00	0.00	0.03	0.00	0.00	0.02	0.00	0.95	0.00
Train track	0.00	0.00	0.00	0.01	0.55	0.00	0.00	0.00	0.44

(**b**)

Figure 11. Confusion matrices for Toronto→Vaihingen: (**a**) NN; (**b**) Siamese-GANs.

5. Discussion

Effect of the reconstruction loss: To investigate the effectiveness of the reconstruction loss on the classification performances of the method, we repeat the above experiments by varying the values of the regularization parameter λ in the range [0, 1]. The results reported in Table 3 clearly suggest that setting this parameter in the range [0.4, 1] yields a stable behavior. For the case $\lambda = 0$, corresponding to the removal of the decoder part (i.e., no-reconstruction loss), the results drop significantly to 77.89%. Yet the results are still better than SVM and NN. This indicates clearly the importance of the decoder part in keeping the geometrical structure of the source and target data when matching the distributions.

Table 3. Sensitivity analysis with respect to the regularization parameter λ. Results are expressed in terms of OA [%] and AA [%] over the 12 scenarios.

Datasets	**Regularization Parameter λ**					
	0	0.2	0.4	0.6	0.8	1
Toronto→Vaihingen	75.74	78.06	85.74	83.06	83.61	82.69
Toronto→Potsdam	73.85	83.85	84.27	84.58	86.56	84.27
Toronto→Trento	73.12	91.98	92.4	91.46	92.08	91.46
Vaihingen→Toronto	72.96	88.24	88.52	89.16	88.06	88.98
Vaihingen→Potsdam	67.5	88.65	87.6	88.33	88.54	88.33
Vaihingen→Trento	78.75	84.79	92.71	92.6	91.98	91.46

Table 3. *Cont.*

Datasets	Regularization Parameter λ					
Potsdam→Toronto	76.25	91.98	91.76	92.5	93.23	92.71
Potsdam→Vaihingen	90.83	98.12	98.23	98.12	98.54	98.44
Potsdam→Trento	85	85.83	87.02	87.02	87.14	87.62
Trento→Toronto	76.15	91.46	91.77	92.7	91.04	91.56
Trento→Vaihingen	87.6	98.12	98.65	98.44	98.85	98.75
Trento→Potsdam	76.9	89.76	89.52	88.57	89.05	87.86
AA [%]	77.89	89.24	90.68	90.55	90.72	90.34

Effect of mini-batch size b: Table 4 shows the results obtained using different mini-batch sizes for aligning the distributions of source and target data. The results exhibits a stable behavior in the range [40 100]. Decreasing further the min-batch size leads to a significant decrease in the classification accuracy. As can be seen, the choice of $b = 100$ is a good compromise between accuracy and computation time.

Comparison with state of the art: We compare the performance of Siamese-GAN with other domain adaptation methods proposed in the literature. These are maximum independence domain adaptation (MIDA) [51], which learns a subspace that has maximum independence with the domain features. The correlation alignment (CORAL) [52], which minimizes the domain shift by aligning the second order statistics of the source and target distributions. The domain adaptation network (DAN) method [37], which aims to project the source and target data into a common space to reduce the discrepancy between source and target distributions while using graph regularization to maintain the geometrical structure of the target data. The adversarial discriminative domain adaptation (ADDA) [49], which combines adversarial and discriminative learning. Table 5 shows that Siamese-GAN provides better results for ten cases except for Toronto→Vaihingen and Vaihingen→Potsdam, where the DAN method yields better results. On average, it yields and AA of 90.34% whereas the DAN method got 85.48%.

Table 4. Sensitivity analysis with respect to the min-batch size $\boldsymbol{b}$. Results are expressed in terms of OA [%] and AA [%] over the 12 scenarios.

Datasets	Mini-Batch Size b					
	10	20	40	60	80	100
Toronto→Vaihingen	74.9	78.06	86.67	93.06	91.57	82.69
Toronto→Postdam	70.1	79.79	85.1	86.77	84.27	84.27
Toronto→Trento	71.67	83.44	90.73	93.02	85.83	91.46
Vaihingen→Toronto	73.15	78.98	88.8	89.44	89.07	88.98
Vaihingen→Postdam	62.5	71.15	86.04	87.29	86.77	88.33
Vaihingen→Trento	72.5	86.25	93.75	86.25	84.16	91.46
Postdam→Toronto	72.81	87.29	90.52	92.19	93.02	92.71
Postdam→Vaihingen	84.48	96.56	98.23	97.92	98.44	98.44
Postdam→Trento	72.62	83.57	89.17	88.33	87.74	87.62
Trento→Toronto	55.63	89.48	91.04	91.46	91.46	91.56
Trento→Vaihingen	75.83	97.7	97.19	97.5	98.75	98.75
Trento→Postdam	73.45	81.67	88.81	90.36	87.74	87.86
AA [%]	71.47	84.50	90.50	91.13	89.90	90.34
Time [minutes]	15.82	8.57	4.83	3.71	3.05	2.84

Table 5. Comparison with several state-of-the-art methods. Results are expressed in terms of OA [%] and AA [%] over the 12 scenarios.

Datasets	DAN	CORAL	MIDA	ADDA	Siamese-GAN
Toronto→Vaihingen	90.00	74.25	70.00	68.51	82.69
Toronto→Potsdam	79.89	72.81	70.83	73.22	84.27
Toronto→Trento	88.12	83.12	66.77	72.08	91.46

Table 5. *Cont.*

Datasets	DAN	CORAL	MIDA	ADDA	Siamese-GAN
Vaihingen→Toronto	77.59	79.35	77.50	77.87	88.98
Vaihingen→Potsdam	91.14	81.66	81.04	76.04	88.33
Vaihingen→Trento	82.08	77.50	75.10	69.27	91.46
Potsdam→Toronto	88.54	72.70	76.14	75.41	92.71
Potsdam→Vaihingen	84.06	86.00	88.43	82.49	98.44
Potsdam→Trento	87.14	84.28	86.04	86.91	87.62
Trento→Toronto	86.77	82.39	72.91	79.68	91.56
Trento→Vaihingen	84.68	80.41	81.56	79.58	98.75
Trento→Potsdam	85.83	82.26	79.76	75.71	87.86
AA [%]	85.48	79.72	77.17	76.39	90.34
Time [minutes]	7.18	2.54	1.77	3.03	2.84

6. Conclusions

In this work, we have proposed a GAN-based method for cross-domain categorization in aerial vehicle images. This method learns invariant feature representations by training two competing networks. The first network aims to reduce the discrepancy between source and target distributions, while the second one seeks to distinguish between them. The experimental results conducted on several datasets acquired by different MAV/UAV platforms and over different locations of the earth surface have shown the effectiveness of our model.

Acknowledgments: This work was supported by the Deanship of Scientific Research at King Saud University through the Local Research Group Program under Project RG-1435-055. The authors would like to thank Farid Melgani from the University of Trento for providing the Trento dataset. The authors would like to acknowledge the provision of the Postdam, Vaihingen, and Toronto datasets by ISPRS and BSF Swissphoto, released in conjunction with the ISPRS, led by ISPRS WG II/4.

Author Contributions: Laila Bashmal and Yakoub Bazi designed and implemented the method, and wrote the paper. Haikel AlHichri, Mohamad M. AlRahhal, Nassim Ammour, and Naif Alajlan contributed to the analysis of the experimental results and paper writing.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
2. Ojala, T.; Pietikäinen, M.; Harwood, D. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognit.* **1996**, *29*, 51–59. [CrossRef]
3. Chen, S.; Tian, Y. Pyramid of spatial relatons for scene-level land use classification. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 1947–1957. [CrossRef]
4. Zhu, Q.; Zhong, Y.; Zhao, B.; Xia, G.S.; Zhang, L. Bag-of-visual-words scene classifier with local and global features for high spatial resolution remote sensing imagery. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 747–751. [CrossRef]
5. Zou, J.; Li, W.; Chen, C.; Du, Q. Scene classification using local and global features with collaborative representation fusion. *Inf. Sci. (Ny)* **2016**, *348*, 209–226. [CrossRef]
6. Zhao, L.J.; Tang, P.; Huo, L.Z. Land-use scene classification using a concentric circle-structured multiscale bag-of-visual-words model. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 4620–4631. [CrossRef]
7. Cheriyadat, A.M. Unsupervised feature learning for aerial scene classification. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 439–451. [CrossRef]
8. Mekhalfi, M.L.; Melgani, F.; Bazi, Y.; Alajlan, N. Land-use classification with compressive sensing multifeature fusion. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 2155–2159. [CrossRef]
9. Zhong, Y.; Zhu, Q.; Zhang, L. Scene classification based on the multifeature fusion probabilistic topic model for high spatial resolution remote sensing imagery. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 6207–6222. [CrossRef]

10. Cheng, G.; Han, J.; Guo, L.; Liu, Z.; Bu, S.; Ren, J. Effective and efficient midlevel visual elements-oriented land-use classification using vhr remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 4238–4249. [CrossRef]
11. Li, Y.; Tao, C.; Tan, Y.; Shang, K.; Tian, J. Unsupervised multilayer feature learning for satellite image scene classification. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 157–161. [CrossRef]
12. Hu, F.; Xia, G.S.; Wang, Z.; Huang, X.; Zhang, L.; Sun, H. Unsupervised feature learning via spectral clustering of multidimensional patches for remotely sensed scene classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 2015–2030. [CrossRef]
13. Zhao, B.; Zhong, Y.; Xia, G.S.; Zhang, L. Dirichlet-derived multiple topic scene classification model for high spatial resolution remote sensing imagery. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 2108–2123. [CrossRef]
14. Mohamed, A.; Dahl, G.E.; Hinton, G. Acoustic modeling using deep belief networks. *IEEE Trans. Audio Speech Lang. Process.* **2012**, *20*, 14–22. [CrossRef]
15. Vega, P.J.S.; Feitosa, R.Q.; Quirita, V.H.A.; Happ, P.N. Single sample face recognition from video via stacked supervised auto-encoder. In Proceedings of the 29th Graphics, Patterns and Images (SIBGRAPI) Conference, Sao Paulo, Brazil, 4–7 October 2016; pp. 96–103.
16. Brosch, T.; Tam, R. Efficient training of convolutional deep belief networks in the frequency domain for application to high-resolution 2D and 3D Images. *Neural Comput.* **2015**, *27*, 211–227. [CrossRef] [PubMed]
17. Hayat, M.; Bennamoun, M.; An, S. Deep reconstruction models for image set classification. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 713–727. [CrossRef] [PubMed]
18. Hinton, G.E. Reducing the dimensionality of data with neural networks. *Science* **2006**, *313*, 504–507. [CrossRef] [PubMed]
19. Hinton, G.E.; Osindero, S.; Teh, Y.-W. A fast learning algorithm for deep belief nets. *Neural Comput.* **2006**, *18*, 1527–1554. [CrossRef] [PubMed]
20. Vincent, P.; Larochelle, H.; Bengio, Y.; Manzagol, P.A. Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th International Conference on Machine Learning, New York, NY, USA, 5–9 July 2008; pp. 1096–1103.
21. Farabet, C.; Couprie, C.; Najman, L.; LeCun, Y. Learning hierarchical features for scene labeling. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1915–1929. [CrossRef] [PubMed]
22. Luus, F.P.S.; Salmon, B.P.; van den Bergh, F.; Maharaj, B.T.J. Multiview deep learning for land-use classification. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 2448–2452. [CrossRef]
23. Zou, Q.; Ni, L.; Zhang, T.; Wang, Q. Deep Learning based feature selection for remote sensing scene classification. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 2321–2325. [CrossRef]
24. Wu, H.; Liu, B.; Su, W.; Zhang, W.; Sun, J. Deep filter banks for land-use scene classification. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 1895–1899. [CrossRef]
25. Zhang, F.; Du, B.; Zhang, L. Scene classification via a gradient boosting random convolutional network framework. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 1793–1802. [CrossRef]
26. zegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
27. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems, Lake Tahoe, Nevada, 3–6 December 2012; pp. 1097–1105.
28. Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; Darrell, T. Caffe: Convolutional architecture for fast feature embedding. In Proceedings of the 22nd ACM international conference on Multimedia, Orlando, FL, USA, 3–7 November 2014; pp. 675–678.
29. Scott, G.J.; England, M.R.; Starms, W.A.; Marcum, R.A.; Davis, C.H. Training deep convolutional neural networks for land-cover classification of high-resolution imagery. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 549–553. [CrossRef]
30. Nogueira, K.; Penatti, O.A.B.; dos Santos, J.A. Towards better exploiting convolutional neural networks for remote sensing scene classification. *Pattern Recognit.* **2017**, *61*, 539–556. [CrossRef]
31. Marmanis, D.; Datcu, M.; Esch, T.; Stilla, U. Deep learning earth observation classification using imagenet pretrained networks. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 105–109. [CrossRef]

32. Hu, F.; Xia, G.S.; Hu, J.; Zhang, L. Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. *Remote Sens.* **2015**, *7*, 14680–14707. [CrossRef]
33. Othman, E.; Bazi, Y.; Alajlan, N.; Alhichri, H.; Melgani, F. Using convolutional features and a sparse autoencoder for land-use scene classification. *Int. J. Remote Sens.* **2016**, *37*, 1977–1995. [CrossRef]
34. Wang, G.; Fan, B.; Xiang, S.; Pan, C. Aggregating rich hierarchical features for scene classification in remote sensing imagery. *IEEE J. Sel. Top. Appl. EARTH Obs. Remote Sens.* **2017**, *10*, 4104–4115. [CrossRef]
35. Weng, Q.; Mao, Z.; Lin, J.; Guo, W. Land-use classification via extreme learning classifier based on deep convolutional features. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 704–708. [CrossRef]
36. Chaib, S.; Liu, H.; Gu, Y.; Yao, H. Deep feature fusion for VHR remote sensing scene classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 4775–4784. [CrossRef]
37. Othman, E.; Bazi, Y.; Melgani, F.; Alhichri, H.; Alajlan, N.; Zuair, M. Domain adaptation network for cross-scene classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 4441–4456. [CrossRef]
38. Radford, A.; Metz, L.; Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. Available online: https://arxiv.org/abs/1511.06434 (accessed on 23 February 2018).
39. Mirza, M.; Osindero, S. Conditional Generative Adversarial Nets. Available online: https://arxiv.org/abs/1411.1784 (accessed on 23 February 2018).
40. Tan, W.R.; Chan, C.S.; Aguirre, H.; Tanaka, K. ArtGAN: Artwork Synthesis with Conditional Categorial Gans. Available online: https://arxiv.org/abs/1702.03410 (accessed on 23 February 2018).
41. Zhang, H.; Xu, T.; Li, H.; Zhang, S.; Huang, X.; Wang, X.; Metaxas, D. Stackgan: Text to Photo-Realistic Image Synthesis with Stacked Generative Adversarial Networks. Available online: https://arxiv.org/abs/1612.03242 (accessed on 23 February 2018).
42. Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. Available online: https://arxiv.org/abs/1609.04802 (accessed on 23 February 2018).
43. Lin, D.; Fu, K.; Wang, Y.; Xu, G.; Sun, X. MARTA GANs: Unsupervised representation learning for remote sensing image classification. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 2092–2096. [CrossRef]
44. He, Z.; Liu, H.; Wang, Y.; Hu, J. Generative Adversarial networks-based semi-supervised learning for hyperspectral image classification. *Remote Sens.* **2017**, *9*, 1042. [CrossRef]
45. Suarez, P.L.; Sappa, A.D.; Vintimilla, B.X. Infrared image colorization based on a triplet DCGAN architecture. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 212–217.
46. Li, J.; Skinner, K.A.; Eustice, R.M.; Johnson-Roberson, M. WaterGAN: Unsupervised generative network to enable real-time color correction of monocular underwater images. *IEEE Robot. Autom. Lett.* **2018**, *3*, 387–394. [CrossRef]
47. Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; Lempitsky, V. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.* **2016**, *17*, 1–35.
48. Liu, M.Y.; Tuzel, O. Coupled Generative Adversarial Networks. Available online: https://arxiv.org/abs/1606.07536 (accessed on 23 February 2018).
49. Tzeng, E.; Hoffman, J.; Saenko, K.; Darrell, T. Adversarial Discriminative Domain Adaptation. Available online: https://arxiv.org/abs/1702.05464 (accessed on 17 February 2017).
50. Bousmalis, K.; Silberman, N.; Dohan, D.; Erhan, D.; Krishnan, D. Unsupervised pixel-level domain adaptation with generative adversarial networks. *arXiv*, 2016. [CrossRef]
51. Yan, K.; Kou, L.; Zhang, D. Learning domain-invariant subspace using domain features and independence maximization. *IEEE Trans. Cybern.* **2018**, *48*, 288–299. [CrossRef] [PubMed]
52. Sun, B.; Feng, J.; Saenko, K. Return of frustratingly easy domain adaptation. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, Arizona, 12–17 February 2016; pp. 2058–2065.