remote sensing

MDPI

Article

Mining Hard Negative Samples for SAR-Optical Image Matching Using Generative Adversarial Networks

Lloyd Haydn Hughes [1], Michael Schmitt [1] and Xiao Xiang Zhu [1,2,*]

[1] Signal Processing in Earth Observation, Technical University of Munich (TUM), 80333 Munich, Germany; lloyd.hughes@tum.de (L.H.H.); m.schmitt@tum.de (M.S.)

[2] Remote Sensing Technology Institute (IMF), German Aerospace Center (DLR), 82234 Wessling, Germany

* Correspondences: xiaoxiang.zhu@dlr.de; Tel.: +49-89-289-22657

Received: 13 August 2018; Accepted: 25 September 2018; Published: 27 September 2018

Abstract: In this paper, we propose a generative framework to produce similar yet novel samples for a specified image. We then propose the use of these images as hard-negatives samples, within the framework of hard-negative mining, in order to improve the performance of classification networks in applications which suffer from sparse labelled training data. Our approach makes use of a variational autoencoder (VAE) which is trained in an adversarial manner in order to learn a latent distribution of the training data, as well as to be able to generate realistic, high quality image patches. We evaluate our proposed generative approach to hard-negative mining on a synthetic aperture radar (SAR) and optical image matching task. Using an existing SAR-optical matching network as the basis for our investigation, we compare the performance of the matching network trained using our approach to the baseline method, as well as to two other hard-negative mining methods. Our proposed generative architecture is able to generate realistic, very high resolution (VHR) SAR image patches which are almost indistinguishable from real imagery. Furthermore, using the patches as hard-negative samples, we are able to improve the overall accuracy, and significantly decrease the false positive rate of the SAR-optical matching task—thus validating our generative hard-negative mining approaches' applicability to improve training in data sparse applications.

Keywords: synthetic aperture radar; generative adversarial networks; data fusion; dataset augmentation

1. Introduction

In recent years, data fusion has become a hot topic in the field of remote sensing, specifically the fusion of heterogeneous image data. This increased interest has largely been driven by the improved availability of remote sensing imagery acquired by different sensors [1].

As with any image based data fusion endeavour, a key first step is the determination of corresponding image parts. While considered a somewhat solved problem in traditional computer vision, image matching remains a challenging task when dealing with heterogeneous remote sensing data. One prominent example of this is matching synthetic aperture radar (SAR) and optical satellite imagery, where the sensors have vastly different geometric and radiometric properties making image matching a deeply complex problem [2].

In order to deal with these challenges, several sophisticated approaches have been proposed. Ye et al. [3] propose exploiting phase congruency as a generalization of the gradient information in order to match multimodal images. The approach presented in [4] extends this use of phase congruency to create a radiation-invariant feature transform, which is less susceptible to nonlinear radiation distortions. Using an epipolar-like search strategy and template matching, Qiu et al. [5]

proposed a strategy for simultaneous tie-point matching and 3D reconstruction relying on classical signal- and descriptor-based similarity measures.

While these approaches perform well in some circumstances, most still rely on hand-crafted features and template matching which are difficult to adapt and often suffer from poor discriminability in very high resolution (VHR) imagery. An example of such a failure case can be found when matching very high resolution (VHR) heterogeneous imagery of urban environments, which—in the SAR case—is often difficult even for trained experts to interpret and match [6].

More recently, deep learning has been applied to numerous optical image matching problems with great success. Initial approaches replaced handcrafted feature descriptors with descriptors learned using convolutional neural networks (CNN) [7,8]. However, these were soon outperformed by learning an end-to-end similarity metric for image matching, directly from the data [9,10].

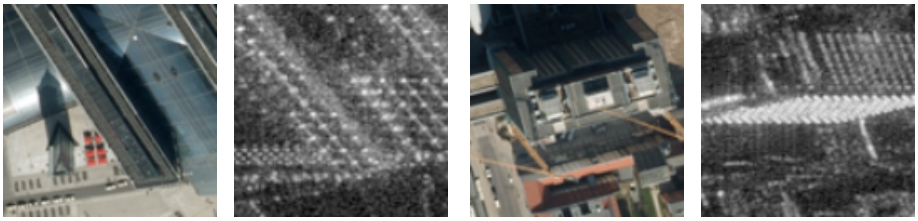
Based on the demonstrated successes in computer vision, deep learning approaches have been gaining interest in the remote sensing community [11]. One possible application is found in the matching of extremely multimodal Earth observation imagery. Merkle et al. [12] proposed the use of a Siamese CNN architecture to compute the relative shift between SAR and optical image patches, with the goal of improving the geo-localization accuracy of optical imagery. Taking inspiration from this success, Mou et al. [13] proposed the use of a pseudo-Siamese CNN in order to frame the SAR-optical correspondence problem as binary classification. With this approach, they provided a proof of concept towards the applicability of CNNs for matching heterogeneous remote sensing imagery. Hughes et al. [14] extended this initial investigation through a modified fusion layer and softmax loss function in order to compute a similarity probability score. Additionally, the investigation was extended to simulate a real-world feature matching scenario and was able to achieve around 86% accuracy with an 11% false positive rate (FPR). Taking a different approach, Merkle et al. [15] proposed the use of a generative adversarial network (GAN) to generate SAR like patches from medium resolution optical images. These generated SAR like patches were then used as the template in a template matching application. This hybrid approach was able to achieve an accuracy of 82% when the threshold for alignment was limited to an error of three pixels.

While these results show great promise for future applications, there has been little to no focus placed on the importance of matching within the scope of a low false positive rate (FPR)—which is arguably more important than achieving a high true positive rate. This requirement is largely driven by the need to reduce outliers in matching results in order to assist downstream applications subsequent to the matching step. This is especially true in multimodal data fusion tasks, such as SAR-optical stereogrammetry [5], where few other methods exist to detect and remove incorrect correspondences.

One common approach to improve the discriminability between classes in classification tasks, and thus reduce the FPR, is known as hard negative mining. This technique uses hard samples (samples which are statistically similar but belong to different classes) as negative examples during the training phase of the classifier [16]. Unlike the randomly assigned negative pairs used in [14], hard negative mining progressively increases the difficulty of the negative examples that the network is trained on. This is done by augmenting the data loading pipeline to replace or append the dataset with data samples which had the greatest misclassification in the previous training iteration. In other words, the samples which are classified as the incorrect class in the most certain manner are now included in the next training iterations as negative examples, thus reinforcing to the classifier that the result is incorrect.

While hard negative mining is simple to implement, it requires that the original dataset is large enough such that, even for low false positive rates, sufficient negative samples exist that can be used as hard negative samples for training. In conventional deep learning applications, this data constraint is often not an issue, as datasets are large enough or can easily be extended. However, for SAR-optical matching applications, this is not the case. While access to remote sensing imagery is becoming easier, and images are geo-coded, the vast differences in imaging geometry mean that geo-coded points cannot be trivially matched. This is particularly true for very high resolution data (see Figure 1).

Thus, expert knowledge or the use of computationally expensive procedures are often required in order to align and match the images such that an accurate patch pair dataset can be produced and labeled [17–19].



(**a**) (**b**) (**c**) (**d**)

Figure 1. An illustrative example of the vast differences between synthetic apature radar (SAR) and optical remote sensing imagery of the same scene. The corresponding image patch-pairs (**a**–**d**) would prove challenging to determine correspondence, even for experts in the field.

To overcome these issues related to data sparsity, researchers have turned to generative networks in order to generate artificial data which can be used to augment or pre-train deep architectures and thus reduce the requirement for large amounts of labeled data [20–23]. Zheng et al. [21] propose using a generative adversarial network (GAN) to generate unlabeled data which was used to improve the baseline in a person-re-identification task. They argued that the imperfect, generated samples act as a form of regularization and thus lead to a more discriminative classifier. In [22], the authors train a SAR to optical transcoding GAN in order to learn key features between various different land surfaces. The top layers of the generator are then used as the main feature extraction sub-network in a multi-modal land cover classifier. Their results show a significant improvement when compared to training the classifier from scratch. Ref. [23] utilizes a GAN to generate negative triplet embeddings in order to allow the discriminator to learn better embedding models.

Marmanis et al. [24] generated VHR SAR patches in order to increase their dataset size for training a SAR image classification network. While the quality of their generated images appears reasonable, they were unable to realize any conclusive results as to whether generated data improved their classification network. Ao et al. [25] proposed a Dialectical-GAN in order to generate VHR SAR imagery from a low resolution Sentinel-1 SAR image prior. However, their results were used as a proof of concept in image translation and were not applied to training of other tasks.

In this paper, we propose an alternative formulation of hard negative mining that can be applied to data sparse applications, such as SAR-optical matching, in order to improve the discriminability of the network and thus reduce the false positive rate. The main contributions are summarized as follows:

Firstly, a GAN architecture is proposed which is capable of generating realistic SAR images which look similar to an existing SAR image, but are modifiable via a continuous latent space. We validate that our generated SAR images are suitable for hard negative mining.

Secondly, we describe how these generated SAR images can be used as hard negative samples to train an existing SAR-optical matching network.

Finally, we demonstrate the effectiveness of our proposed approach by evaluating it on the matching network proposed in [14], and show how we are able to significantly decrease the false positive rate via hard negative mining for the first time.

2. Generative Framework for Hard Negative Mining

In this section, the main structure of our proposed approach to hard negative mining will be described, including our GAN based architecture and training procedure. We will further describe how this architecture can be incorporated into the SAR-optical matching network proposed in [14] in order to augment the training procedure with hard negative samples and thus reduce the false positive rate. An overview of our approach can be seen in Figure 2.

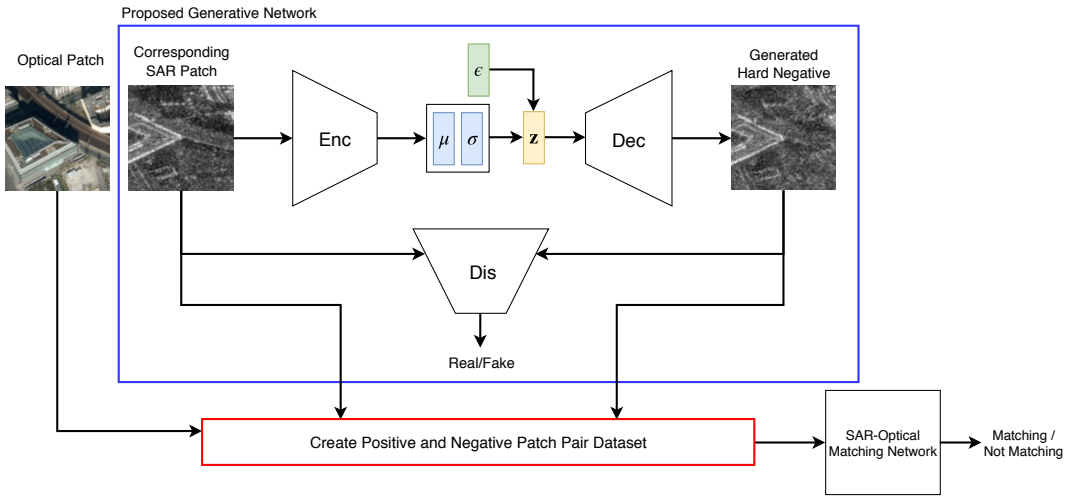


Figure 2. The proposed generative hard-negative mining framework. The GAN is trained to create hard-negative samples based on the input image patch. Together with the original corresponding optical patch, these samples are then used to train the SAR-optical matching network.

2.1. Proposed Generative Architecture

2.1.1. Generator

In order to generate hard negative SAR samples, which are suitable for training a VHR SAR-optical matching network, we make use of a generative model which is trained in an adversarial setting. More specifically, we extend the ProGAN architecture proposed by Karras et al. [26] to include an encoder network which learns a latent representation of our data. This latent representation in turn is used to generate new images. These modifications re-position the original generator network as the decoder network in an autoencoder (AE). We additionally impose a prior over our latent space, $p(\mathbf{z})$, to transform this AE into a variational autoencoder (VAE) which learns a distribution for our input data rather than a discrete latent code. Our proposed VAE consists of two sub-networks: an encoder network and a decoder network. The encoder network (Enc) learns to produce a latent representation, $\mathbf{z}$, from an input sample, $\mathbf{x}$, by

$$\mathbf{z} \sim \text{Enc}(\mathbf{x}) = q(\mathbf{z}|\mathbf{x}). \tag{1}$$

Analogously, the decoder network (Dec), which follows the structure of the generator in [26], learns the mapping from $\mathbf{z}$ back to the data space by

$$\tilde{\mathbf{x}} \sim \text{Dec}(\mathbf{z}) = p(\mathbf{x}|\mathbf{z}). \tag{2}$$

Additionally, we regularize the encoder network by imposing a unit Gaussian prior on the latent distribution $p(\mathbf{z})$, such that $\mathbf{z} \sim \mathcal{N}(1, 0)$.

The decoder network of our VAE follows the design of the ProGAN [26] generator network and is made up of a fully connected bottleneck layer followed by multiple convolutional modules, each of which consists of a nearest neighbor upsampling layer, followed by a convolutional layer, leaky rectified linear unit (LReLU) activation function and a pixel-wise feature vector normalization stage. The pixel-wise normalization layer was added by Karras et al. [26] in order to improve training stability as GANs are inherently unstable and suffer from mode collapse where the generated data collapses to a single sample and thus loses diversity. For full details of the workings of each of the layers in the decoder, we refer the reader to [26] for brevity.

Our encoder network is created by mirroring the structure of the decoder network, and replacing the upsampling operations with an average-pooling downsampling operation. Additionally, a fully connected layer with linear activation is added to the top convolutional layer in order to create the

bottleneck required for mapping 2D features to a latent distribution. This mirrored structure has the benefit of simplifying the training procedure (as will become evident in Section 2.2). The structure of our generator VAE is shown in detail in Table 1.

Table 1. A detailed overview of the encoder and decoder network structure.

Encoder	**Act.**	**Output Shape**
Conv 1×1	LReLU	$N \times 1 \times 128 \times 128$
Conv 3×3	LReLU	$N \times 128 \times 128 \times 128$
Conv 3×3	LReLU	$N \times 256 \times 128 \times 128$
Downsample	-	$N \times 256 \times 64 \times 64$
Conv 3×3	LReLU	$N \times 256 \times 64 \times 64$
Conv 3×3	LReLU	$N \times 512 \times 64 \times 64$
Downsample	-	$N \times 512 \times 32 \times 32$
Conv 3×3	LReLU	$N \times 512 \times 32 \times 32$
Conv 3×3	LReLU	$N \times 512 \times 32 \times 32$
Downsample	-	$N \times 512 \times 16 \times 16$
Conv 3×3	LReLU	$N \times 512 \times 16 \times 16$
Conv 3×3	LReLU	$N \times 512 \times 16 \times 16$
Downsample	-	$N \times 512 \times 8 \times 8$
Conv 3×3	LReLU	$N \times 512 \times 8 \times 8$
Conv 3×3	LReLU	$N \times 512 \times 8 \times 8$
Downsample	-	$N \times 512 \times 4 \times 4$
Conv 3×3	LReLU	$N \times 512 \times 4 \times 4$
Conv 4×4	LReLU	$N \times 512 \times 1 \times 1$
Fully Connected	Linear	$N \times 1024 \times 1 \times 1$
Mean	Split	$N \times 512 \times 1 \times 1$
Std. Deviation		$N \times 512 \times 1 \times 1$
Decoder	**Act.**	**Output Shape**
Latent Vector	-	$N \times 512 \times 1 \times 1$
Conv 4×4	LReLU	$N \times 512 \times 4 \times 4$
Conv 3×3	LReLU	$N \times 512 \times 4 \times 4$
Upsample	-	$N \times 512 \times 8 \times 8$
Conv 3×3	LReLU	$N \times 512 \times 8 \times 8$
Conv 3×3	LReLU	$N \times 512 \times 8 \times 8$
Upsample	-	$N \times 512 \times 16 \times 16$
Conv 3×3	LReLU	$N \times 512 \times 16 \times 16$
Conv 3×3	LReLU	$N \times 512 \times 16 \times 16$
Upsample	-	$N \times 512 \times 32 \times 32$
Conv 3×3	LReLU	$N \times 512 \times 32 \times 32$
Conv 3×3	LReLU	$N \times 512 \times 32 \times 32$
Upsample	-	$N \times 512 \times 64 \times 64$
Conv 3×3	LReLU	$N \times 256 \times 64 \times 64$
Conv 3×3	LReLU	$N \times 256 \times 64 \times 64$
Upsample	-	$N \times 256 \times 128 \times 128$
Conv 3×3	LReLU	$N \times 128 \times 128 \times 128$
Conv 3×3	LReLU	$N \times 128 \times 128 \times 128$
Conv 1×1	Linear	$N \times 1 \times 128 \times 128$

Following the standard procedure for VAEs, we can define the loss for our proposed generator as the reconstruction error and a prior regularization term, such that $\mathcal{L}_{\text{VAE}} = \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{prior}}$. However, using pixel-wise reconstruction errors with images often leads to blurry and noisy results [27]. Thus, we follow the approach proposed by Larsen et al. [28]. By exploiting the fact that our decoder network can be viewed as the generator network of a standard GAN, we incorporate the standard GAN loss into our VAE loss [29]. In doing so, we combine the advantages of the high-quality generative nature

of GANs with the VAEs ability to encode data into an inherently probabilistic latent space $\mathbf{z}$. Our loss terms can now be defined as $\mathcal{L}_{\text{VAE}} = \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{prior}} + \mathcal{L}_{\text{GAN}}$, with:

$$\mathcal{L}_{\text{recon}} = \|\mathbf{x} - \tilde{\mathbf{x}}\|, \tag{3}$$

$$\mathcal{L}_{\text{prior}} = D_{\text{KL}}\left(q(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})\right), \tag{4}$$

$$\mathcal{L}_{\text{GAN}} = \log\left(\text{Dis}(\mathbf{x})\right) + \log\left(1 - \text{Dis}(\tilde{\mathbf{x}})\right) + \log\left(1 - \text{Dis}(\text{Dec}(\mathbf{z}_p))\right), \tag{5}$$

where $\mathbf{z}_p$ is a sample from our prior $p(\mathbf{z})$, Dis is our discriminator, and D_{KL} is the Kullback–Leibler divergence.

2.1.2. Discriminator

The discriminator network for our proposed hard negative GAN is designed to be able to distinguish between real SAR image patches and generated SAR-like image patches. The discriminator accepts grayscale images in the form of either the original SAR image patch $\mathbf{x}$ or the generated patch $\tilde{\mathbf{x}} = \text{Dec}(\text{Enc}(\mathbf{x}))$ as input and outputs a scalar score representing how real the images are. This approach is slightly different to standard GANs where the output of the discriminator is a probability [29].

Apart from the bottleneck layers, our discriminator follows the same structure as our encoder network described in Section 2.1.1. The top layers of the discriminator consist of two fully connected layers which reduce the output of the convolutional layers to a single scalar. A linear activation function is then applied to this value in order to obtain a scalar score of image 'realness'. An additional difference between the encoder and discriminator architecture is the inclusion of a mini-batch standard deviation layer which adds an additional feature map to one of the last layers of the discriminator. Karras et al. [26] added this layer in order to increase variation in the network. The full details of the discriminator are described in Table 2.

Table 2. A layer-wise overview of the discriminator network structure.

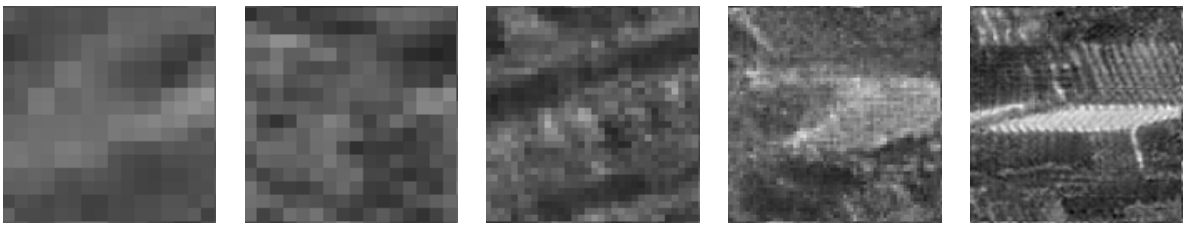
Discriminator	Act.	Output Shape
Conv 1 × 1	LReLU	N × 1 × 128 × 128
Conv 3 × 3	LReLU	N × 128 × 128 × 128
Conv 3 × 3	LReLU	N × 256 × 128 × 128
Downsample	-	N × 256 × 64 × 64
Conv 3 × 3	LReLU	N × 256 × 64 × 64
Conv 3 × 3	LReLU	N × 512 × 64 × 64
Downsample	-	N × 512 × 32 × 32
Conv 3 × 3	LReLU	N × 512 × 32 × 32
Conv 3 × 3	LReLU	N × 512 × 32 × 32
Downsample	-	N × 512 × 16 × 16
Conv 3 × 3	LReLU	N × 512 × 16 × 16
Conv 3 × 3	LReLU	N × 512 × 16 × 16
Downsample	-	N × 512 × 8 × 8
Conv 3 × 3	LReLU	N × 512 × 8 × 8
Conv 3 × 3	LReLU	N × 512 × 8 × 8
Downsample	-	N × 512 × 4 × 4
Mini-batch Std. Dev.	-	N × 513 × 4 × 4
Conv 3 × 3	LReLU	N × 512 × 4 × 4
Conv 4 × 4	LReLU	N × 512 × 1 × 1
Fully Connected	Linear	N × 1 × 1 × 1

2.2. Training Procedure

Our training procedure combines the training procedure of [26] with the dual GAN and VAE loss definitions of [28], as described in Section 2.1.1.

2.2.1. Progressive Growing

We initialize our networks to start the training process with an image resolution of 4×4 pixels. We then gradually increase this resolution by a factor of 2 after a specified number of training iterations. In order to prevent jolting the system when a new layer is added, we closely follow the process described in [26]. Adding new layers to the networks in a smooth manner consists of a two stage approach. During the *transition phase*, we treat layers which operate on the higher resolution as a residual block whose weight α increases linearly from 0 to 1 over a set number of training iterations. Additionally, we interpolate between two resolutions of the input image, in a similar manner to how the generator combines the new and old resolution. The second stage is the *stabilization phase*, whereby the networks are trained for a specific number of iterations before the resolution is doubled again. All of the networks are grown in this manner from a low resolution of 4×4 pixels to our final resolution of 128×128 pixels. Using networks that have a similar structure simplifies the process of managing multi-resolution data and eases the complexity involved in transitioning between layers. An example of the networks training progression is depicted in Figure 3.



(**a**) (**b**) (**c**) (**d**) (**e**)

Figure 3. An example of progressively grown images taken at increasing image resolutions during the training processes. (**a**–**e**) shows the resolution growth from 8×8 pixels up to 128×128 pixels with the resolution doubling at each stage.

This progressive growing approach drastically speeds up training of the GAN and improves the overall training stability as the network only needs to learn small transformations between the previous and next layers.

2.2.2. WGAN-GP Loss

While the proposed training approach greatly improves stability and reduces the chances of mode collapse, it does not solve the issue of large gradients which occur when generating high resolution images. This gradient issue occurs due to the fact that fake images are significantly easier to distinguish at high resolutions and thus large gradients propagate from the discriminator.

In order to prevent this gradient problem, and further increase the stability of training, we make use of the improved Wasserstein GAN loss with gradient penalty (WGAN-GP) [30]. This loss function is used to train the discriminator network, as well as to replace the standard GAN loss $\mathcal{L}_{\text{GAN}}$ which is used to train our generator network (Equation (5)). Thus, our new loss functions can be defined as

$$\mathcal{L}_{\text{Dis}} = \underbrace{\mathop{\mathbb{E}}_{\mathbf{x}\sim\mathbb{P}_{\text{real}}}[\text{Dis}(\mathbf{x})] - \mathop{\mathbb{E}}_{\tilde{\mathbf{x}}\sim p(\mathbf{x}|\mathbf{z})}[\text{Dis}(\tilde{\mathbf{x}})]}_{\text{Original Critic Loss}} + \underbrace{\lambda \mathop{\mathbb{E}}_{\hat{\mathbf{x}}\sim\mathbb{P}_{\hat{\mathbf{x}}}}[(\|\nabla_{\hat{\mathbf{x}}}\text{Dis}(\hat{\mathbf{x}})\| - 1)^2]}_{\text{Gradient Penalty}}, \tag{6}$$

$$\mathcal{L}_{\text{VAE}} = \underbrace{- \mathop{\mathbb{E}}_{\tilde{\mathbf{x}}\sim p(\mathbf{x}|\mathbf{z})}[\text{Dis}(\tilde{\mathbf{x}})] - \mathop{\mathbb{E}}_{\mathbf{z}_p\sim p(\mathbf{z})}[\text{Dis}(\text{Dec}(\mathbf{z}_p))]}_{\text{Original Generator Loss}} + \underbrace{\mathop{\mathbb{E}}_{q(\mathbf{z}|\mathbf{x})}[\log(\frac{q(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})})]}_{\text{KL-Divergence}} + \underbrace{\gamma \sum_{i}^{N} \|\mathbf{x} - \text{Dec}(\text{Enc}(\mathbf{x}))\|}_{\text{Reconstruction Error}}, \tag{7}$$

where $\mathbb{P}_{\hat{\mathbf{x}}}$ is implicitly defined as sampling uniformly between pairs of points sampled from the data distribution $\mathbb{P}_{\text{real}}$ and the decoder distribution $p(\mathbf{x}|\mathbf{z})$ and λ and γ are weighting coefficients which are set as hyper-parameters.

2.2.3. Additional Training Details

Using the losses defined in Equations (6) and (7), we train our network using the *Adam* gradient descent with the momentum approach. The learning rate is initialized to 0.001 for the decoder and discriminator networks and 0.0005 for the encoder network. Additionally, the moving average filter parameters for the Adam optimizer are set to $\beta_1 = 0$, $\beta_2 = 0.99$ for all networks. Training data is fed to the network using an initial mini-batch size of 128 samples. However, this number is decreased to 16 samples as the resolution increases. All three sub-networks are grown simultaneously with a *transition rate* and *stabilization rate* of 60,000 images or approximately 10 epochs each.

Additionally, as per the findings of [28], we do not propagate the error signals from the $\mathcal{L}_{\text{GAN}}$ losses to the encoder network. Furthermore, as the decoder network receives error signals from both $\mathcal{L}_{\text{GAN}}$ and $\mathcal{L}_{\text{recon}}$, we set the weighting term $\delta = 0.6$ to add a slight preference to the network's ability to reconstruct the input over its ability to fool the discriminator. We also include reconstructed samples $\tilde{\mathbf{x}}$, as well as samples from our prior distribution $p(\mathbf{z})$ in our GAN objective, as this was found to produce better results than using only samples from the prior distribution. The inner training loop is detailed in Algorithm 1.

Algorithm 1: Training our Hard Negative GAN

$\boldsymbol{\Theta}_{\text{Enc}}, \boldsymbol{\Theta}_{\text{Dec}}, \boldsymbol{\Theta}_{\text{Dis}} \leftarrow$ Glorot uniform initialization
repeat
  $\mathbf{X} \leftarrow$ random mini-batch from dataset
  $\mathbf{Z} \leftarrow \texttt{Enc}(\mathbf{X})$
  $\mathcal{L}_{\text{prior}} \leftarrow D_{\text{KL}}\left(q(\mathbf{Z}, \mathbf{X}) | p(\mathbf{Z})\right)$
  $\tilde{\mathbf{X}} \leftarrow \texttt{Dec}(\mathbf{Z})$
  $\mathcal{L}_{\text{recon}} \leftarrow \left\|\mathbf{X} - \tilde{\mathbf{X}}\right\|$
  $\mathbf{Z}_p \leftarrow$ samples from prior $\mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{X}_p \leftarrow \texttt{Dec}(\mathbf{Z}_p)$
  $\mathcal{L}_{\text{WGAN}} \leftarrow \texttt{Dis}(\mathbf{X}) - \texttt{Dis}(\tilde{\mathbf{X}})$ See Equation (6)
  $\mathcal{L}_{\text{Dec}} \leftarrow \texttt{Dec}(\tilde{\mathbf{X}}) + \texttt{Dis}(\texttt{Dec}(\mathbf{Z}_p))$ See Equation (7)
  Update network according to gradients
  $\boldsymbol{\Theta}_{\text{Dis}} \overset{+}{\leftarrow} -\nabla_{\boldsymbol{\Theta}_{\text{Dis}}}(\mathcal{L}_{\text{WGAN}} + \lambda \mathcal{L}_{\text{GP}})$
  $\boldsymbol{\Theta}_{\text{Enc}} \overset{+}{\leftarrow} -\nabla_{\boldsymbol{\Theta}_{\text{Enc}}}(\mathcal{L}_{\text{prior}} + \mathcal{L}_{\text{recon}})$
  $\boldsymbol{\Theta}_{\text{Dec}} \overset{+}{\leftarrow} -\nabla_{\boldsymbol{\Theta}_{\text{Dec}}}(\gamma \mathcal{L}_{\text{recon}} - \mathcal{L}_{\text{Dec}})$
until *convergence*;

2.3. Generating Hard Negative Samples

In order to generate hard negative samples, we train our proposed GAN on 6629 SAR images from the training data which are used to train the SAR-optical matching network. In doing so, the encoder network learns the latent distribution of our training data and the decoder network learns to reconstruct the input data from this distribution. After training, the discriminator network is discarded and the VAE is used to generate hard negative SAR samples. As the latent space is continuous and follows a unit normal distribution, we can create novel, yet similar SAR patches by sampling the latent distribution near to the location of the encoded input image. This process is depicted in Figure 4.

These generated, SAR-like images are then used as hard negative samples in the SAR-optical matching training dataset. This is done by creating a non-corresponding patch-pair which consists of the generated SAR image, and the optical image which corresponds to the original SAR image, which was used to generate the hard-negative. Some examples of the appended dataset can be seen in Figure 5.

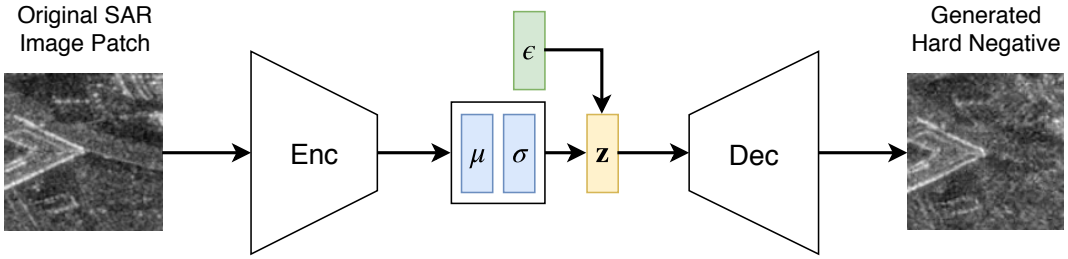
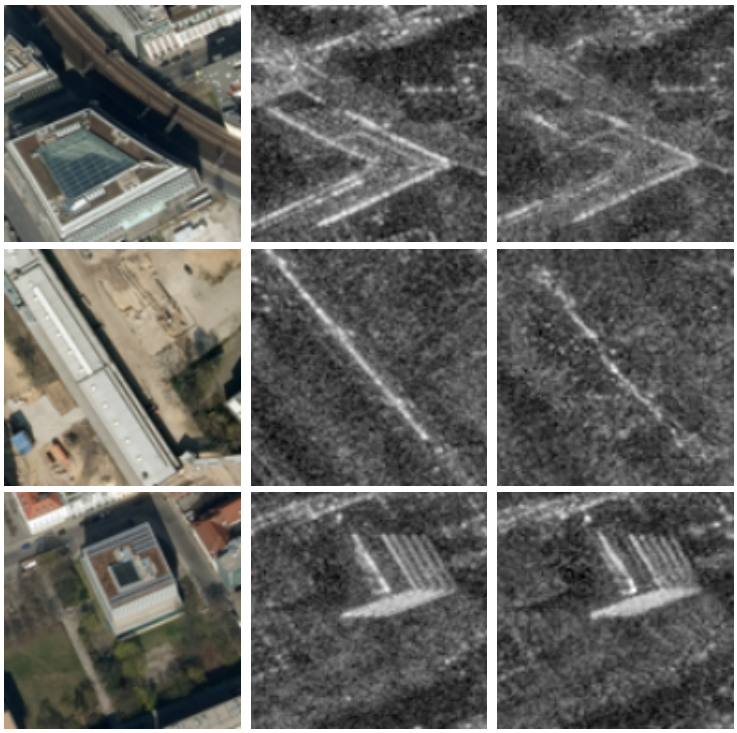


Figure 4. The inference network used to generate hard negative samples. The latent code **z** used to generate patches is created by sampling the latent distribution near to the original image. To keep the network end-to-end differentiable, this sampling is done via a re-parameterization trick using $\epsilon \sim \mathcal{N}(1, 0)$ to add randomness.



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

Figure 5. Using the proposed generative framework, we are able to generate SAR-like image patches (**c**) that can then be combined in conjunction with the original SAR patch (**b**), and a corresponding optical patch (**a**) in order to create a training dataset containing hard-negative samples.

3. Experiments and Results

In this section, we describe our experimental procedure and present results with respect to our network's ability to generate realistic SAR patches, and the suitability of these patches as hard negative samples for training the SAR-optical matching network of [14].

3.1. Dataset

We train our proposed hard negative GAN and the SAR-optical matching network of [14] on a dataset of corresponding unfiltered TerraSAR-X and UltraCam image patch pairs [17]. The patch pairs are generated from imagery taken of a study area in Berlin, Germany, which is depicted in Figure 6.

Figure 6. The common region of interest, in Berlin, Germany from which TerraSAR-X and UltraCam image patches were cut to generate the the SARptical dataset [17].

The dataset is deterministically split into a training, testing and validation set using the *cutting-cake* method proposed in [14]. Using this deterministically split dataset, we reduce the chances of the training and testing datasets having too similar distributions. Our datasets consist of 6629 (75%), 1327 (15%), and 885 (10%) corresponding image pairs for the training, testing and validation sets, respectively.

The non-corresponding pairs for the testing and validation datasets are created by assigning a randomly selected SAR image patch to each optical image. In doing so, we ensure that all experiments are subject to the same testing and validation datasets, and that our datasets are balanced in terms of corresponding and non-corresponding pairs.

The non-corresponding pairs for our training dataset are assigned according to the requirements of each experiment, in order to evaluate the success of our method.

The optical data was converted to gray-scale and all data were normalized to a radiometric range of $[0;1]$ and then standardized by subtraction of their means [14,19]. Furthermore, we make use of pair-wise data augmentation steps which include rotation, horizontal flipping, and translation.

3.2. Qualitative Evaluation of Generated Negative Samples

Measuring the quality of generated images is a challenging task, especially in the case of high resolution data [31]. Thus, we resort to a visual qualitative assessment of the generated hard negative SAR patches. These results can be seen in Figure 7.

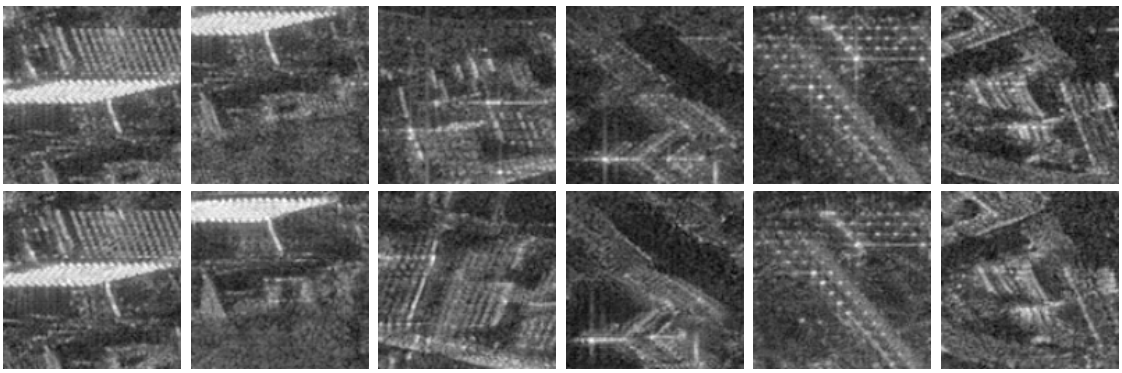


Figure 7. A selection of generated hard-negative samples (**bottom row**) and the corresponding training TerraSAR-X patch (**top row**). It can be seen that the generated patches have strong SAR-like features, which resemble those of the original patch, and are difficult to distinguish from the original patches.

3.3. Matching SAR and Optical Images

We apply our methods to the SAR-optical matching network proposed in [14]. This network has a pseudo-Siamese architecture which learns modal specific features for SAR and optical images in parallel. It then combines these features through a data fusion layer in order to obtain a prediction of whether the two patches match based on the content of the center pixel. The network architecture can be seen in Figure 8.

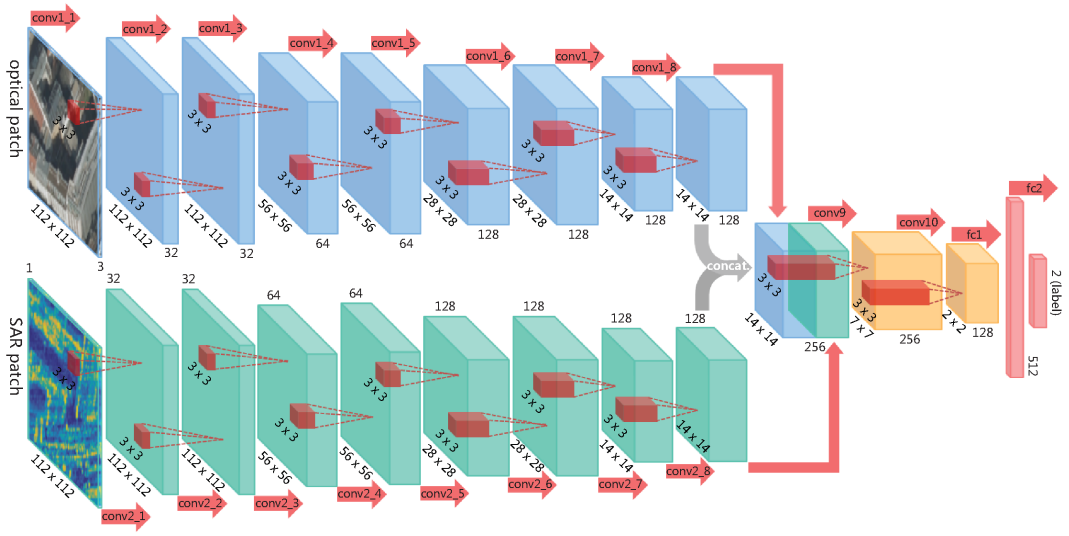


Figure 8. The pseudo-Siamese convolutional matching network proposed in [14]. The network attempts to predict the probability that the given input pair are corresponding in terms of the alignment of the center pixel in each patch. (© 2018 IEEE).

For all of our experiments, we train the network using the Adam optimizer with a learning rate of $\alpha_{\text{lr}} = 0.00005$. All of the networks are trained using early stopping based on the validation accuracy for a maximum period of 20 epochs.

3.4. Effect of a Hard Negative Inclusion Method

In order to evaluate our approach, we need to include our generated non-corresponding pairs into the existing training dataset. As we were unsure of the best approach for training the matching network with generated hard negatives, we evaluated two different training approaches with two dataset inclusion methods. These four approaches are defined below:

1. Fine-tuning with generated hard negatives,
2. Fine-tuning with concatenated dataset,
3. Training from scratch with generated hard negatives,
4. Training from scratch with concatenated dataset.

The generated hard negative dataset consists only of the original corresponding patch-pairs and their respective hard-negative patch-pairs, which were created as described in Section 2.3. In order to create the concatenated dataset, we combined the generated hard-negative dataset with the original training dataset in order to form a final dataset with both generated hard-negatives and randomly assigned hard-negatives for each of the corresponding patch-pairs. In order to keep the positive and negative classes balanced, we included each corresponding patch-pair twice.

To allow us to fine-tune the matching network, we first pre-trained it using the original training dataset which consists of randomly assigned negative patch-pairs for 30 epochs. This network was then fine-tuned using a lower learning rate of $\alpha = 0.000008$ and early stopping.

We evaluated the trained networks performance using the receiver–operator characteristic (ROC) in order to determine which approach leads to the most favourable results. The ROC curves for each of the four approaches are depicted in Figure 9.

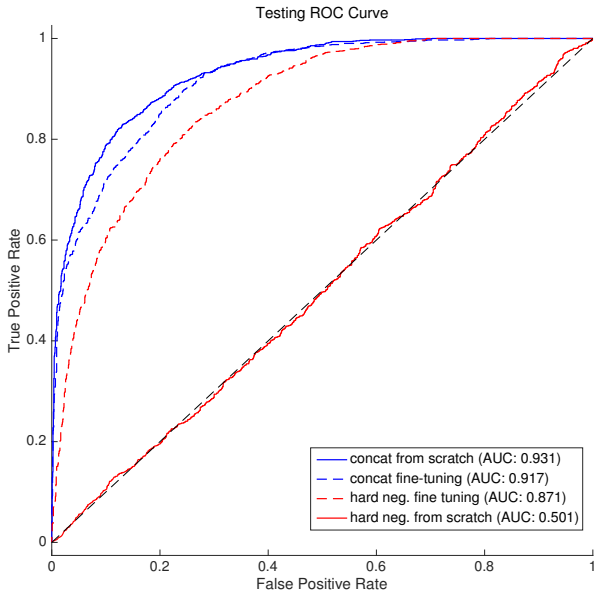


Figure 9. The receive operator characteristic (ROC) curves for various approaches for using generated hard-negatives in the training of a SAR-optical matching network. From these various experiments, it can be seen that including the generated hard-negatives into the original dataset leads to a matching network with better performance than using only generated negative samples for training.

A simple measure of the network's performance as a binary classifier is the area under the curve (AUC) of the ROC curve. From Figure 9, one can see that training the matching network used from scratch using a combined dataset provides the best performance. Thus, we select this approach as the proposed method of hard-negative inclusion, and will use it in further experiments.

3.5. Comparison to Existing Approaches

We compare the performance of the matching network trained using our proposed method to three alternative approaches, namely, random negative assignment, traditional hard-negative mining, and nearest neighbor assignment. These approaches are further detailed below:

Random negative assignment creates non-corresponding negative patch pairs by randomly selecting a SAR image patch from the patch pool and assigning it to a randomly selected optical patch. This random selection is done in a non-replacement manner such that every randomly created patch pair is unique and non-corresponding. This method is the most computationally efficient method for negative pair assignment, but it makes strong assumptions about the 'closeness' of the data.

Traditional hard-negative mining [32], starts training with random negative assignment and then iteratively updates the set of non-corresponding patch pairs at the end of each training epoch. The updates are performed by tracking the classification score of each patch-pair during training. The patch-pairs which were most severely mis-classified (non-corresponding pairs which were classified as corresponding with a high probability) are then explicitly labelled as negative pairs and added to the dataset, the remaining negatives pairs are reinitialized using random negative assignment. This process is computationally expensive and degrades to continuous random assignment when the false positive rate is low and/or the training dataset is small.

Nearest neighbor assignment [33], is a boostrapping method for hard-negative mining and is performed prior to training. For each positive patch pair, a non-corresponding pair is created by selecting the nearest neighbor SAR image from the training set. The nearest neighbor is defined as the

image patch in the training dataset that has the greatest similarity to the positive image patch. In our case, we make use of the normalized cross correlation (NCC) score to determine which SAR images are most similar to each other. We then generate a non-corresponding patch pair using the SAR image with the greatest NCC score when compared to the positive = pair SAR image.

A detailed comparison of the results is presented in the form of an ROC comparison plot (see Figure 10). From the ROC plot, we can see that our approach provides a higher accuracy under the constraint of a low false positive rate. We further provide a detailed account of the precision and recall of the various approaches, as well as the respective accuracies when the decision boundary is tuned (on the validation set after training) to provide a maximum FPR of 5% or a maximum overall accuracy (see Table 3). From these results, our method is shown to boost the performance of the matching network on almost all fronts.

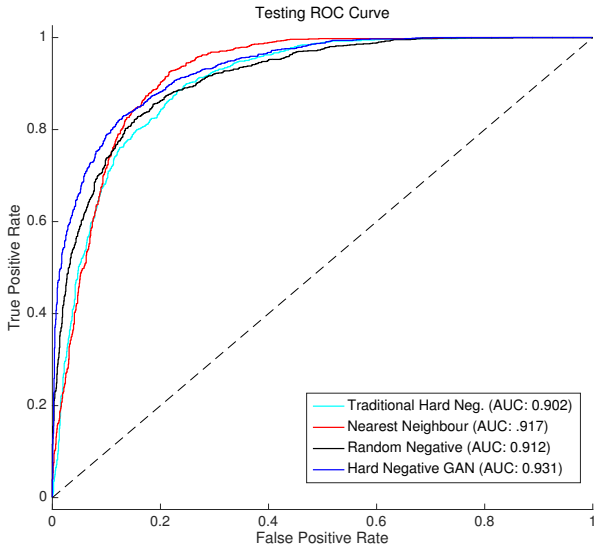


Figure 10. Comparison of training results, in the form of ROC curves, for the matching network performance on a test set when trained using randomly assigned negative pairs to three hard-negatives mining approaches. It can be seen that our proposed approach has a steeper onset than the other approaches, thus indicating better performance during matching.

Table 3. Details of SAR-optical matching results under the application of various hard-negative training strategies and at different false positive rates (FPR).

Method	Precision	Recall	Acc. (5% FPR)	Max Acc.	Max Acc. FPR
Random	**0.83**	0.84	0.76	0.83	0.16
Nearest Neighbour	0.77	**0.96**	0.70	0.85	0.21
Traditional Hard Neg.	0.79	0.89	0.72	0.83	0.19
Proposed Approach	**0.83**	0.87	**0.81**	**0.86**	**0.13**

4. Discussion

Generally, the results presented in Section 3 show that we are able to generate high quality hard-negative samples, and to use them to successfully train a SAR-optical matching network. The results further indicate that following this approach leads to a significant improvement in the overall performance and discriminability of the matching network, without the need for additional training data. In this section, we will further explore these results to gain a deeper understanding of the mechanisms at play.

4.1. Generative Ability

Considering the generated images presented in Figure 7, it is clear that our proposed generative framework is able to learn a diverse latent representation of the training dataset. By sampling this

latent distribution, we are able to generate realistic VHR SAR-like images. The generated images are largely indistinguishable from the original TerraSAR-X patches and depict many SAR-like features such as layover, speckle, and radar shadow. Additionally, and arguably most importantly for our application, the images generated by sampling the posterior are visually similar to the original images but still contain novel components.

4.2. Effects of Data Inclusion Approach

As Figure 9 clearly indicates, the method of incorporating the generated hard negatives into the training procedure of the matching network plays a large role in the effectiveness of the approach. Training the network using both randomly assigned and generated non-corresponding pairs produced significantly better results than using only the generated samples. This is likely due to the generated samples adding sufficient variability to the dataset to act as independent datapoints, thus essentially increasing the size of the training dataset. This theory is backed up by the result of training from scratch using only the generated images as negative samples. In this case, it becomes clear that the validation and training dataset have a larger disparity in their distributions. Thus, apart from increasing the dataset size, training using both distributions likely has a regularization effect on the training of the network.

Additionally, Figure 9 shows that training from scratch is only a better approach when we include non-correspondences created using real data, even if these are just created using a random assignment approach. This is evident, from the case of training, that the matching network from scratch using only generated negative features where the network fails to learn any discriminative boundary that is suitable for matching real data. This is likely a consequence of the generated manifold being a Gaussian approximation to the original manifold and thus the distributions could have disjoint supports, and is subject to future investigation.

4.3. SAR-Optical Matching Performance

The comparison of our proposed approach to three alternative training methodologies shows promise for the use of generative hard-negative mining in improving matching performance in data sparse applications. As Figure 10 and Table 3 clearly indicate, training with generative hard-negative mining significantly improves the discriminative power and accuracy of the matching network when evaluated on an independent test dataset. Using this approach, we were able to train the matching network to achieve an accuracy exceeding 80% when the false positive rate is fixed to 5%. Additionally, the matching network was able to achieve an overall higher accuracy with a 3% point reduction in false positives.

The results of the traditional hard-negative mining agree with the literature, which states that the technique fails to add benefits if the dataset is significantly not large enough, as it effectively falls back to a random negative procedure [16,32]. Overall, this approach fails to improve any aspect of the original matching network, and in many ways preforms as a combination of the worst aspects of the other approaches, achieving an overall accuracy that matches that of the randomly assigned negatives, but with a worse false positive rate.

An interesting result is that of the nearest neighbor hard-negatives. These negative patches are assigned according to which image in the training dataset is the closest to the input image in a normalized cross-correlation (NCC) sense. As NCC is often used as a signal-based measure for multi-modal (including SAR-optical) image matching, it would appear to be a good choice for selecting hard-negatives. However, this approach produces the worst overall accuracy and false positive performance. It is suspected that the NCC hard-negatives cause the non-corresponding pairs to be too similar to the corresponding pairs, thus creating a matching problem which is too complex for the given network to resolve. This suspicion is further backed up by the high recall but low precision, which indicates that the network has become biased towards predicting patch pairs as corresponding (see Table 3).

4.4. Comments on Computational Overhead

Although our approach leads to the best performance of the SAR-optical matching network, this comes at the cost of a large computational overhead. This is caused by the fact that we need to train a relatively large and complex generative network. However, we can compute the dataset of negative samples prior to training the matching network, which allows for swapping out the online requirements of RAM and an additional GPU for additional storage capacity. In doing so, we reduce the computational burden of our approach to a once-off cost per dataset. Training this generative network for our small training dataset took 96 hours on a single NVidia GTX 1080 GPU. During training, the matching network using our offline approach took around 20 min on the same hardware.

On the other hand, traditional hard-negative mining directly impacts the computational cost of training the matching network. As it is performed on-the-fly, the computational burden persists across experiments and training operations. In the case of our investigation, the training time of the matching network increased to 25 min; however, this time increase grows along with the dataset size. Additionally, the training time memory requirements for the network increase as we need to keep a history of predicted labels for each item in the dataset so that items can be replaced by better hard-negative samples.

Thus, the added upfront computational expense of our approach may work better in environments with limited computational resources but sufficient storage capacity.

5. Conclusions

With this paper, we have proposed a generative framework for hard-negative mining that can be used in data sparse image matching applications to improve the discriminability and accuracy of the matching network. By combining the strong latent space encoding features of a variational autoencoder with the high quality generative capabilities of generative adversarial networks, we are able to produce realistic SAR-like image patches in a conditional manner. In doing so, we are able to produce a structurally similar, but novel SAR patch for each SAR image in our training dataset. We can then combine these SAR and SAR-like images with a corresponding optical image in order to create a balanced dataset of corresponding and non-corresponding patch pairs that can be used for training SAR-optical matching networks.

By applying this generative hard-negative approach to the existing SAR-optical matching network proposed in [14], we were able to confirm the capabilities of our approach in improving matching accuracy and reducing a false positive rate when tested on an independent dataset. Within the scope of sparse training data, our proposed method shows a significant improvement in matching accuracy at low FPRs and a small improvement in overall accuracy (but with a significant improvement in FPR) when compared to two commonly applied hard negative mining techniques.

Our generative hard-negative mining framework has applicability outside of the realm of SAR-optical matching. It is believed that this approach to hard-negative mining can be applied to many other problems that suffer from similar data constraints, both within and outside of remote sensing.

Author Contributions: Conceptualization, L.H.H. and M.S.; Methodology, L.H.H.; Software, L.H.H.; Data Curation and Investigation, L.H.H. and M.S.; Writing, L.H.H. and M.S.; Supervision, M.S. and X.Z.; Project Administration, M.S.; Funding Acquisition, M.S.; Resources: M.S. and X.Z.

Funding: This work was supported by the German Research Foundation (DFG), grant SCHM 3322/1-1, the Helmholtz Association under the framework of the Young Investigators Group SiPEO (VH-NG-1018, www.sipeo.bgu.tum.de), and the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. ERC-2016-StG-714087, Acronym: *So2Sat*).

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

References

1. Schmitt, M.; Zhu, X.X. Data Fusion and Remote Sensing—An Ever-Growing Relationship. *IEEE Geosci. Remote Sens. Mag.* **2016**, *4*, 6–23. [CrossRef]
2. Schmitt, M.; Tupin, F.; Zhu, X.X. Fusion of SAR and optical remote sensing data—Challenges and recent trends. In Proceedings of the 2017 IEEE International Geoscience and Remote Sensing Symposium, Fort Worth, TX, USA, 23–28 July 2017; pp. 5458–5461.
3. Ye, Y.; Shan, J.; Bruzzone, L.; Shen, L. Robust Registration of Multimodal Remote Sensing Images Based on Structural Similarity. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 2941–2958. [CrossRef]
4. Li, J.; Hu, Q.; Ai, M. RIFT: Multi-modal Image Matching Based on Radiation-invariant Feature Transform. *arXiv* **2018**, arXiv:1804.09493.
5. Qiu, C.; Schmitt, M.; Zhu, X.X. Towards automatic SAR-optical stereogrammetry over urban areas using very high resolution imagery. *ISPRS J. Photogramm. Remote Sens.* **2018**, *138*, 218–231. [CrossRef] [PubMed]
6. Palubinskas, G.; Reinartz, P.; Bamler, R. Image acquisition geometry analysis for the fusion of optical and radar remote sensing data. *Int. J. Image Data Fusion* **2010**, *1*, 271–282. [CrossRef]
7. Balntas, V.; Johns, E.; Tang, L.; Mikolajczyk, K. PN-Net: Conjoined triple deep network for learning local image descriptors. *arXiv* **2016**, arXiv:1601.05030.
8. Yi, K.M.; Trulls, E.; Lepetit, V.; Fua, P. Lift: Learned invariant feature transform. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2016; pp. 467–483.
9. Zagoruyko, S.; Komodakis, N. Deep compare: A study on using convolutional neural networks to compare image patches. *Comput. Vis. Image Underst.* **2017**, *164*, 38–55. [CrossRef]
10. Han, X.; Leung, T.; Jia, Y.; Sukthankar, R.; Berg, A.C. Matchnet: Unifying feature and metric learning for patch-based matching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3279–3286.
11. Zhu, X.X.; Tuia, D.; Mou, L.; Xia, G.S.; Zhang, L.; Xu, F.; Fraundorfer, F. Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources. *IEEE Geosci. Remote Sens. Mag.* **2017**, *5*, 8–36. [CrossRef]
12. Merkle, N.; Luo, W.; Auer, S.; Müller, R.; Urtasun, R. Exploiting Deep Matching and SAR Data for the Geo-Localization Accuracy Improvement of Optical Satellite Images. *Remote Sens.* **2017**, *9*, 586. [CrossRef]
13. Mou, L.; Schmitt, M.; Wang, Y.; Zhu, X. A CNN for the Identification of Corresponding Patches in SAR and Optical Imagery of Urban Scenes. In Proceedings of the 2017 Joint Urban Remote Sensing Event (JURSE), Dubai, UAE, 6–8 March 2017.
14. Hughes, L.H.; Schmitt, M.; Mou, L.; Wang, Y.; Zhu, X.X. Identifying Corresponding Patches in SAR and Optical Images With a Pseudo-Siamese CNN. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 784–788. [CrossRef]
15. Merkle, N.; Auer, S.; Müller, R.; Reinartz, P. Exploring the Potential of Conditional Adversarial Networks for Optical and SAR Image Matching. *IEEE J-STARS* **2018**, *11*, 1811–1820. [CrossRef]
16. Sung, K.K.; Poggio, T. Example-based learning for view-based human face detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1998**, *20*, 39–51. [CrossRef]
17. Wang, Y.; Zhu, X.; Zeisl, B.; Pollefeys, M. Fusing meter-resolution 4-D InSAR point clouds and optical images for semantic urban infrastructure monitoring. *IEEE Trans. Geosci. Remote Sens.* **2016**, *55*, 14–26. [CrossRef]
18. Auer, S.; Hornig, I.; Schmitt, M.; Reinartz, P. Simulation-based Interpretation and Alignment of High-Resolution Optical and SAR Images. *IEEE J-STARS* **2017**, *10*, 4779–4793. [CrossRef]
19. Wang, Y.; Zhu, X.X. The SARptical Dataset for Joint Analysis of SAR and Optical Image in Dense Urban Area. *arXiv* **2018**, arXiv:1801.07532.
20. Ding, J.; Chen, B.; Liu, H.; Huang, M. Convolutional neural network with data augmentation for SAR target recognition. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 364–368. [CrossRef]
21. Zheng, Z.; Zheng, L.; Yang, Y. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. *arXiv* **2017**, arXiv:1701.07717.
22. Ley, A.; d'Hondt, O.; Valade, S.; Hänsch, R.; Hellwich, O. Exploiting GAN-based SAR to optical image transcoding for improved classification via deep learning. In Proceedings of the 12th European Conference on Synthetic Aperture Radar, Aachen, Germany, 4–7 June 2018; pp. 396–401.
23. Wang, P.; Li, S.; Pan, R. Incorporating GAN for Negative Sampling in Knowledge Representation Learning. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.

24. Marmanis, D.; Yao, W.; Adam, F.; Datcu, M.; Reinartz, P.; Schindler, K.; Wegner, J.D.; Stilla, U. Artificial generation of big data for improving image classification: A generative adversarial network approach on SAR data. *arXiv* **2017**, arXiv:1711.02010.
25. Ao, D.; Dumitru, C.O.; Schwarz, G.; Datcu, M. Dialectical GAN for SAR Image Translation: From Sentinel-1 to TerraSAR-X. *arXiv* **2018**, arXiv:1807.07778.
26. Karras, T.; Aila, T.; Laine, S.; Lehtinen, J. Progressive Growing of GANs for Improved Quality, Stability, and Variation. *arXiv* **2018**, arXiv:1710.10196.
27. Khan, S.H.; Hayat, M.; Barnes, N. Adversarial Training of Variational Auto-encoders for High Fidelity Image Generation. *arXiv* **2018**, arXiv:1804.10323.
28. Larsen, A.B.L.; Sønderby, S.K.; Larochelle, H.; Winther, O. Autoencoding beyond pixels using a learned similarity metric. *arXiv* **2015**, arXiv:1512.09300.
29. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
30. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A.C. Improved training of Wasserstein GANs. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5769–5779.
31. Theis, L.; van den Oord, A.; Bethge, M. A note on the evaluation of generative models. In Proceedings of the 2016 International Conference on Learning Representations, San Juan, Puerto Rico, 2–4 May 2016.
32. Sung, K.K. *Learning and Example Selection for Object and Pattern Detection*; Computer Science and Artificial Intelligence Lab: Cambridge, MA, USA, 1996.
33. Jiang, F. SVM-Based Negative Data Mining to Binary Classification. Ph.D. Thesis, Georgia State University, Atlanta, GA, USA, 2006.