Translation of remote sensing images for the classification of unlabeled SAR data using Deep Convolutional Generative Adversarial Networks

MASTERTHESIS

for obtaining the academic degree
Master of Science (M. Sc.)

submitted at the
Faculty of Mathematics and Natural Sciences
Department of Physics
Humboldt University of Berlin

Julius Lange
born on October 6, 1993 in Berlin

Examiners:

1. *Prof. Dr. habil. Heinz-Wilhelm Hübers*
2. *Prof. Dr. -Ing. Morris Riedel*

submitted at: *February 26, 2019*

Contents

1 Introduction **1**

2 Remote Sensing **5**

2.1 Radar Imaging 5
2.2 Optical Imaging 8
2.3 Datasets 9

3 Deep Learning Methods **13**

3.1 Deep Neural Networks 13
3.2 Convolutional Neural Networks 16
3.3 Residual Neural Networks 17
3.4 Improving the Networks Performance 18
3.5 Generative Adversarial Networks 20

4 The Pix2Pix Framework for Image Translation **24**

4.1 Implementation Details 27
4.2 Baseline Tests 28
4.3 Translate Optical to SAR Images 32

5 An Artificial Dataset for SAR Classification **39**

5.1 Training the Classification Algorithms 40
5.2 The ESRGAN Framework for Image Super-Resolution 47
5.3 Testing the Classifiers on a different Dataset 52

6 Conclusion **54**

Chapter 1

Introduction

The term remote sensing summarizes the contact-free measurement and processing of data about the earth's surface [1]. Understanding the development of thematic classes of the Earth's surface is required by numerous present-day applications. The monitoring of the Earth's surface enables to address many challenging problems such as the surveillance of the growth of cities, desertification and the changing of bodies of water. Nowadays, the monitoring is mainly done by airplanes and satellites, which acquire huge amounts of data. Single satellites are capable of providing multiple terabytes of data per day[1]. Even with the improvements of high performance and distributed computing over the last years, processing and utilizing data this large is still a challenging problem and topic of current research [2]. Due to the availability of different sensors and instruments, a large variety of data is generated which additionally complicates the data processing.

Machine (deep) learning has been a growing field dealing with the processing of large amounts of data. One of its main objectives is to generate knowledge by optimizing an algorithm in contrast to use fixed rules predetermined by a human programmer. Optimizing an algorithm, also referred to as training, means that its performance is improved with knowledge accumulated from observed data. The capability of scaling machine learning algorithms is a challenge that has to be addressed with big data because more complex tasks require more complex machine learning models that have to be trained using more data [3][4].
Since the publication of the paper 'ImageNet Classification with Deep Convolutional Networks' by Alex Krizhevsky, Ilya Sutskever and Geoffrey Hinton in 2012, the field of deep learning has attracted more and more attention [3]. In their paper, the authors describe how they outperformed all other approaches at the ImageNet

---

[1] https://science.orf.at/stories/2777574/

Large-Scale Visual Recognition Challenge (ILSVRC) using a Convolutional Neural Network (CNN). Henceforward CNNs and other deep neural networks had great success in many computer vision tasks like classification [5][6][7], object detection [8][9] and image segmentation [10].

In remote sensing deep learning techniques are exploited to analyze data as well. Utilizing similar approaches in remote sensing and computer vision allows to adapt the concepts and improvements that occur in computer vision for enhanced results and insights in remote sensing tasks.

As deep learning is a sub-category of machine learning, the same subdivision into supervised and unsupervised learning can be carried out. Supervised learning refers to the training of an algorithm by using labeled data, i.e. the input data that needs to be processed on the one hand and a description of this data on the other hand. The objective is to find a general mapping between the input data and its label. In unsupervised learning the specification of the desired output given an input is not available.

One application example of supervised learning is image classification, where each image has to be mapped to the correct label. This task requires pairs of images and their assigned category to train the algorithm. Another example is image translation, where a computer program learns how to generate an image given another one, e.g. create a photo-realistic scene given a sketch of it. In this example, pairs of images are used for training.

Clustering algorithms on the contrary are usually trained in an unsupervised fashion to group data points and structure the data.

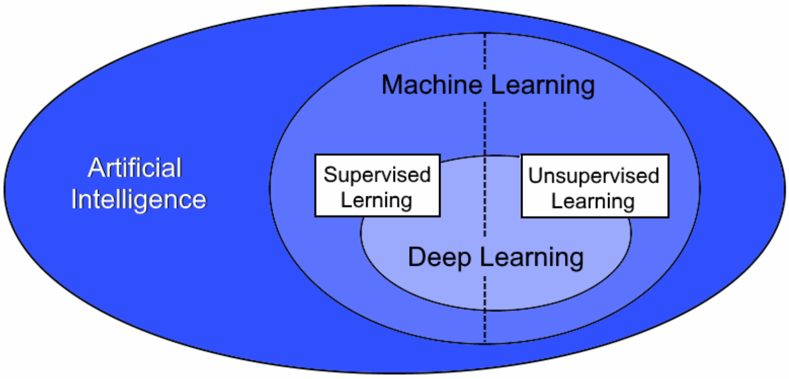


Figure 1.1: Deep Learning and its classification in the area of Artificial Intelligence

The aim of this thesis is to train a land cover classification algorithm with an unsupervised learning strategy. Classifying different land cover classes assists the monitoring of the Earth's surface. Desertification for example can be recognized through land cover changes in multiple observations of the same area over time.
In remote sensing two main imaging modalities are present. While optical images, acquired by passive sensors, mainly display the observed scene based on reflected sunlight, radar images provide range based measurements of physical properties produced by active illumination of an observed scene [11]. Synthetic Aperture Radar (SAR) is a technique based on active imaging sensors. It offers more advantages than previous radar techniques (see section 2.1). In this thesis, the term radar is commonly replaces by SAR.
For human observers, SAR data is less straightforward to interpret than images. Due to the similarity between optical data and human perception, there is a large availability of annotated optical datasets which is not the case for SAR data. Without labeled SAR data it is not possible to train a classifier in the conventional, supervised fashion. This justifies the demand for a SAR land cover type classifier that can be trained without labeled data.

To enable the training without labels, the approach depicted in fig. 1.2 is considered. Initially an existing labeled optical dataset is selected. Thereafter an open dataset of co-registered optical and SAR images is adopted. This co-registered dataset contains pairs of optical and SAR images that show the same geographical area. The image pairs are utilized for the training of an image translator algorithm. The objective of the translator is to learn how to generate SAR images given optical images. Once it is trained, it can be applied to the labeled optical dataset that was initially selected. The outcome of this procedure is an artificial SAR dataset with labels that correspond to the labels of the optical dataset.

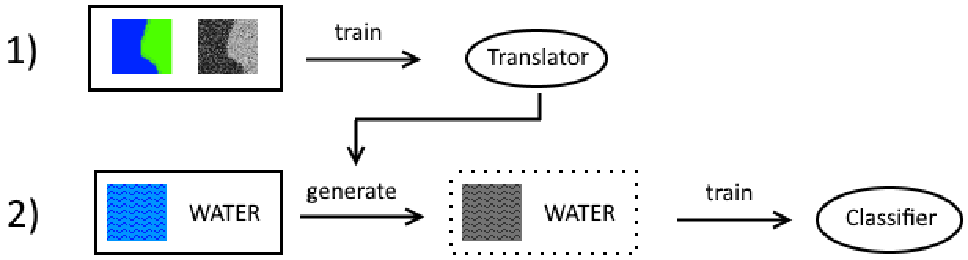


Figure 1.2: The framework of the thesis (unsupervised training of a SAR classifier)

Finally the artificial labeled SAR dataset can be utilized for the training of a SAR classifier which is the main objective of the thesis. Although no labeled SAR data was primarily available, the training of the SAR classifier can thus be carried out in a supervised fashion.

As shown in fig. 1.2, the first step consists of the training of an image translator algorithm. The selected approach is based on Generative Adversarial Networks (GANs), that is a class of deep learning algorithms. The GAN framework has been proposed by Goodfellow et. al. in 2014 [12] to produce random samples of a given distribution. Many variations and improvements have been proposed since the first version of GAN. One of the most popular is the Pix2Pix framework for image translation [13], which has shown to be applicable to many different tasks. Moreover GANs can provide more visual appealing results than other deep learning algorithms used for image translation, like e.g. variational autoencoders [14].

The approach of this thesis is to (1) train a GAN for the translation task utilizing the co-registered dataset and (2) apply the translator to the labeled optical dataset for producing the artificial labeled SAR dataset. Hereafter different deep learning approaches for the training of a classifier can be tested.
For the GAN algorithm two main approaches will be tested and described later on. The first is the Pix2Pix GAN, while the second one is a variant of the ESRGAN [15], that is a more recent network adopted from image super-resolution. The co-registered dataset used for the training of the translators is the Sen1-2 dataset [16], while the labeled optical dataset is EuroSAT [17].

The remainder of this elaboration is structured as follows.
At first the theoretical background of remote sensing will be described, starting with optical imaging and proceeding with radar imagery in general and synthetic aperture radar in particular. Subsequently the datasets and the reasons for their usage are depicted. A chapter explaining the utilized deep learning algorithms completes the theoretical part.
Then follows the description and comparison of the experiments which is concluded by the evaluation of the approach and an outlook.

Chapter 2

Remote Sensing

In remote sensing electromagnetic waves are utilized to gain knowledge about the Earth's surface. Remote sensing relies on two main different strategies for measuring. In contrast to optical imagery which mainly displays the observed scene based on the reflected sunlight, imaging radars provide range based measurements of physical properties produced by the active illumination of the observed scene [18]. These exploit different sensor instruments, which have distinct properties that target specific applications. While optical imagery provides human interpretable images [18], passive sensors are unable to capture scenes independently from daylight, cloud coverage and weather conditions as radars do [19]. Additionally the imaging modalities differ in their geometry and resolution, which will be described in the upcoming sections.

2.1 Radar Imaging

The radar is an active sensor that acquires information by radiating electromagnetic energy and detecting the echo returned from the reflecting target [20]. Since the beginning of the 20th century when the first radar systems were developed many applications appeared, e.g. radar sensors embedded on-board different airborne and spaceborne platforms. In these cases the radar is used to create 2D images of the Earth's surface. Besides all advances in radar techniques, the development of Synthetic Aperture Radar (SAR) led to a big improvement compared to previous state of the art techniques. Nowadays, several airborne and spaceborne SAR systems are operational [19].

2.1.1 Synthetic Aperture Radar

The operating principles of SAR systems [19][20] are based on the concept of the side-looking airborne radar (SLAR) which was developed in the 1950's. Both systems rely on an imaging radar device mounted on a moving platform (e.g. airborne or spaceborne).

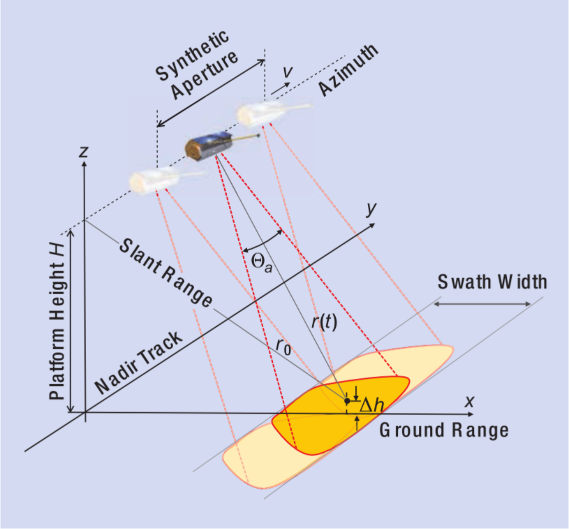


Figure 2.1: Geometry of a SAR device [19]

Fig. 2.1 shows an example of an imaging radar device mounted on a platform that orbits at height H. The coordinate system that is adopted in fig. 2.1 includes a z-axis corresponding to the height, the y-axis parallel to the flight path (Nadir Track) and the x-axis oriented to the measuring direction of the device. Electromagnetic waves are transmitted by the device and afterwards their backscattered echoes are measured. The outcome of this process is an image, where each pixel represents the reflected signal intensity of the ground. This means that the device has to be able to distinguish the locations reflecting the signals. Therefore it sends out frequency modulated pulses (i.e. chirp pulses) and then waits to receive the scattered signal before sending the next chirp. The amount of reflected chirp is defined by the area of the footprint which depends on the focus used during the transmission (dark yellow area in fig. 2.1). Echoes reflected by the area overlay with each other. However, the different echoes can be separated through the modulation of the receiver (i.e. the backscattered intensity can be displayed range-resolved).

The main difference between SAR and SLAR is found in the azimuth direction. For SLAR the azimuth extend of the footprint is the azimuth resolution.

Eq. 2.1 illustrates the issue of SLAR. The azimuth resolution δ_a is defined as

$$\delta_a = r \cdot \theta = r \cdot \frac{\lambda}{L_a} \tag{2.1}$$

with the range r, the aperture θ, the wavelength λ and the antenna length L_a. The equation shows, that with increasing altitude the azimuth resolution of the antenna decreases.

This problem was resolved with the introduction of SAR, which observes each given point on the ground multiple times. The multiple observation of a point with one moving antenna is equivalent to the usage of an array of antennas. In general the azimuth resolution δ_a is the product of the range r and the effective beamwidth θ_{eff} which can be written as

$$\theta_{\text{eff}} = \frac{\lambda}{2L_{\text{eff}}} \tag{2.2}$$

analogously to the previous equation, with the factor of 2 corresponding to a round-trip phase shift (compare [20]) and L_{eff} being the length of the synthetic aperture. L_{eff} itself can again be expressed as a product of r and the ratio of the wavelength to its horizontal aperture D. The δ_a azimuth resolution is then expressed as:

$$\delta_a = \frac{\lambda r}{2L_{\text{eff}}} = \frac{\lambda r}{2}\frac{D}{r\lambda} = \frac{D}{2} \tag{2.3}$$

Eq. 2.3 shows that the azimuth resolution does neither depend on the range, nor on the antenna length which led to the huge success of SAR.

The device has to be capable of detecting the azimuth displacement of each scatterer in the footprint. This is achieved by exploiting the movement of the device in azimuth direction. The Doppler-shift of the chirp signal is used to separate the source location of the echoes. Afterwards the range of each scatterer can be easily determined from the traveling time of the chirp, the knowledge of the platforms height and the beam angle through the Pythagoras' law. Nonetheless SAR data has to be processed before any type of interpretation and information extraction.

Two separate filters are applied along the azimuth and the range dimension as depicted in fig. 2.2 and extensively described in [19]. Additionally the SAR image has to be calibrated and geocoded to show the radar cross section of the reflectivity normalized to area for a specific position of the ground.

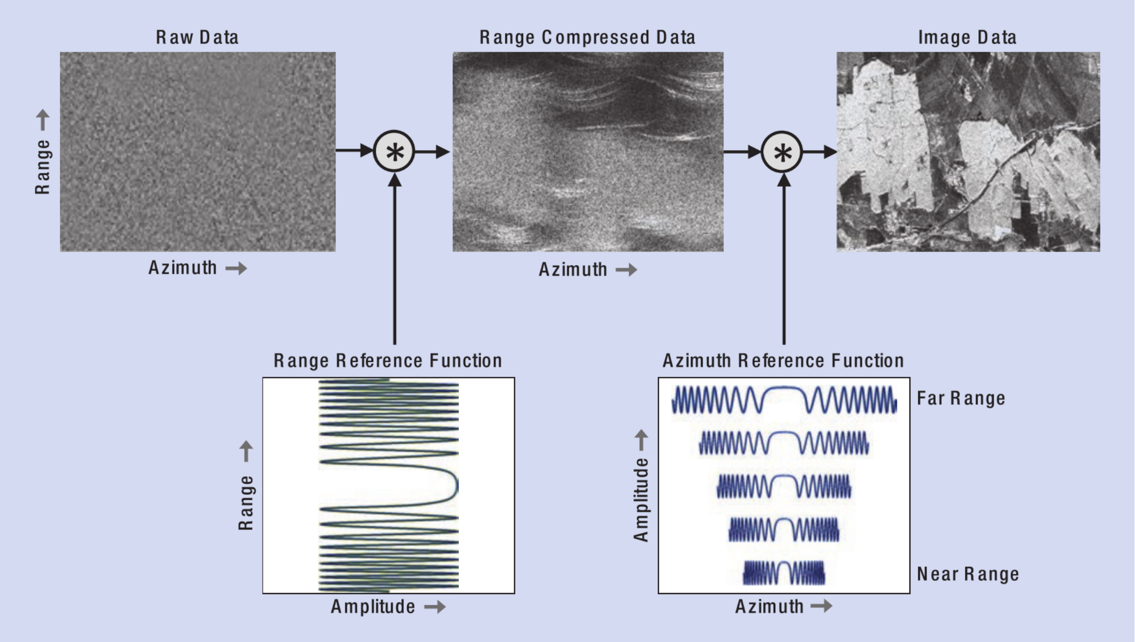


Figure 2.2: Required processing steps to visualize a SAR image [19]

An essential difference between radar imaging and optical imaging lies in the freedom of choice in terms of the transmitted signal (carrier) frequency. The variation of this frequency influences the backscattering behavior of the Earth's surface and the applicability in real scenarios. Longer wavelengths typically lead to more penetration of the surface which allows the detection of subterrestrial structures like pipelines while higher frequencies are used for surface monitoring. Tab. 2.1 shows the most commonly used frequency bands and their corresponding frequency and wavelength values [19].

Frequency band	Ka	Ku	X	C	S	L	P
Frequency [GHz]	40-25	17.6-12	12-7.5	7.5-3.75	3.75-2	2-1	0.5-0.25
Wavelength [cm]	0.75-1.2	1.7-2.5	2.5-4	4-8	8-15	15-30	60-120

Table 2.1: Commonly used bands and corresponding frequencies and wavelengths

2.2 Optical Imaging

In optical or passive remote sensing [21][22], an aircraft or satellite platform is equipped with sensors to measure the solar radiation reflected by targets on the Earth's surface. The radiation that hits a target can be transmitted, absorbed or reflected. For every material, the reflectance spectrum, i.e. the reflectance as function of the wavelength is unique. Thus the spectral and radiometric resolution are essential for optical sensors. Commonly the sensors operate in the range of visible and infrared light. Depending on the application, different types of sensors are adopted. Panchromatic sensors use a single channel detector sensitive to a broad wavelength range. This results in a grey scale image image with every pixels brightness being proportional to the intensity of the solar radiation reflected by the targets. The number of (brightness) values that each pixel can store corresponds to the radiometric resolution.

Multispectral sensors contain multiple channels. Each channel is sensitive to a different wavelength range. The advantage of multispectral sensors is the possibility to combine channels to create colour composite images. In the simplest case the three channels corresponding to red, green and blue can be combined to create a true colour image. Although other combinations like near infrared, red and green are also useful and common, RGB images benefit the ease of interpretation for a human observer.

Besides multispectral sensors also hyperspectral sensors exist. Those are characterized by the ability to measure more than hundred contiguous spectral bands. The information acquired resembles the measurements of a spectrometer and allows the improved characterization and classification of targets.

2.3 Datasets

For training a SAR classifier without labeled SAR data two datasets are required. At first a dataset of co-registered optical and SAR images has to be acquired. Since it is not trivial to generate these datasets, only few are available. First testings have been performed with the SARptical dataset [23]. It turned out that the translation between the images did not work well. The most likely reason for this was the major difference between the image geometries as depicted in fig. 2.3.

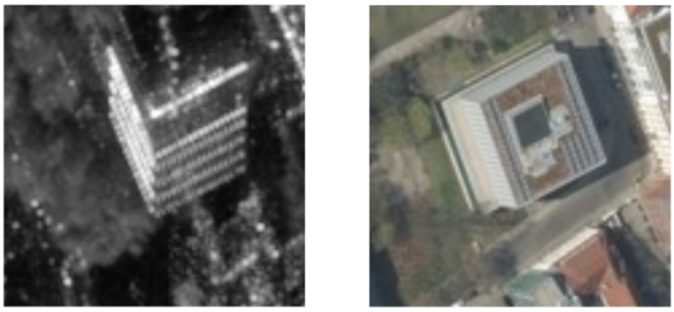


Figure 2.3: Example of co-registered optical and SAR images from the SARptical dataset [23]; large differences in the image geometries become apparent

Additionally the dataset considers only urban area and thus is too one-sided to find a translation algorithm for diverse remote sensing scenes. The Sen1-2 dataset [16] described in section 2.3.2 promised to overcome these limitations.

The EuroSAT dataset [17] includes images that were acquired by the same satellite as the optical images in the Sen1-2 dataset. This similarity promises to improve the translation, making these two datasets a beneficial starting point. For a more detailed description of the following compare [16] resp. [17].

2.3.1 The EuroSAT RGB Dataset

Since 2014 the European Space Agency (ESA) runs the Copernicus program. This program includes a fleet of satellites called Sentinels. Sentinel-2A and Sentinel-2B mount multispectral sensors for monitoring land use and vegetation. Working together, the satellites scan the entire land surface every five days, allowing to monitor the development of areas of interest. An important aspect of the Copernicus program is the open and free access to the data.

The EuroSAT dataset[1] consists of 27.000 labeled images, each assigned to one of 10 different classes. There exist two variants of the dataset, the first carrying the complete spectral information (13 bands), whereas the second contains only the three spectral channels B04, B03 and B02 representing the red, green and blue channels, respectively. Since the Sen1-2 dataset only includes the RGB channels, this work exploits the second variant.
The scenes utilized to generate the dataset represent European area and are chosen to have a small cloud coverage. Additionally one primary objective when selecting the images was a high intra-class variance. Thus images of many countries, recorded over a complete year were extracted.
Each provided image patch has a size of 64x64 pixels with a resolution of 10 meters per pixel. The classes are relatively well balanced, each containing 2000 to 3000 images. An overview of the classes is depicted in fig. 2.4. As can be seen, built-up areas, water bodies and agricultural land uses are further differentiated.

The images remaining in the dataset were checked manually multiple times to sort out mislabeled images and images containing snow or ice. Atmospheric correction has not been performed. This sometimes results in a color cast which will also be discussed later on.

---

[1] downloaded at: http://madm.dfki.de/downloads

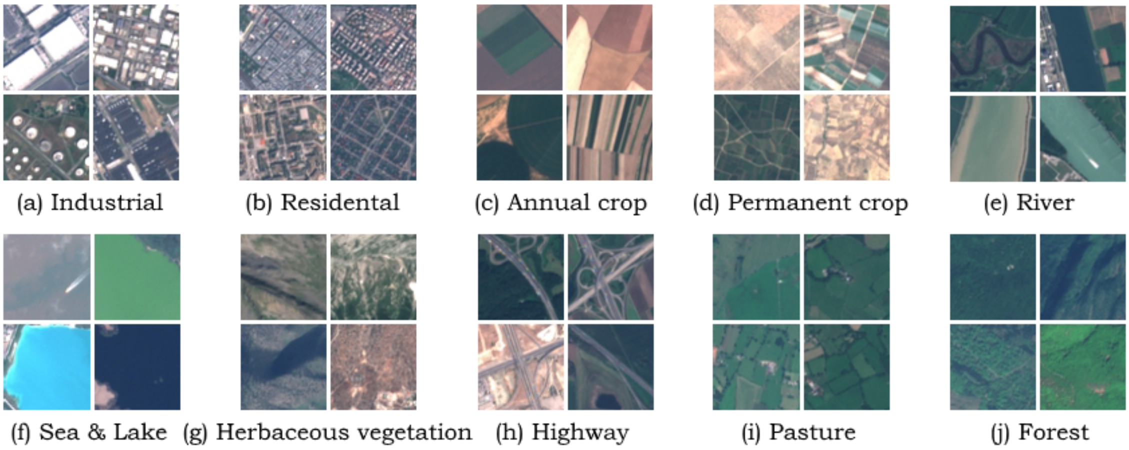


Figure 2.4: Examples of each class of the EuroSAT dataset [17]

The data was stored using one .jpg file for each image with 24 bits per pixel i.e. 256 radiometric channels.

2.3.2 The Sen1-2 Dataset

The Sen1-2 dataset[2] contains 282.384 pairs of co-registered SAR and optical images. The optical image patches are chosen under roughly the same conditions as in the EuroSAT dataset. Only the RGB channels of the Sentinel-2 images were used and no pre-processing was applied. The initial selection of viable scenes was filtered such that only images with less than 1% cloud coverage were taken into account.

The corresponding SAR scenes were acquired with the two Sentinel-1 satellites. Both satellites are equipped with C-band SAR sensors that measure the backscatter coefficient for each pixel in dB scale. The pixel spacing is 5m in azimuth and 20m in range direction. Only the vertically polarized (VV) channel was used, i.e. single-polarized images are provided.

To create the co-registered dataset with focus on the diversity of the data, a number of steps had to be carried out, described in detail in [16]. In the end randomly sampled scenes in between 56° South and 83° of the Earth's landmass were included. An artificial bias was introduced by choosing 2/3 of the data points randomly and 1/3 randomly over urban areas. Only data acquired in 2017 was taken into account and sorted by the seasons.

---

[2] downloaded at: https://mediatum.ub.tum.de/1436631

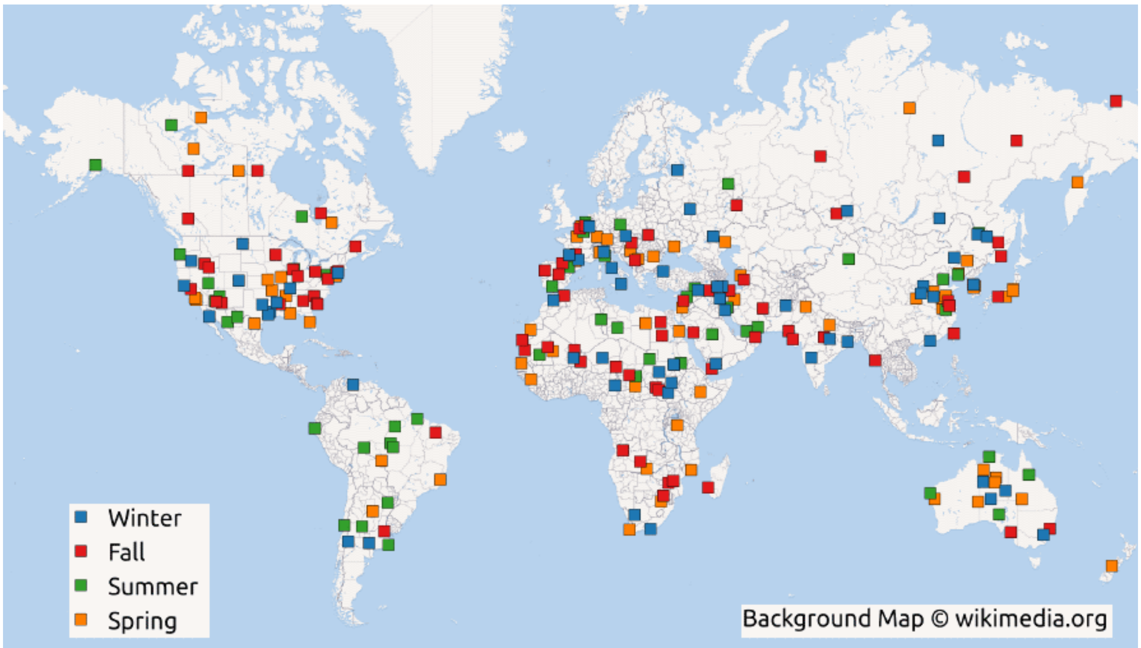


Figure 2.5: Representation of the scenes utilized to create the Sen1-2 dataset [16]

From the collected scenes which are depicted in fig. 2.5 patches of 256x256 pixels were generated using a stride of 128 pixels i.e. neighbouring patches are overlapping. An additional manual inspection of all patches was carried out. Fig. 2.6 shows examples of patch pairs in the dataset.

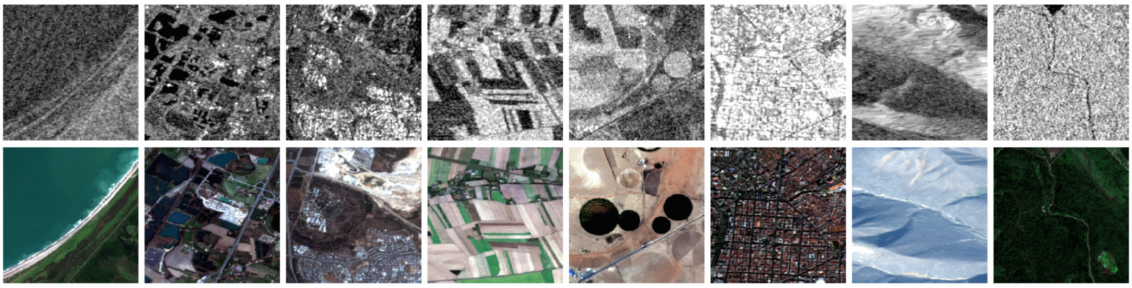


Figure 2.6: Example patch pairs of the Sen1-2 dataset [16]; top: single-polarization SAR images, bottom: RGB optical images

The top level structure of the dataset corresponds to the four meteorological seasons. Each contains the scenes acquired which were split into image patches. The patches are stored using .png files with every pixel containing one resp. three color channels for SAR resp. optical image patches with 256 radiometric channels each.

Chapter 3

Deep Learning Methods

To optimize classification and translation algorithms given the previously described datasets, different deep learning techniques are used.
Machine learning promises to be the best approach here, since the transformation between active and passive imaging methods is very complex i.e. no simple rules exist to program the algorithms by hand. This applies especially when the algorithms need to be robust to deviations e.g. in illumination, scattering, geometry etc.

The upcoming section describes the basic (deep) neural network structure and setup. This is followed by more recent types of algorithms, i.e. convolutional and residual networks. Afterwards a brief overview about methods to improve the performance of the networks is given. Finally generative adversarial networks are introduced with focus on the task of image translation.

3.1 Deep Neural Networks

The objective for any type of neural network is to map a given input to some desired output. As mentioned, the inputs for the same output can vary which means that the algorithm has to be robust and capable of abstracting features. An example would be the classification of handwritten digits, where several representations of the number two are present, but they should all be mapped to the label "2". Theoretically, there is a neural network to approximate any arbitrary mapping function (i.e. universality theorem) [24]. However, the theorem does not state how to find the fitting network structure. One reason of the huge success of deep learning techniques over the last years comes from the formulation of the back-propagation algorithm in 1988 [25], an efficient algorithm to optimize the network for a given task.

A neural network is a structure built up using one fundamental element, the neuron. Leaving aside the neurophysiological background from which the concept is derived, the structure of these neurons is shown in fig. 3.1.

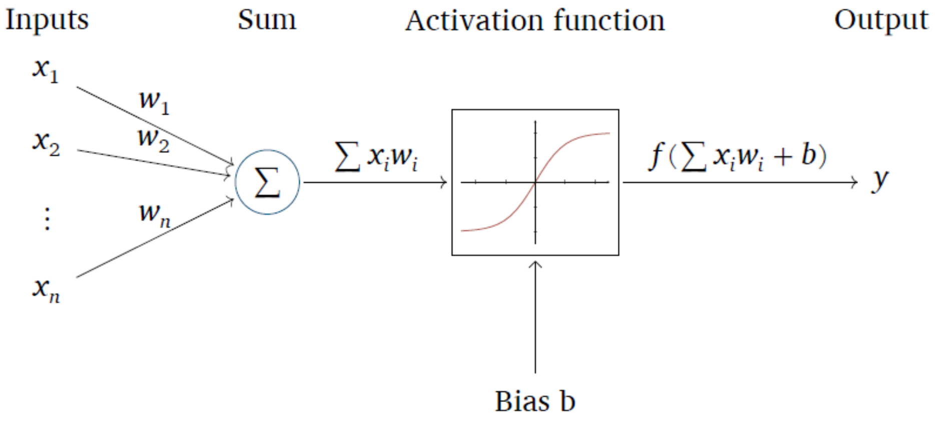


Figure 3.1: The structure of a standard artificial neuron, depicting how the output is calculated, given the input.

Each neuron represents an input-output-function by itself. The neuron receives the sum of the products between the n inputs x_i and their corresponding weights w_i. A bias b is added before the activation function is applied in order to shrink the input sum into a desired range. The function value is then the output of the neuron. A more detailed description of the design principles and reasons can be found in [26] and [27]. The most commonly used activation functions are the sigmoidal function and the rectified linear unit (ReLU) function $f_{relu} = \max(0, x)$.

A simple approach for building a neural network from multiple neurons is the feed-forward network structure.
As depicted in fig. 3.2 it consists of multiple layers, where every neuron of a specific layer is fully connected to all neurons of the previous and next layer. The first layer is called input layer and its neurons receive one specific feature of the data as input. In the handwritten digit example the input could be a gray-scale image. Then each neuron in the first layer would get the value of one pixel of the image as input. The last layer is referred to as output layer. It consists of c neurons, where c is the number of classes of the classification task (ten for the handwritten digits 0-9). All layers in between are called hidden layers and their input resp. output comes from the previous resp. goes to the subsequent layer. Using multiple hidden layers leads to the term Deep Neural Network (DNN).

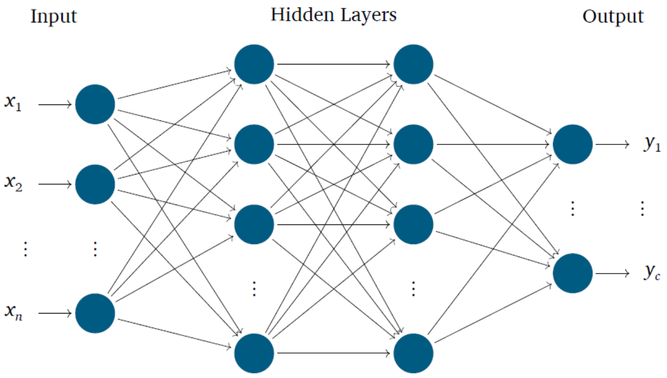


Figure 3.2: A sketch of the layer-wise structure of a feed-forward neural network. It consists of input-, output- and hidden layers. A neuron in a specific layer is connected to all neurons of the neighbouring layers.

Besides the structure itself, the parameters describing the DNN are the weights and biases of every neuron. For sufficiently deep networks the number of parameters to estimate can be in the range of millions.
Supervised learning can be used to compute these parameters. The optimization of the algorithm is performed with training data (i.e. an annotated dataset consisting of input-output pairs). The input value is given to the algorithm which produces an output estimation. This estimation is than compared to the real output and in case of a wrong prediction the parameters get updated accordingly. Finding the best set of parameters for a given DNN structure can be formulated as an optimization problem. A cost function is introduced to measure the distance between the prediction and the real output (compare [26] for more details). Be N the number of samples in the training set. Then this set contains N input-output-pairs (x, y), where y is the label of x. The cost function C measures - for a given input - the distance from the networks output $\hat{y}$ to the desired output y. A simple example for a cost function is the mean squared error summed up for all elements of the training set.

$$C(w, b) = \frac{1}{2N} \sum_{x} (y - \hat{y}(x))^2 \tag{3.1}$$

C depends on all the networks weights and biases, because $\hat{y}$ is a function of these parameters. The optimization task is to find the minimum of this cost function through the minimization of the difference between the networks output and the real output. One way to solve this optimization problem is the usage of the stochastic gradient descent (SGD) algorithm or one of its more recent variants like RMSProp [28], AdaGrad [29] or Adam [30].

All these algorithms calculate the gradients of the cost function depending on the weights and biases. The parameters are then updated in the direction of a (local) minimum. The step size is scaled by a hyper parameter called learning rate. A small learning rate corresponds to small updates and vice versa. In the cost function shown before all training samples are used for the calculation. This is often not ideal, since one update requires the computation of gradients for every training sample and thus quickly becomes computational expensive. A solution for this problem is to use only a subset (or mini-batch) of the training data to approximate the real gradients and perform (stochastic) updates.

Finding the gradients is possible with the Back-propagation algorithm mentioned before. It takes the deviation of the networks output from the desired output and passes this backwards through the network, making thereby use of the chain rule for derivation. Thus the gradient is made available for all parameters in every layer. For every connection between layer l and $l-1$ the proportion of the error passed back depends on the weight of this connection. Thus, connections with a big influence change more than the ones with a smaller influence.

3.2 Convolutional Neural Networks

Although the universality theorem states that there exists a DNN that can compute any function, it turns out that it often can not cope with real applications. Using the example of computer vision, where usually RGB images are passed into those networks, one finds that in DNNs the neurons are connected to every pixel of the image, not making use of the expected locality of these data structures. Encoding this already in the networks structure improves the performance in the majority of the cases dramatically. CNNs, proposed by LeCun et al. in 1998 [31], are designed to include the locality in their structure. Fig. 3.3 shows a visualization of a standard CNN structure. To build a CNN two new types of layers are defined in addition to the fully connected layer used in DNNs.

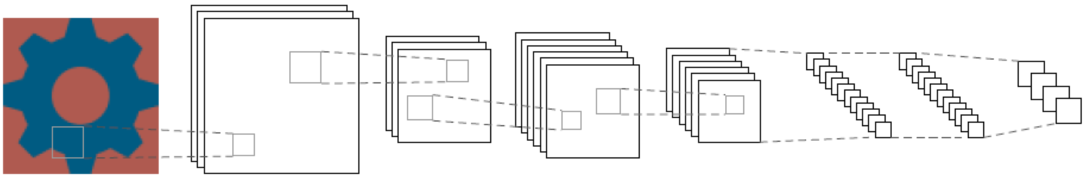


Figure 3.3: Visualization of a standard CNN structure, starting with convolutional and pooling layers and concluding with fully connected layers.

The eponymous convolutional layer applies a number of learnable filters which are slided over the input in order to create an activation map. This map encodes the similarity of a local structure and the filter using dot products. Higher similarity leads to larger activation values. The activation maps are then passed to the next layer. The convolutional layer thus works as a feature detector which learns the filters necessary to categorize the data while training.

Close to the networks input layer the filters adapt to small features like edges and corners because their input is strongly correlated to the original image. To obtain more abstract filters pooling layers can be utilized to reduce the spatial dimension of the tensor passed through the layers. A Pooling layer takes small windows (typically 2×2) of the input and maps them to one value, repeating this for the whole image. This produces tensors of lower resolution on which in general the next convolutional layer is applied, which now receives higher level features.
After some repetitions of this process the most abstract filters are often reshaped to a one dimensional vector which can then be fed into fully connected layers for prediction. The standard CNN structure thus is composed of two parts, the first extracting local features and the second using them for predictions.

3.3 Residual Neural Networks

Artificial neural networks benefit from being deep [5]. Intuitively a not shallow network includes the detection of low- to high-level features which can be enriched by adding more layers i.e. depth to it [4]. He et. al. argued that the network depth is very important but the training becomes more difficult the deeper the network become. In 2016 they introduced a new type of building block to ease the training of very deep neural networks [7].
The group observed that "with the network depth increasing, accuracy gets saturated [...] and then degrades rapidly." They argue that compared to a network with l layers, a network with $l + d$ layers would work at least as well as the smaller one if one copies the l layers and adds d identity layers afterwards. Since current networks were unable to find these solutions by themselves, the group came up with the idea of encoding this behaviour in the network structure. Therefore they introduced the residual building block as shown in fig. 3.4.
The core idea is that after a fixed number of (e.g. convolutional) layers, in this case two, the input to the first layer is added to the output of the last layer of the block. When the networks is trained, the weights learned are the deviations from the iden-

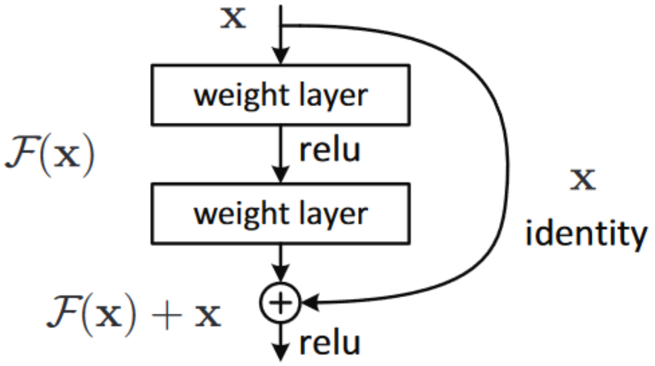


Figure 3.4: The basic building block of a residual network as proposed by [7]

tity i.e. the residuals. If the identity would be optimal, the weights of these layers would be pushed towards zero which the group assumes to be easier then learning the identity from scratch. In general they suppose that the optimal function is closer to an identity mapping than to a zero mapping. Thus the residual building block is overall beneficial.

After the paper was released, many enhancements have been proposed. The building blocks actually used here will be presented later on.

3.4 Improving the Networks Performance

Due to its increasing popularity, many people work on the topic of deep learning. Thus a huge amount of innovations and improvements in designing and training deep neural networks have been proposed. Some of those are used in the upcoming sections and shall be introduced here briefly.

3.4.1 Regularization

One of the main problems in Deep Learning is overfitting [26]. The input data is split into training, validation and test sets. While the training set is used to optimize the parameters of the network, the validation and test sets are adopted for the estimation of the generalization error [26]. When there exists a gap between the performances on these sets, overfitting might occurred.
Since this is a common problem for networks with many parameters, different solutions arose.
Two main regularization techniques exist. The first requires a change of the cost function to penalize specific network parameters. Thereby the parameters are re-

stricted which leads to more generalizing networks. A second approach is Dropout [32] which randomly switches off a specified ratio of the connections between neurons. This results in a more robust network because it forces the network to perform well even without relying on specific connections. Typically regularization techniques lead to worse results in the training phase but improved results on unseen data.

3.4.2 Augmentation

Sometimes overfitting occurs due to insufficient diversity in the dataset. A core assumption in deep learning is that the training data and the real data, which is observed later on in the application phase, are from the same distribution [33]. Using facial recognition as an example, the faces in the training data might always be centered in the image due to dataset preparation steps. If the face of a person in the application phase has an offset in one direction (i.e. the person did not properly position himself in front of the camera), the algorithm might struggle to work correctly. One solution to this problem can be data augmentation. To augment the data means to either change or enlarge the dataset with respect to different transformation rules. Simple examples are rotation, translation or mirroring of images. In the facial recognition example the translation of a part of the images in the dataset would very likely solve the problem. While mirroring could also improve the learning of the algorithm, rotation would probably hinder the training unnecessarily if one expects the subjects to be upright. This shows that it is of essential importance to exactly know and understand the field of application for each algorithm.

Augmentation can be applied in two ways. Either the augmented data is added to the dataset or the samples are replaced by their augmented versions. While a larger dataset often seems preferable, it also slows the training and increases the necessary memory which makes the decision again dependant on the use case.

3.4.3 Batch Normalization

Since it has been proposed by LeCun et. al. in 1998 [34], the whitening of the neural networks input has become common in practice. Most of the time whitening means subtracting the mean and dividing by the variance of the input, whereas the decorrelation is often too costly or complicated to perform. This pre-processing of the dataset often leads to faster convergence thus easing the tuning of the networks parameters [34].

The idea behind batch normalization is to perform this normalization in between layers. Ioffe and Szegedy proposed batch normalization in 2015 [35] where they also demonstrated that this technique is capable of improving the training performance and stability while reducing the time required. For several reasons, described in [35], the normalization (which also refers to zero mean and a variance of 1 here) can not be done independent of the gradient descent step. Therefor the batch normalization is included in the network just like other layers and placed before each activation function. Applying batch normalization before the activation function is especially beneficial for saturating functions like sigmoidal and hyperbolic tangent as it hinders the saturation in the backpropagation.

Since the optimization is commonly done using stochastic gradient descent, the normalization uses the statistics of each mini-batch $\{x_{1 \ldots m}\}$ in the training set. Given the input x_i and the output y_i the algorithm can be written as

$$\mu = \frac{1}{m} \sum_{i=1}^{m} x_i \tag{3.2}$$

$$\sigma^2 = \frac{1}{m} \sum_{i=1}^{m} (x_i - \mu)^2 \tag{3.3}$$

$$\hat{x}_i = \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}} \tag{3.4}$$

$$y_i = \gamma \cdot \hat{x}_i + \beta \tag{3.5}$$

with the mean μ, the variance σ and a small ϵ for numerical stability. In the last line two parameters γ and β were introduced to scale and shift the normalized value. Without these parameters the output of the batch normalization would be constrained which afterwards could limit the usable regime of the subsequent activation function. Both γ and β are trainable parameters and thus optimized through the gradient descent algorithm.

3.5 Generative Adversarial Networks

The discriminative algorithms introduced so far aim to classify the input data, i.e. predicting a label y given some input features x, e.g. pixels. In contrast, the objective for generative algorithms is to predict the probability $p(x|y)$ that a certain feature x is present, given the label y.

This objective can be formulated as approximating the distribution of x. In the past it has been very difficult to set up generative models [12].

Generative adversarial networks have been introduced in 2014 by Goodfellow et. al. [12]. The main contribution of this paper was to present a framework to effectively train a generative network with gradient descent and backpropagation.

In a GAN, two models, the generative network (generator) and a discriminative network (discriminator) pit against each other, thus the term adversarial. The discriminator hereby tries to distinguish whether a given sample is from the real distribution or the model distribution, i.e. whether the sample is real or generated (fake). Both networks are trained in an alternating fashion. In doing so both improve, leading to real and fake samples becoming closer and finally being indistinguishable i.e. their distributions become equal.
The principle GAN framework is shown in fig. 3.5.

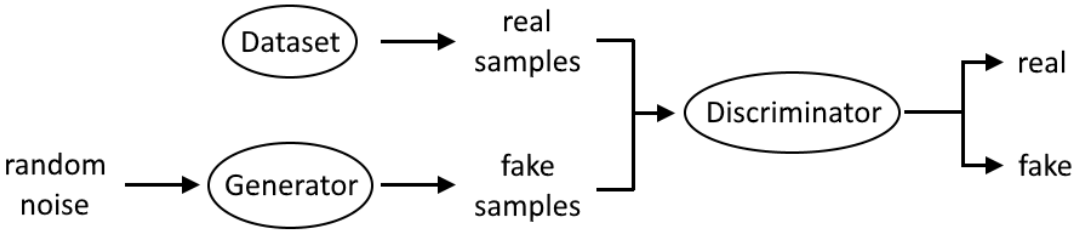


Figure 3.5: The principle structure of a generative adversarial network

For not being deterministic, the generator G receives noise of distribution p_z as input. G then represents a mapping from the noise variables $p_z(z)$ to the data space, $G(z; \theta_g)$ being a differentiable function (commonly some kind of artificial neural network) defined by a set of parameters θ_g. The discriminator $D = D(x; \theta_d)$ is then a second network (i.e. differentiable function) defined by another set of parameters θ_d that outputs a single scalar. Its inputs are both real and generated data. $D(x)$ represents the probability that x is part of the real (training) data rather than generated by G (i.e. of the generators distribution p_g). An output of 1 resp. 0 corresponds to the discriminator considering the input to be real resp. fake.

To optimize the networks, a cost or objective function needs to be defined. Coming from a game theory perspective, the objective can be formulated as a min-max game, where the discriminator tries to maximize the probability that real data is real and fake data is fake. The generator on the other side tries to fool the discriminator, minimizing the probability that fake data is classified as fake. Using the log-likelihood [26] as value function L_{gan}, the objective of the discriminator resp. generator can be formulated as shown in eq. 3.6 resp. eq. 3.7.

$$\max_D L_{dis} = \mathbb{E}_{x \in p_{data}(x)}[\log(D(x))] + \mathbb{E}_{z \in p_z(z)}[\log(1 - D(G(z)))] \tag{3.6}$$

$$\min_G L_{gen} = \mathbb{E}_{z \in p_z(z)}[\log(1 - D(G(z)))] \tag{3.7}$$

In practice eq. 3.7 does not lead to sufficient gradients. Especially in the often observed scenario that G outputs poor samples in the beginning of the training, D can reject these samples with high confidence. This leads to $\log(1 - D(G(z)))$ saturating. Therefor eq. 3.7 is replaced with eq. 3.8 which provides stronger gradients in early stages of the training while conserving the same equilibrium point $p_g = p_{data}$.

$$\max_G L_{gen} = \log(D(G(z))) \tag{3.8}$$

The process of training of a GAN is graphically illustrated in fig. 3.6.

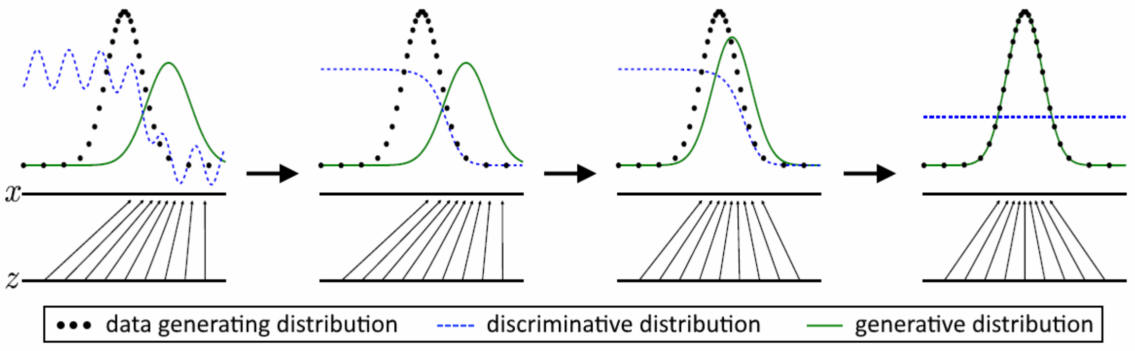

x
z
••• data generating distribution
----- discriminative distribution
— generative distribution

Figure 3.6: The (ideal) training scenario for a GAN (adapted from [12])

In the beginning, the data generating distribution and the generative distribution do not correspond. Also the discriminative distribution can only poorly distinguish them. Going ahead in the training, the discriminator improves in distinguishing both distributions. This leads to improved gradients for the training of the generator which also improves. The procedure repeats until the data generating and the generative distribution correspond and the discriminator is unable to distinguish between them. This is of course the ideal case, which does not necessarily emerge while training.

Actually the training of GANs turns out to be complicated due to several reasons e.g. instability and mode collapse [36]. Instability refers to possible large jumps and oscillations in the training process in contrast to a smooth convergence to the equilibrium. Mode collapse on the other side describes the occasional behaviour of having only a small variance in the output, independent on the generators input, e.g. a generator for faces that only produces three different faces.

Lots of new methods and architectural guidelines appeared to tackle these problems and improve the framework (compare e.g. [36][37][38][39]).

One structural innovation was the introduction of conditional GANs (cGANs) by Mirza and Osindero [40]. It enables to control the generation of samples in the application phase by passing additional information y to the model. The extension to a cGAN requires to pass this information to the generator and the discriminator in the training phase. For any type of neural network this can be done by defining an additional input channel for the discriminator and by concatenating the information to the noise input of the generator. A typical use case would be to add the label information given a labeled dataset, although any arbitrary information can be passed to the network. The objective function differs only from eq. 3.6 and eq. 3.8 by the discriminator and Generator depending on their input given the additional information ($D(x) \rightarrow D(x|y)$ resp. $G(z) \rightarrow G(z|y)$).

The structure of cGANs allows to utilize the GAN framework in image translation tasks by passing complete images as condition.

Chapter 4

The Pix2Pix Framework for Image Translation

The upcoming chapter introduces the Pix2Pix framework utilized to translate optical to SAR images. Initially the framework and its implementation details are described. Since Python is used as programming language here instead of Lua which was utilized in [13], the code has to be implemented manually. Thus a baseline test is performed first to provide comparable results. Subsequent the results of the translation procedure are shown and evaluated.

The paper "Image-to-Image Translation with Conditional Adversarial Networks" [13] by Isola et. al. introduces a cGAN capable of image to image translation. A key point of their contribution is to add a general purpose solution for this kind of problem which does not depend on the dataset used.
The examples shown in the paper range from day-to-night and black-white-to-color conversion to edges-to-photo and satellite-to-map translation as depicted in fig. 4.1.

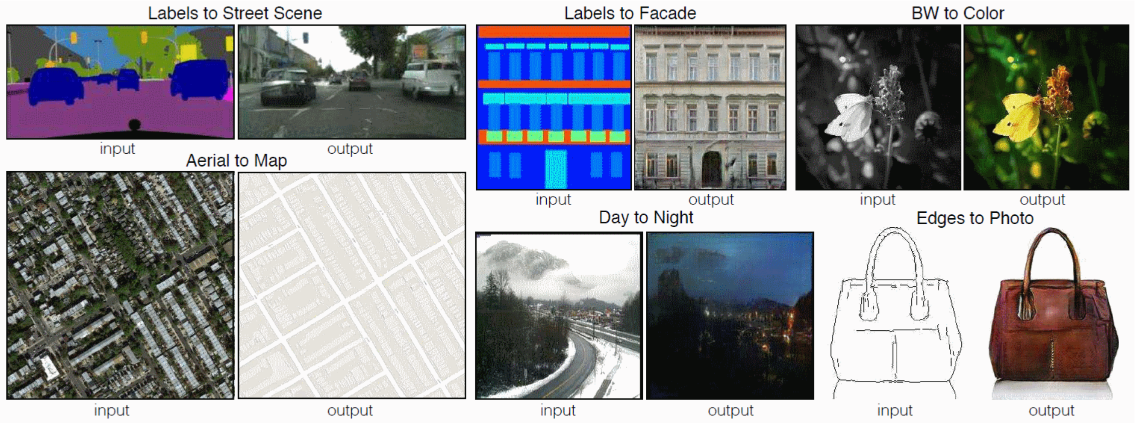


Figure 4.1: Example applications of the Pix2Pix framework [13]

Due to the diversity of already existing applications, the framework was selected as starting point for the translation between optical and SAR images.

From an algorithmic perspective the paper made two main contributions. The objective function is modified to better match the image translation task. Also a specific network structure is proposed and tested.

The objective corresponds to the previously defined eq. 3.8 with a pixelwise L1 distance loss term (eq. 4.1) added.

$$L_{L1} = \mathbb{E}_{x,y,z}[||x - G(y, z)||_1] \tag{4.1}$$

The L1 term is added since it was found to be "beneficial to mix the GAN objective with a more traditional loss, such as L2 distance." [13] by previous papers. Using the L1 instead of the L2 distance results in less blurry images. The complete loss is given by eq. 4.2.

$$L_{p2p} = L_{gen} + \lambda L_{L1} \tag{4.2}$$

Before adding the L1 term to the generator objective it is weighted by a factor λ which allows to either balance the terms or to focus more on one objective.
For the noise variable z the generator was found to ignore it after some training, relying only on the condition y. The group therefore applied a variant of dropout which was not switched of in the application phase and introduced the noise that way. Still they only found minimal randomness in the networks output.

The network architecture build in the paper adapts the model guidelines proposed in [36]. This means that neither fully connected layers nor pooling layers are used, only convolutional layers. Batch normalization is applied in the generator and the discriminator. As activation function the ReLU function is used in the generator, only the last layer uses the hyperbolic tangent function. For the discriminator the leaky ReLU function defined in eq. 4.3 is used.

$$f_\alpha(x) = \begin{cases} x & \text{if } x \geq 0 \\ \alpha \cdot x & \text{otherwise} \end{cases} \tag{4.3}$$

$\alpha \in [0, 1]$ specifies a smaller slope for negative values of x.

The structure of the generator corresponds to an encoder-decoder network (autoencoder) [41]. Usually this kind of network is used to reduce the dimensionality of data for example to reduce the memory required to store an image. It is composed of two parts. The encoder, which is the first part of the network, downsamples the input image by applying convolutional layers. This produces a tensor with small spatial dimensions which is the input to the decoder. The task of the decoder is to reproduce the initial input, i.e. reverse the process, by upsampling the tensor that was passed through the so called bottleneck layer. Since all information has to flow through the bottleneck layer, the network is forced to learn to solely pass important information. After successfully training the autoencoder, the encoder can be used to reduce the dimensionality, whereas the decoder is capable of reconstructing the initial data.

A problem that often arises with autoencoders, especially if they are not trained in an adversarial way but with gradient descent and backpropagation, is that they aren't capable of reproducing high frequency information, since it is lost when the information flows through the bottleneck layer. In image translation this corresponds to not restoring small objects, sharp edges etc., which also leads to blurry results. To overcome this, skip connections as proposed in [42] and shown in fig. 4.2 are adapted. Utilizing skip connections means that the output of an encoding layer is concatenated to the output of the corresponding (i.e. having the same dimensionality) decoder layer. This is motivated by assumption that low-level information, e.g. prominent edges, are shared between the input and the output images.

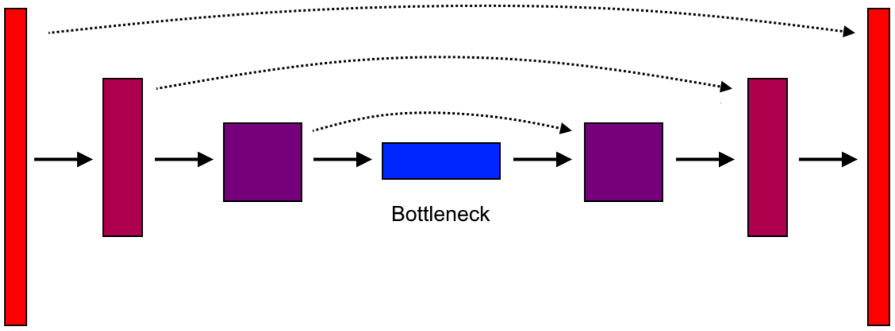


Figure 4.2: Principle structure of an autoencoder with skip connections (U-net)

To further improve the sharpness of the output, the discriminator is restricted to evaluate image patches of $n_{patch} \times n_{patch}$ pixels. This discriminator architecture, named PatchGAN, forces the GAN to improve the rendering of smaller structures. The final discriminator output is an average over the outputs of all evaluated image patches.

4.1 Implementation Details

For the implementation Google's machine learning library Tensorflow [43] is applied here in conjunction with the Keras library [44].

The reproduction of the Pix2Pix framework consists of three parts, the architecture of the discriminator, generator and the training algorithm. Each of these parts has many parameters which are tunable and are initialized to resemble the original ones. Both, the discriminator and generator consist of fundamental building units that combine several of the previously described layers.

The basic discriminator layer includes a convolutional layer with leaky ReLU activation and optional batch normalization. As in [13] filters of size 4×4 are used with a stride of 2 and padding such that the layers output tensors of half the spatial dimension they received as input. The number of filters doubles in subsequent layers starting with an initial value of 64. For the leaky ReLU function $\alpha = 0.2$ is chosen and the batch normalization, if applied, is initialized with a momentum of 0.9.
The discriminator consists of four of these layers, where batch normalization is applied for all but the first layer. After the fourth basic layer a convolution with one filter (size 4×4, stride 1, sigmoid activation function) is added that keeps the spatial dimensions but outputs one channel corresponding to the predicted probability of the input being real. As input the discriminator receives the spectral concatenation of the real resp. generated image and the condition.

Since the generator comprises an encoder and decoder, there are two basic layers. The basic layer of the encoder part is equivalent to the basic discriminator layer. The basic decoder layer performs spatial upsampling through a combination of interpolation (i.e. doubling the spatial size and copying values to new neighbouring pixels) and convolution (with filter size 4×4, stride 1 and ReLU activation). This is again followed by batch normalization. In [13] encoder and decoder consist of 8 layers resulting in a spatial dimension of 1×1 for the bottleneck layer if the input is of shape 256×256 as proposed. For the number of filters 64-128-256-512-512-512-512-512 is proposed for the encoder and 512-512-512-512-256-128-64-(1/3) for the decoder. The last layer of the decoder differs from the other layers. It applies a hyperbolic tangent activation function which compresses the output into [-1, 1] and the number of filters corresponds to the desired number of channels in the output image. Both, the first and the last layer omit do not utilize batch normalization. Collectively the GAN is composed of 57,2 million (generator: 54.4 million) trainable parameters for an input size of 256×256. Since the application of dropout did not influence the proposed network as described earlier and the translation between optical and SAR images does not require randomness, it is omitted here.

The optimization of the networks is implemented in an alternating fashion. For the training of the discriminator the generators weights are frozen and vice versa. Applying stochastic gradient descent, the Adam optimizer [30] is used with a learning rate of 2×10^{-4} for the generator and 1×10^{-4} for the discriminator and momentum parameters $\beta_1 = 0.5$ and $\beta_2 = 0.999$. Thus the discriminator is trained slower than the generator. The weighting factor of the generators objective function (eq. 4.2) is set to $\lambda = 100$ which yielded good results in [13]. 200 epochs and a batch size of 10 is used for the first experiments. All image values are linearly projected from $[0, 255]$ to $[-1, 1]$ before training the networks.

4.2 Baseline Tests

Since [13] does not describe the training behaviour but gives a qualitative and quantitative analysis, the self-implemented framework is tested on a task where it is already known to work. This confirms the implementation correctness and provides comparable information about the training modalities. To test the implementation, a dataset of Google maps image pairs gathered in [13] is utilized[1] due to its similarity to the translation between optical and SAR imagery. It consists of two types of images, the first being aerial photographs, i.e. optical satellite images and the second RGB street maps. An image pair of the training set is depicted in fig. 4.3.

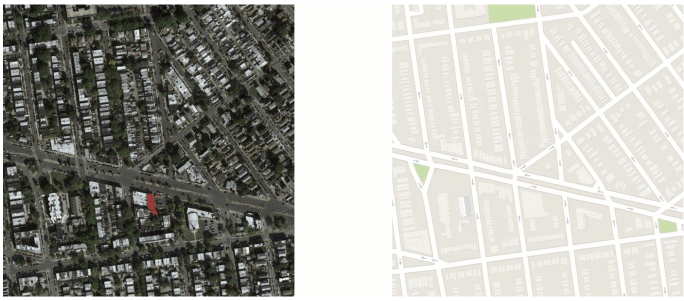


Figure 4.3: Example image pair of the Google maps dataset

---

[1] http://efrosgans.eecs.berkeley.edu/pix2pix/datasets/maps.tar.gz

The dataset consists of 1096 training and 1098 test image pairs of size 600 × 600 pixels each. For the training subimages of 256 × 256 pixels are utilized. [13] also applies augmentation by mirroring images. Since no details are provided about the fraction of images that are changed and if they are added to the dataset or replace the original images, it is omitted in the first tests.
For comparison the qualitative results of training the Pix2Pix framework on this datasets given in the original work are shown in fig. 4.4.

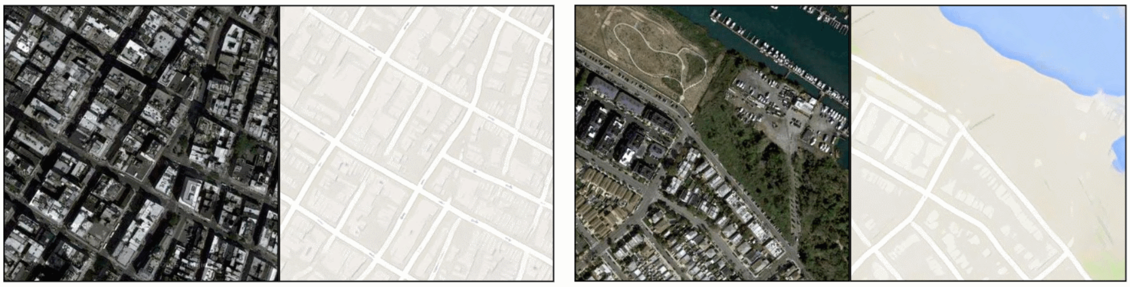


Figure 4.4: Two examples of optical satellite images translated to maps from [13]

The training was performed on images of 256 × 256 pixels, as previously described, but the example images are merged to size 512 × 512.

The first run is performed using the same parameters as in [13] and described earlier. Fig. 4.5 shows some of the images generated in the last stages of the training using (unseen) test data as input.

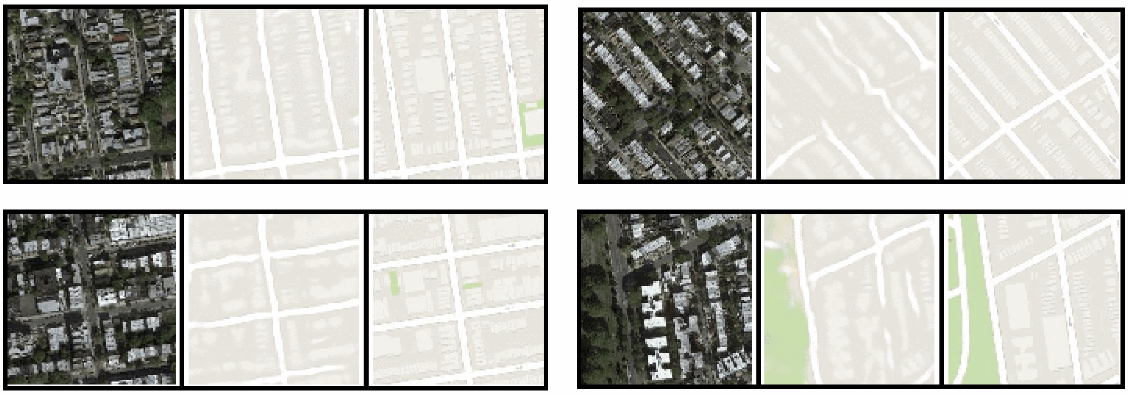


Figure 4.5: Four examples from the last epochs of the training. Each example consists of the condition (left), the generated image (middle) and the original image from the dataset (right).

Both examples on the left side in fig. 4.5 show relatively well translated images, whereas the examples on the right side depict poorly translated ones. When describing examples as well or poorly generated, this relates to the comparison with

the original image on the one hand and to the plausibility of the example on the other hand. Both these indicators for the visual quality of an example are difficult to quantify and metrics for the general evaluation of the analyzed algorithm are discussed more deeply in the upcoming section. The examples on the right side of fig. 4.5 can easily be classified as fake images by a human observer, mainly because the streets aren't straight and have gaps. Besides streets, buildings are another feature of urban areas. In the given examples they appear reasonably, which is also the case for most of the images translated by the GAN, although the edges are not as sharp as in real images. Green areas could also be compared, but due to the relatively small amount of images in the dataset that contain parks, grassed area or accumulations of trees, the inability of translating them can be expected. The same holds for water surfaces, as can be seen in the right example of the original framework (fig. 4.4). The right part of the translated image does not represent the green area sufficiently and the transition from land to water is not rendered correctly. Looking closer at the left example in fig. 4.4 one can also find gaps in the streets that do not exist in the aerial image.
Although there are images which are translated poorly, the results overall suggest that the implementation works correctly.

Monitoring the training process shows that the discriminator saturates at around 100% accuracy from the second epoch on. This indicates that the adversarial training does not work as intended. Two possible reasons are the ratio of the generator resp. discriminator learning rates and the weighting between the adversarial and L1 objectives. Lowering the discriminator learning rate $\eta^{dis} = 10^{-4}$ or increasing the generator learning rate $\eta^{gen} = 2 \times 10^{-4}$ could thus be a valid solution, as well as adjusting the weighting value $\lambda = 100$ of the objective function (compare eq. 4.2). Therefore three additional training runs with with specifications as depicted below have been performed.

$$\begin{aligned}
&1)\ \eta^{dis} = 2 \cdot 10^{-5} \\
&2)\ \eta^{dis} = 2 \cdot 10^{-5}\ ,\ \eta^{gen} = 5 \cdot 10^{-4} \\
&3)\ \eta^{dis} = 2 \cdot 10^{-5}\ ,\ \eta^{gen} = 5 \cdot 10^{-4}\ ,\ \lambda = 10
\end{aligned} \tag{4.4}$$

For each run the discriminator accuracy saturates slower (roughly epoch 4, 6 and 10 for run (1), (2) and (3)) and the values differ more after the saturation point (as can be seen e.g. by a histogram analysis).

The visual quality is difficult to compare due to the many different test images and the variation in the training process. For the first two runs it does not deviate much, but the run with the adjusted weighting value yields clearly worse results. This shows the importance of the L1 term regarding the perceptual quality of generated images.

An additional test was performed to evaluate the generator architecture, especially the size. The smallest possible network input is 256 × 256 because the spatial downsampling layers will reduce this to 1 × 1 in the bottleneck layer. For the translation task considered here, the network needs to be capable of handling input of sizes down to 64×64 because the EuroSAT dataset consists of images of that size. Thus, two basic encoder and two basic decoder layers were removed before run (2) was repeated. Run (4) resulted in a slightly lower visual quality, but still produced reasonable results. The training behaviour in terms of the discriminator accuracy was similar to run (2).
Overall the training of the Pix2Pix GAN worked reasonably well. Since no training details were given by [13] it allows to get an idea how the training looks like for a working example and which problems can be expected to appear. Besides the saturating discriminator accuracy, which is most probably a consequence of the novel objective function, the tests also indicated instabilities in the training, which is exemplary shown in fig. 4.6 and appeared in all runs so far.

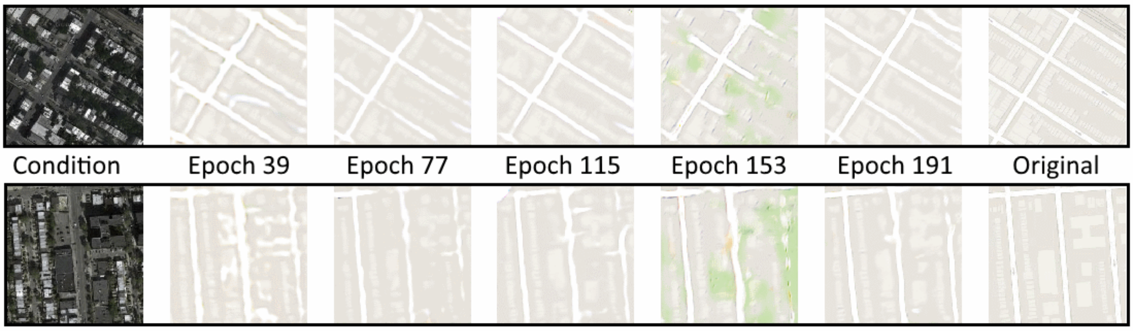


Figure 4.6: Exemplary courses of the training; left: the input image; right: the original image from the dataset; middle: results at different stages of the training.

As can be seen, the translation becomes better while training, but at epoch 153 it crashes, while working fine afterwards. This unstable training behaviour is especially problematic if it accidentally happens in the last stages of training.

4.3 Translate Optical to SAR Images

After gathering information about the training behaviour of the Pix2Pix framework, it is now applied to the Sen1-2 dataset.

Due to the huge number of 282.384 image pairs in this dataset, compared to the previously used set with 2.194 image pairs, only a subset is considered for the beginning. As the top level structure corresponds to the four meteorological seasons the smallest subset, namely summer, is picked. It consists of 49 globally distributed scenes adding up to 53.508 image pairs. Each scene is indexed by a specific number and represented by a directory.

The dataset structure allows different approaches on how to select images for the training and how to divide them in training resp. test sets. The splitting can be done either by scenes i.e. using some scenes for training and others for testing or by dividing each scene and merge the first part of each scene to generate the training- and the second part to generate the test set. Since the first approach prevents the training and test set from sharing information (due to the overlap of images as explained in section 2.3.2), it seems to be the go to method. However, the variance of the dataset is very high, so one scene maybe contains forest, another urban areas and water surfaces and the next desert. If one area type is not included in the training set, the algorithm can not be expected to handle it correctly. For the initial experiments the second approach is therefore chosen. To reduce the effect of shared information, the selected images are augmented by mirroring and rotation.

As one of the training runs of the previous section took roughly 4 hours on a Nvidia Tesla P100 GPU using below 1100 training image pairs, the usage of the complete summer subset is not feasible. To run multiple experiments for testing the influence of different training modalities, 10 scenes are pseudo-randomly selected for the beginning, such that different landscape types are present. Each scene is randomly split into 80% training and 20% test data. From the generated training set 75% are omitted to arrive at roughly 2000 training and test samples. The training of the GAN is executed with the specification of run (4) from the previous section.

Fig. 4.7 shows example triplets of the condition, the generated and the original image.

Although the generator has no problems to capture the borders and edges correctly, the comparison with the real images shows that the existing noise is not present. This becomes especially evident for smooth surfaces like water bodies. The noise, which is called speckle, is a common phenomenon in radar imagery. Hence it is an important property of the images and not generating it would result in unrealistic and therefore useless artificial training images. Besides the speckle, small structures

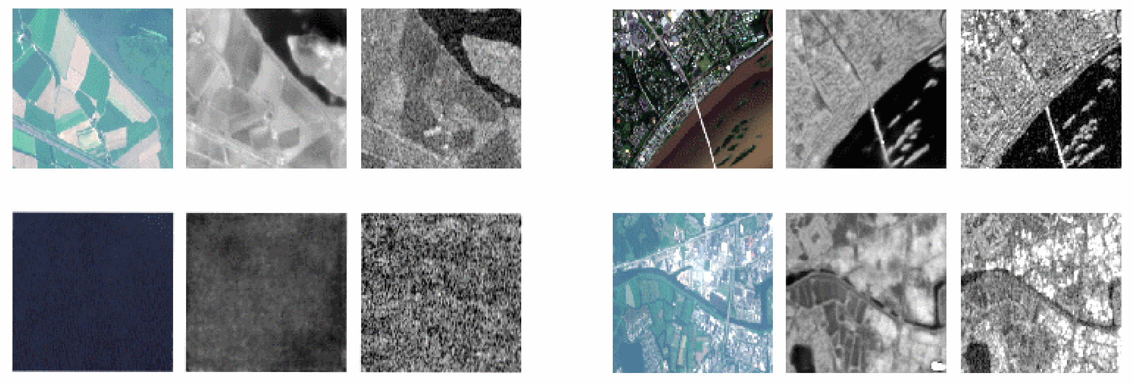


Figure 4.7: Four example results for the generator trained with a subset of the Sen1-2 dataset in run configuration (4). Each triplet consists of condition (left), generated image (middle) and original image (right).

appear less sharp as can be seen in the urban areas in the right images in fig. 4.7. The brightness on the other hand is chosen correctly. In the top left triplet one can see that the brightness for different crops resembles the SAR characteristics in contrast to the brightness of the optical image. The same holds for water surfaces that differ in color for the different optical images, but are rendered dark for all examples in fig. 4.7.

As already seen in the baseline tests, the discriminator saturates to 100% accuracy quickly i.e. in the 7^{th} epoch. This also becomes manifest in the adversarial loss that is growing instead of decreasing as the L1 loss does. Since the saturation is expected to hinder at least the adversarial part of the training, the same run is performed with λ from eq. 4.2 set to 10 instead of 100.

In opposition to the baseline tests, this procedure improves both the training behaviour and the perceptual quality of the images for the Sen1-2 dataset. Both losses decrease in the curse of the training and the discriminator accuracy converges to roughly 60% as can be seen in fig. 4.8.

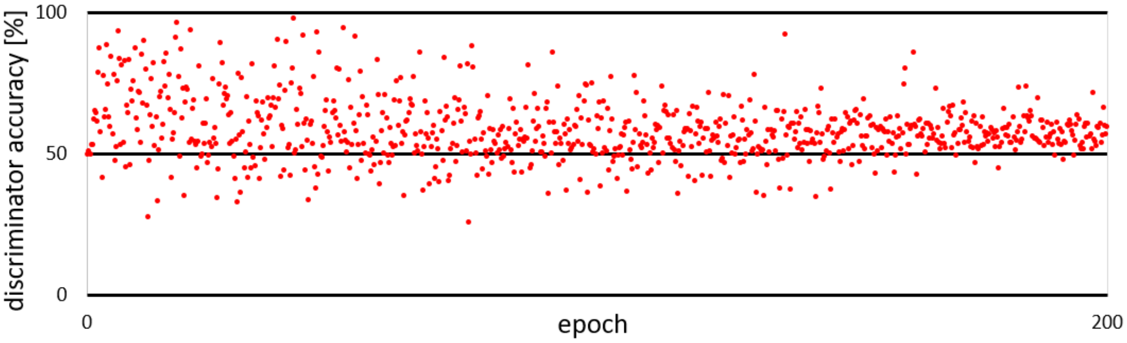


Figure 4.8: Discriminator accuracy while training the GAN with $\lambda = 10$.

This resembles the ideal training scenario described in section 3.5 where the discriminator becomes less and less certain which images are real and fake. In the ideal case the accuracy would converge to 50%, but the objective function contains an L1 term here which influences the adversarial process, especially for $\lambda > 1$. The improvement of the training behaviour goes along with an enhanced similarity between generated and original SAR images. Fig. 4.9 shows examples for comparison with fig. 4.7.

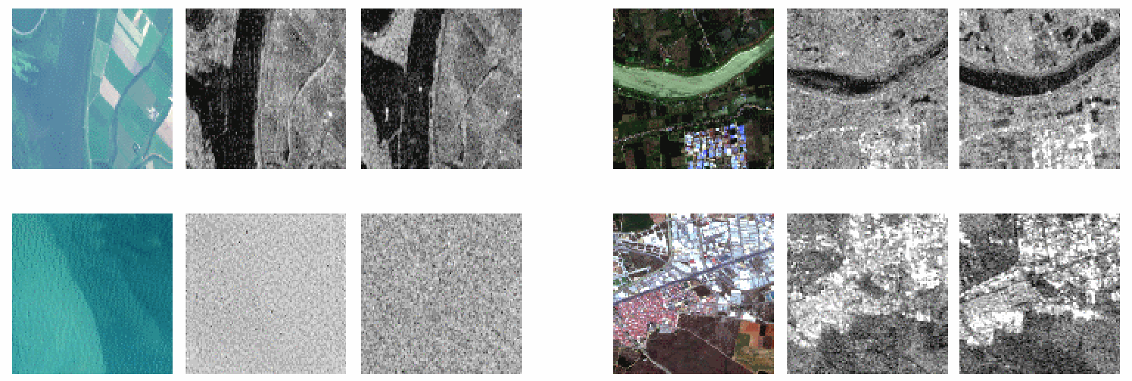


Figure 4.9: Four example results for the generator trained with a Sen 1-2 subset with λ changed to 10 from 100, i.e. the adversarial objective becoming more important. (condition: left, generated: middle, original: right)

While original and generated images could be easily distinguished in fig. 4.7 mostly due to the missing speckle noise, this is not the case in fig. 4.9. Regarding for example the top left triplet, an observer can hardly tell which image is generated and which is real. The sea surface in the lower left shows that the speckle although present, differs from the original noise. Still the indifference regarding the different bright areas in the condition is captured correctly. The top left triplet shows that agricultural areas are still captured correctly when comparing to the real image, whereas the top right image is an example for differences in urban and water areas. Over all there are still differences in small structures that become apparent in the urban area depicted in the bottom right image.

As the lack of speckle noise seems to be resolved with the adjustment of the weighting value λ, the question arises if this holds for more (diverse) input data, i.e. if the algorithm is scalable. Therefore the same training run is repeated using 20 instead of 10 different scenes which results in roughly double the training data. While deep learning algorithms usually benefit from additional training data, the increased variance can also be challenging. In this case, the training becomes less convergent as can be seen in the diagram in fig. 4.10.

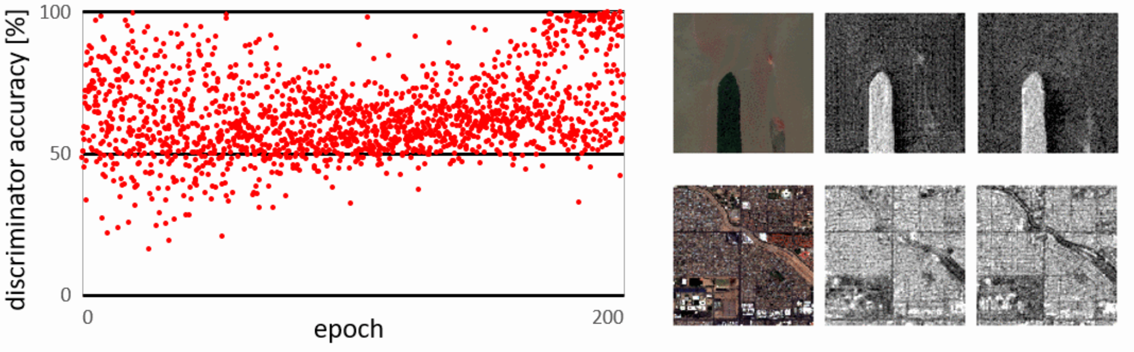


Figure 4.10: Results of increasing the amount of training data. Left: discriminator accuracy while training the GAN. Right: Example translation results (condition: left, generated: middle, original: right)

For the first half of the training phase the discriminator accuracy behaves like before but afterwards differs more. This implies that the discriminator is able to better distinguish real and generated samples i.e the generator is unable to produce convincing images. Looking at the variety of images it is hard to tell whether their quality changed. The lower triplet in fig. 4.10 gives an example of a convincing urban area but differs from the real image in terms of the river. In the upper triplet the speckle noise covering the water body seems to be more structured than on the real image, pointing towards possible generation artifacts. On the other hand many images show the same visual quality as before.

For this case in particular, but for image translation in general a metric that allows to quantify the quality of translation processes would be necessary. However, the evaluation of synthesized images is still a challenging and open problem [13]. One part of the problem is already the exact formulation of the objective. As mentioned when evaluating the baseline tests, one important point is the comparison with the real image. A traditional way of quantifying this is the usage of per pixel losses like L1 or L2. Often reproducing the exact value of a specific pixel is not the goal though. For GANs in general the task is to match distributions, not specific examples. For optical to SAR translation in particular the inherent speckle noise also hinders the per pixel evaluation. The second important characteristic of well translated images is the plausibility. Comparing the output of the generator to the input, does it match the expectations? Regarding the image to map translation task one may ask if the streets are rendered white, straight and without gaps. For the optical to SAR translation one expects water bodies to be dark, urban areas to be bright and with sharp edges, the existence of speckle noise etc. These properties are dependant on the specific task i.e. require knowledge, which makes it difficult to quantify or evaluate them in an automated manner.

In [13] two methods based on the second objective are proposed. The first is to run perceptual studies on Amazon Mechanical Turks. This means showing generated and real images to different people under certain conditions, let them evaluate if the presented image is real and counting how often the generator is able to fool them. It is basically the same task the discriminator fulfills, but with human observers.
The second approach is to train artificial neural networks on classifying or detecting objects in real images and then apply them to the generated samples and compare the outcome to their results on real images. This approach is based on the idea that the classifier or detector will produce the same result if the generated images are similar to the real ones.
For the optical to SAR translation the second approach can not be applied, since the lack of labeled SAR data and thus operating classifiers was the initial reason for this study. The first approach works best if many human observers are judging many samples. Additionally they need at least basic knowledge of the evaluation of SAR images. As these resources are not available for this study, the initial evaluation is based on the perceptual quality of the images as described so far.
The final objective of this work is to generate an artificial SAR dataset allowing the training of classifiers that work well on unseen data. Thus, the perceptual quality of the translated EuroSAT dataset and the accuracy of the trained classifiers will be the metric used here.

Translating the EuroSAT dataset requires the application of the generator to smaller images (64 × 64 pixels). Since the Pix2Pix generator exclusively consists of convolutional, pooling and upsampling layers, it can be utilized for other sizes than it has been trained on. Before this is done, the quality of the translation is evaluated on the subset the generator was trained and tested on.
The training instabilities mentioned at the baseline tests are very relevant here, as the generator output differs heavily while training. This is already indicated by fig. 4.8 (although converging, the discriminator accuracy fluctuates between 50% and 70%) and also becomes apparent by viewing the networks output for the same condition over several epochs.
Multiple network configurations are stored and compared to pick the best possible generator weights. Using this configuration, the generator is applied to the same image divided in subsamples of 64 × 64 pixels. Fig. 4.11 shows example images compared with the generator applied to the standard image size of 256 × 256 pixels.

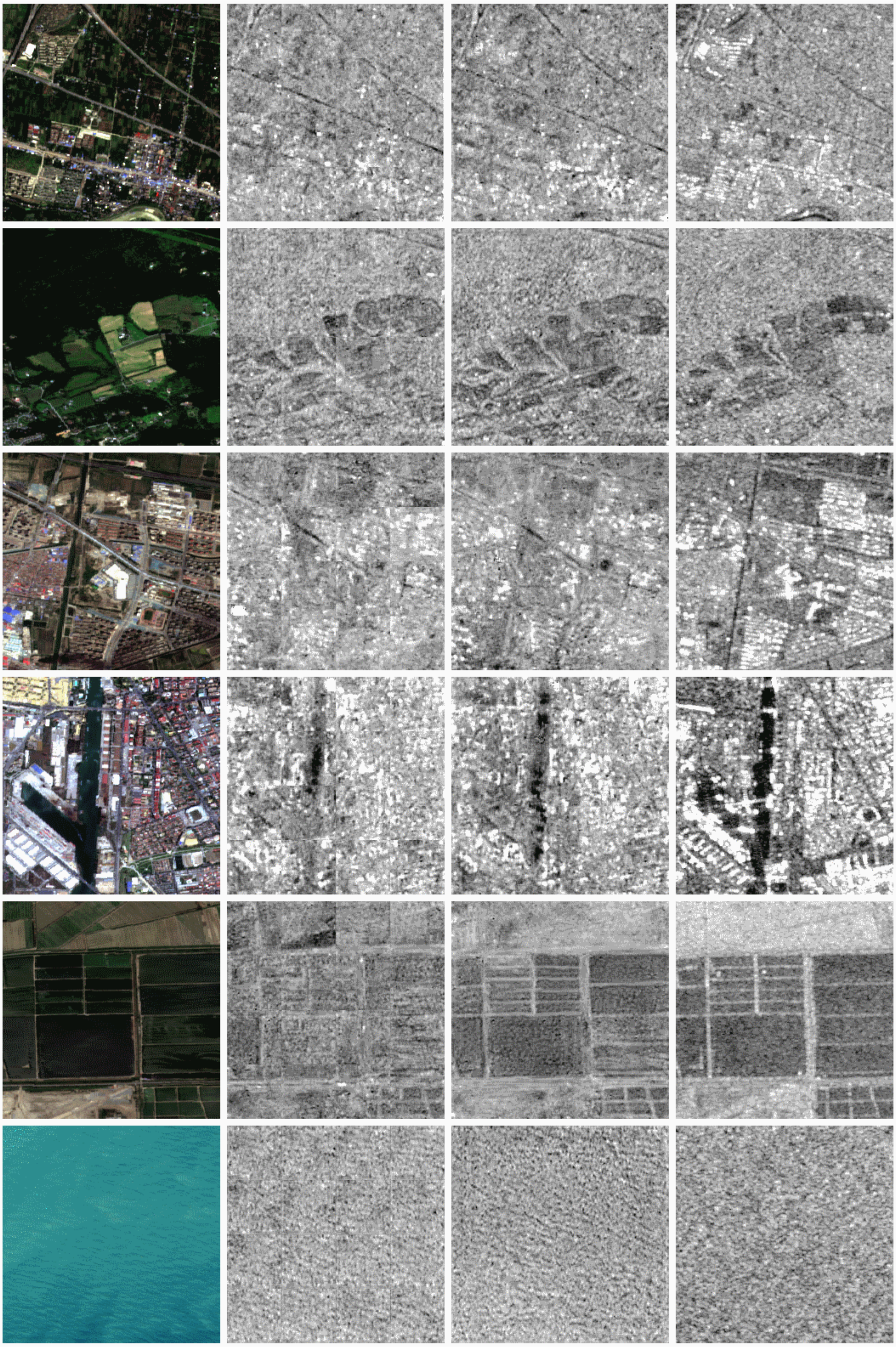


Figure 4.11: Test of smaller input size and comparison to the standard input size. From left to right: condition, 64 × 64 input (rearranged), 256 × 256 input, real image

All sixteen 64×64 output samples are rearranged to correspond to the 256×256 output images. Presenting the 64×64 output in the second columns it becomes apparent in all six examples that artifacts are introduced at the borders of the small patches. These are bigger differences in neighbouring pixels than usual which are introduced by the limited spatial information provided to the generator in case of smaller inputs. The artifacts differ in their intensity dependent on the landscape type and the contrast present at intersections but are usually not problematic if only the small images are regarded. In the first two rows in fig. 4.11 well translated examples for both modalities are presented. The main spatial features are present, the overall brightness corresponds to the real sample and the artifacts barely influence the visual quality of the complete image. Even small features like the highways in the first examples are recognized and translated.
Examples 3 and 4 on the other hand show poorly translated images. The representation of the urban areas in example 3 lacks details and is overall very blurry for both modalities. Regarding the fourth example the same holds and additionally the water surface is underrepresented by the 256×256 generator and hardly visible for the 64×64 generator. Example 5 shows the case of the 256×256 generator working way better than the 64×64 generator. Although this does not happen often in this explicitness, the behaviour appears for images containing structures with high level contrasts, i.e. on scales that can't be captured by the small patches. Lastly the sea surface can be reproduced successfully by both generator types.

These examples are shown to illustrate that both, the difference in quality between real and generated samples and the difference between the translation modalities vary substantially. As the generator with the smaller receptive field receives less high level information, it can not be expected to provide the same quality. Still in most of the cases the visual quality is only slightly reduced compared to the 256×256 generator.
Therefore the evaluation of a GAN trained on images of size 64×64 is omitted here. This is also motivated by the difficulty of comparing samples of this small size which tends to be more subjective.

Chapter 5

An Artificial Dataset for SAR Classification

Two generator weight configurations are used for the translation of the EuroSAT dataset. Besides the generator configuration previously used for the analysis of the 64 × 64 image translation (called $G^{(10)}$ here), the generator trained on 20 scenes ($G^{(20)}$) was stored after 100 epochs, i.e. before the discriminator accuracy diverged (compare fig. 4.10). It will be regarded to compare the visual quality and the classification results.

Applying the generator with both weight configurations results in two datasets that can be utilized for the training of a classifier. Example images from these datasets are shown in fig. 5.1.

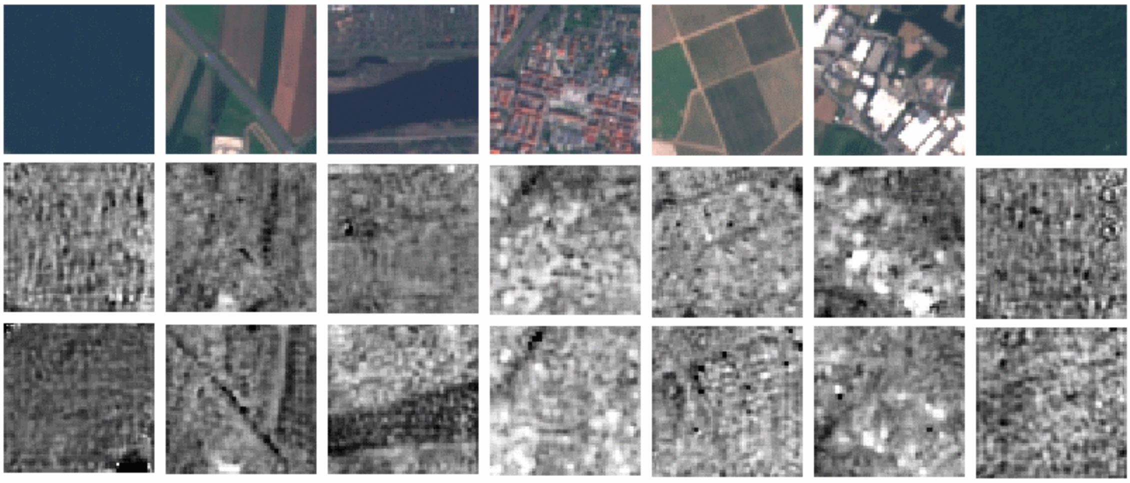


Figure 5.1: Example images from the translated EuroSAT dataset; top: original images; middle: translation with generator trained on 10 Sen1-2 scenes; bottom: translation with generator trained on 20 scenes.

As already mentioned, evaluating the perceptual quality of samples of small size is very challenging. For the training task the presence of differentiating features and structures is mandatory. Nonetheless inherent attributes of SAR images like speckle, which is contradictory to clear features, is required to train a classifier applicable to non-artificial data.

The presence of features is more explicit in the dataset generated with $G^{(20)}$. This can be seen e.g. in the second and third column in fig. 5.1 that display examples of the highway resp. river class. Both key features appear more clear in the bottom row i.e. for $G^{(20)}$. Like already observed, urban areas (represented in the fourth resp. sixth column by images from the residential resp. industrial class) appear blurry. In addition it can be seen that the difference between the sea / lake (left) and the forest class (right) are relatively small, even in the optical dataset. For $G^{(20)}$ artifacts sometimes appear as shown in the bottom left image.

These observations will influence the training of a classifier which can be seen later on. The classification algorithms utilized, as well as the comparison of their performances is described in the upcoming section.

5.1 Training the Classification Algorithms

Remote Sensing data analysis using deep neural networks is usually performed by selecting a pre-trained network and fine-tuning it with the investigated imagery (e.g. [2]). This practice was introduced due to the small number of labeled images compared to datasets like ImageNet. Classification algorithms that won the ImageNet Large-Scale Visual Recognition Challenge are common choices in the area of remote sensing, even though they were trained on completely different data. In this case the basic network structure in conjunction with the pre-trained weights is adopted and only the final classification layers are retrained [2].

The VGG19 network introduced in [5] and the ResNet50 network [7] are common choices and therefore applied here.

One main feature of the VGG19 network is the simplicity of its structure. It consists of sixteen convolutional layers with a receptive field of 3×3 and increasing number of filters. Maximum pooling layers are added in between for spatial downsampling. After the final convolution and pooling, three fully connected layers are added with the last layers number of neurons corresponding to the number of classes i.e. 1000 in the original specification. All layers apply ReLU activation, except for the final layer that uses softmax. When reusing the VGG19 network, the three fully connected layers are usually retrained, with the other network weights remaining constant.

With 50 layers, the ResNet50 includes considerably more layers than the VGG19 network. To bypass the training difficulties of such a deep network the residual block described in section 3.3 and shown in fig. 3.4 is used as basic unit of the network. The exact specifications can be seen in [7], where also a comparison to the VGG19 network is illustrated. In contrast to the VGG19 network the spatial downsampling is performed by strided convolutions rather than pooling except for the first convolution that is followed by a maximum pooling layer and the last convolution is followed by an average pooling layer. For downsampling a so called projection shortcut block is used where the first weight layer uses a strided convolution and the identity function is also replaced by a strided convolution to match the spatial output dimensions. Batch normalization is applied after every convolutional layer. Subsequent to the average pooling layer one fully connected layer is appended for the mapping to the number of classes. Similar to the VGG19 network the retraining is usually performed on this very last layer (e.g. [2]).

Before the networks are trained with the artificial SAR dataset, their training results on the optical EuroSAT dataset as existing are tested for comparison later on.
All experiments on both the optical and SAR datasets are executed identically. Initially the network type is chosen, i.e. VGG19 or ResNet50. For the ResNet50 the final layer is replaced by a fully connected layer with 10 neurons corresponding to the 10 landcover classes. The final fully connected layers of the VGG19 network are changed to contain 512, 512 and 10 neurons as the original 4000 neurons in the third and second to last layers are computationally very expensive and only 10 instead of 1000 classes are present. Additionally dropout with a rate of 30% was added since first experiments did show that the network tends to overfit.
The dataset is divided into 80% training, 10% validation and 10% test data. This differs from the division into 80% training and 20% as performed in [17], the paper associated with the EuroSAT dataset. The reason to use a validation set is that each experiment consists of multiple training and evaluation runs. In each of these runs the network configuration with the best accuracy on the validation set is stored and later on evaluated using the test set to avoid cheating.
After loading the datasets, the input images are linearly transformed from values in [0, 255] to [-1, 1]. The training set is shuffled and reflections and rotations are applied. Since the input to the classifier needs to contain three channels which is fulfilled for the optical, but not for the SAR images, the latter are transformed to three channel images by copying each pixel value. The proper functioning of this method has been tested beforehand for both networks using gray-scale images.

The training is performed applying the Adam optimizer with a learning rate of 0.001 and its parameters initialized to $\beta_1 = 0.9$ and $\beta_2 = 0.999$. As loss function the cross entropy [26] defined below is selected (nomenclature as in eq. 3.1).

$$C_x(w,b) = -\frac{1}{N}\sum_x [y \cdot \ln(\hat{y}) + (1-y) \cdot \ln(1-\hat{y})] \tag{5.1}$$

The networks are trained for 20 epochs with a batch size of 50. All experiments include 5 runs. For each run the accuracy and loss on the test set are stored as well as the confusion matrix, which displays the networks performance regarding single classes. At the end the mean and standard deviation are calculated.

Training the VGG19 and the ResNet50 network on the optical EuroSAT dataset yields the results shown in tab. 5.1.

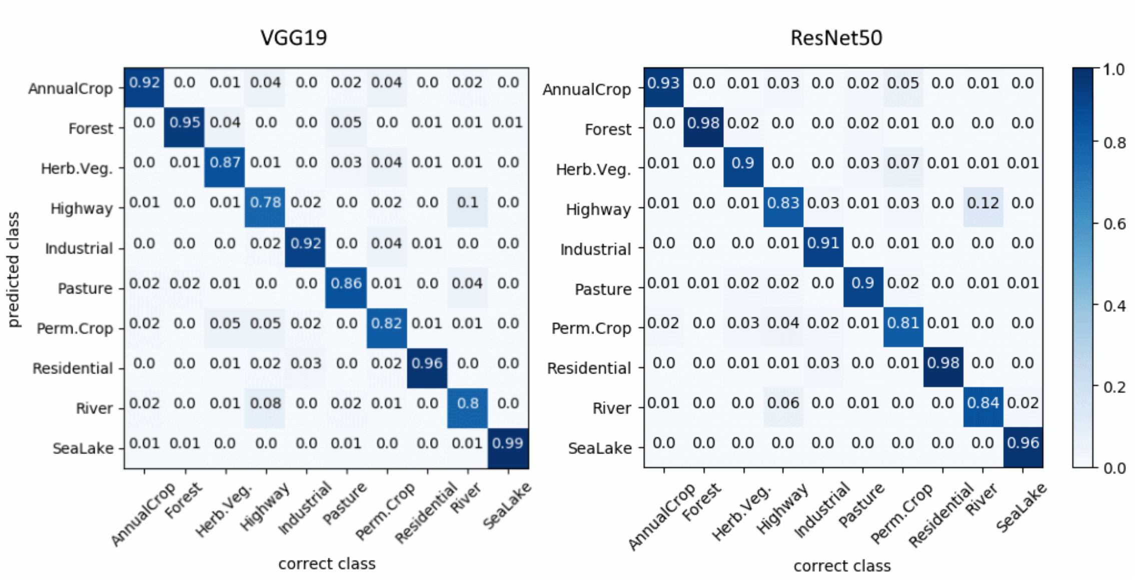
dataset	accuracy: VGG19	accuracy: ResNet50
optical	(89.2 ± 0.2) %	(90.8 ± 0.3) %

Table 5.1: Test accuracy for both classifiers trained on the optical EuroSAT dataset

The achieved classification accuracy is roughly 90% for both networks. While the ResNet50 performs slightly better, it yields clearly worse results than the ones presented in [17]. For the RGB version of the dataset their benchmark is 98.57%. There are only minor differences between the training routines previously described and those presented in the paper. Besides the difference in splitting the dataset into training, validation and test set, the learning rate was initially set to 0.01 and later on changed to a value between 0.001 and 0.0001 for fine-tuning in [17]. The exact procedure, as well as the choice of the optimization algorithm and the cost function are not stated. Testing different optimization algorithms and cost functions did not improve the results here.

More insight than the overall classification accuracy is provided by the confusion matrices for both networks shown in fig. 5.2.

To generate the confusion matrix, the classifier is applied to all classes of the dataset one by one. The number of all predictions for one class, divided by the number of the training samples, is shown in one column of the matrix, i.e. the values add up to one and correspond to probabilities. A confusion matrix thus allows to evaluate the strengths and weaknesses of a classifier. High diagonal values as in both matrices in fig. 5.2 match the overall high accuracy on the test data. Apparently both classifiers have the most issues with the classes 'Highway', 'River' and 'Permanent


VGG19

	AnnualCrop	Forest	Herb.Veg.	Highway	Industrial	Pasture	Perm.Crop	Residential	River	SeaLake
AnnualCrop	0.92	0.0	0.01	0.04	0.0	0.02	0.04	0.0	0.02	0.0
Forest	0.0	0.95	0.04	0.0	0.0	0.05	0.0	0.01	0.01	0.01
Herb.Veg.	0.0	0.01	0.87	0.01	0.0	0.03	0.04	0.01	0.01	0.0
Highway	0.01	0.0	0.01	0.78	0.02	0.0	0.02	0.0	0.1	0.0
Industrial	0.0	0.0	0.0	0.02	0.92	0.0	0.04	0.01	0.0	0.0
Pasture	0.02	0.02	0.01	0.0	0.0	0.86	0.01	0.0	0.04	0.0
Perm.Crop	0.02	0.0	0.05	0.05	0.02	0.0	0.82	0.01	0.01	0.0
Residential	0.0	0.0	0.01	0.02	0.03	0.0	0.02	0.96	0.0	0.0
River	0.02	0.0	0.01	0.08	0.0	0.02	0.01	0.0	0.8	0.0
SeaLake	0.01	0.01	0.0	0.0	0.0	0.01	0.0	0.0	0.01	0.99

ResNet50

	AnnualCrop	Forest	Herb.Veg.	Highway	Industrial	Pasture	Perm.Crop	Residential	River	SeaLake
AnnualCrop	0.93	0.0	0.01	0.03	0.0	0.02	0.05	0.0	0.01	0.0
Forest	0.0	0.98	0.02	0.0	0.0	0.02	0.01	0.0	0.0	0.0
Herb.Veg.	0.01	0.0	0.9	0.0	0.0	0.03	0.07	0.01	0.01	0.01
Highway	0.01	0.0	0.01	0.83	0.03	0.01	0.03	0.0	0.12	0.0
Industrial	0.0	0.0	0.0	0.01	0.91	0.0	0.01	0.0	0.0	0.0
Pasture	0.01	0.01	0.02	0.02	0.0	0.9	0.02	0.0	0.01	0.01
Perm.Crop	0.02	0.0	0.03	0.04	0.02	0.01	0.81	0.01	0.0	0.0
Residential	0.0	0.0	0.01	0.01	0.03	0.0	0.01	0.98	0.0	0.0
River	0.01	0.0	0.0	0.06	0.0	0.01	0.0	0.0	0.84	0.02
SeaLake	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.96

Figure 5.2: Confusion matrices for both classifiers trained on the optical dataset.

Crop'. While the latter is confused with many other classes, 'Highway' and 'River' tend to be mixed up with each other. This result is conform with the evaluation in [17]. In more than 10% of the cases a river is erroneously classified as highway for both classifiers. Overall only small differences between the accuracies of single classes are present for the networks. This indicates that the varying quality between classes is mainly introduced by the dataset. Since the classes are well balanced in terms of the number of samples, shared features, e.g. the presence of border lines in the case of highways and rivers, are a probable reason.

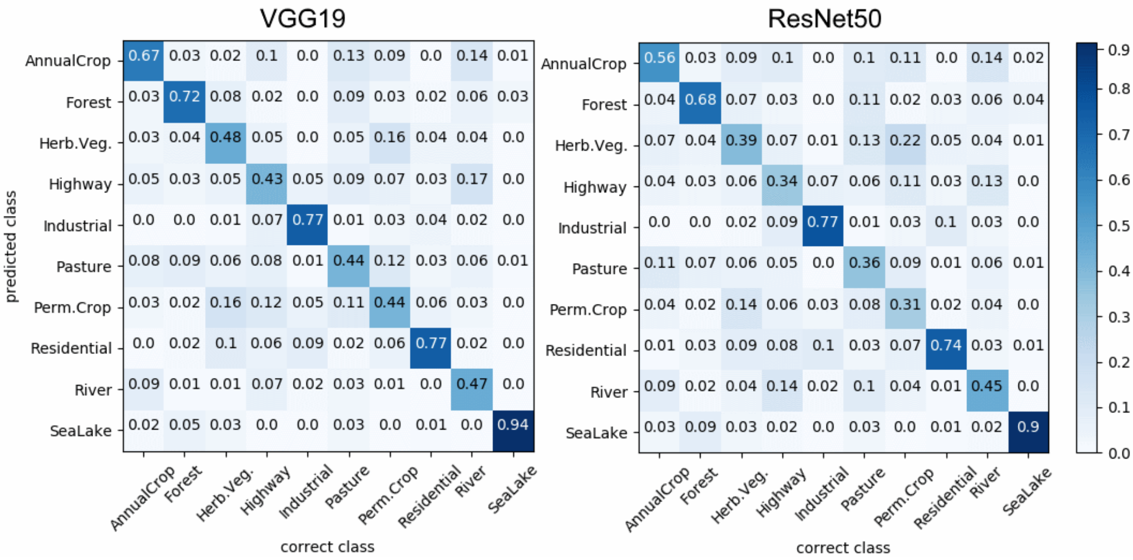
The training runs are repeated for the two versions of the translated EuroSAT dataset i.e. translated by the generator $G^{(10)}$ trained on 10 different Sen1-2 scenes and the second generator $G^{(20)}$ trained on 20 scenes. Tab. 5.2 shows the results in terms of the classification accuracy of both classifiers trained on both datasets.

dataset	accuracy: VGG19	accuracy: ResNet50
SAR $G^{(10)}$	(56.3 ± 0.5) %	$(49,8 \pm 0.5)$ %
SAR $G^{(20)}$	(62.5 ± 0.5) %	(56.4 ± 0.3) %

Table 5.2: Test accuracy for both classifiers trained on the translated datasets

All results are substantially worse than on the optical dataset. In contrast to the initial experiments the VGG19 network performs better on the SAR dataset than the ResNet50. The training behaviour of the classifiers also vary. While the validation accuracy of the VGG19 network increases more steady, never deviating much from the training accuracy, the validation accuracy of the ResNet50 saturates quickly (usually after 5 epochs) as against the training accuracy that still increases in the course of the training. The latter point is a clear indicator that overfitting occurs which is hindered for the VGG19 network by the usage of multiple fully connected layers combined with dropout.

Overall the experiments show better results for the dataset translated with $G^{(20)}$. This corresponds to the observation that differentiating features are more present on this dataset.


VGG19

	AnnualCrop	Forest	Herb.Veg.	Highway	Industrial	Pasture	Perm.Crop	Residential	River	SeaLake
AnnualCrop	0.67	0.03	0.02	0.1	0.0	0.13	0.09	0.0	0.14	0.01
Forest	0.03	0.72	0.08	0.02	0.0	0.09	0.03	0.02	0.06	0.03
Herb.Veg.	0.03	0.04	0.48	0.05	0.0	0.05	0.16	0.04	0.04	0.0
Highway	0.05	0.03	0.05	0.43	0.05	0.09	0.07	0.03	0.17	0.0
Industrial	0.0	0.0	0.01	0.07	0.77	0.01	0.03	0.04	0.02	0.0
Pasture	0.08	0.09	0.06	0.08	0.01	0.44	0.12	0.03	0.06	0.01
Perm.Crop	0.03	0.02	0.16	0.12	0.05	0.11	0.44	0.06	0.03	0.0
Residential	0.0	0.02	0.1	0.06	0.09	0.02	0.06	0.77	0.02	0.0
River	0.09	0.01	0.01	0.07	0.02	0.03	0.01	0.0	0.47	0.0
SeaLake	0.02	0.05	0.03	0.0	0.0	0.03	0.0	0.01	0.0	0.94

ResNet50

	AnnualCrop	Forest	Herb.Veg.	Highway	Industrial	Pasture	Perm.Crop	Residential	River	SeaLake
AnnualCrop	0.56	0.03	0.09	0.1	0.0	0.1	0.11	0.0	0.14	0.02
Forest	0.04	0.68	0.07	0.03	0.0	0.11	0.02	0.03	0.06	0.04
Herb.Veg.	0.07	0.04	0.39	0.07	0.01	0.13	0.22	0.05	0.04	0.01
Highway	0.04	0.03	0.06	0.34	0.07	0.06	0.11	0.03	0.13	0.0
Industrial	0.0	0.0	0.02	0.09	0.77	0.01	0.03	0.1	0.03	0.0
Pasture	0.11	0.07	0.06	0.05	0.0	0.36	0.09	0.01	0.06	0.01
Perm.Crop	0.04	0.02	0.14	0.06	0.03	0.08	0.31	0.02	0.04	0.0
Residential	0.01	0.03	0.09	0.08	0.1	0.03	0.07	0.74	0.03	0.01
River	0.09	0.02	0.04	0.14	0.02	0.1	0.04	0.01	0.45	0.0
SeaLake	0.03	0.09	0.03	0.02	0.0	0.03	0.0	0.01	0.02	0.9

Figure 5.3: Confusion matrices for both classifiers trained on the $G^{(20)}$ dataset.

Regarding the confusion matrices of the $G^{(20)}$ dataset depicted in fig. 5.3 one key insight can be drawn from the diagonal elements. The classes can be divided into two groups, the first is poorly classified, whereas the networks performance on the second is not far worse compared to the optical dataset. The 'Sea/Lake' class was nearly perfectly classified using the optical dataset. For the SAR $G^{(20)}$ dataset it is still correctly classified in 90% resp. 94% of the cases. Also the 'Industrial' and 'Residential' i.e. the urban classes are properly assigned roughly 75% of the time. Although this is worse than for the optical dataset, most of the wrongly classified 'Industrial' images are of the class 'Residential' and vice versa, i.e. falling into the same category. The 'Forest' class with around 70% accuracy is also relatively well distinguishable but sometimes confused with other vegetation types.

The poorly predicted classes are 'Herbaceous Vegetation', 'Pasture', 'Permanent Crop', 'Highway' and 'River'. The last three were already the more problematic classes in the optical dataset. Like before river and highway are mixed up often, but also the confusion with other classes is present. Although the human observer can be expected to distinguish them clearly due to the characteristic presence of two borders incorporating a different surface, it has to be mentioned that this feature is often only present at a small part of the image regarding e.g. the number of pixels actually representing the highway or the river. This means that often greater part of the image corresponds to another class. In the ImageNet challenge the possibiltiy of assigning multiple classes to an image is also present and tackled by considering a top five error which is not reasonable here due to the small numbers of classes. Also the orientation and location of the primary features are arbitrary which increases the difficulty of generating appropriate feature maps. While at least the surface inbetween the two borders can be relatively well distinguished in optical images due to different colors, this is not necessarily the case for single polarization SAR images. This effect is amplified by the suboptimal translation of the dataset. The influence of the translation is fortified by the fact that the ResNet50 resp. VGG19 networks only achieve 21% resp. 27% accuracy on the highway class using the SAR $G^{(10)}$ dataset.

For additional comparison the classifiers are trained without pre-trained weights on the $G^{(20)}$ dataset. The optimization process did not lead to improvements for the VGG19 network as the accuracy stagnated on roughly 10%, corresponding to random guessing. However, the VGG19 network is known to be hard to train, in [5] it was therefore initialized with the weights of the trained VGG16 network, which is not practicable here. The training of the ResNet50 on the other hand did work well, although overfitting did appear. Training for 30 instead of 20 epochs, the network was able to achieve $(66.6 \pm 0.7)\%$ accuracy. Applying dropout (30%) between the convolutional layers of the basic building blocks led to a slightly improved accuracy of $(68.2 \pm 1.1)\%$. Qualitative the confusion matrix resembles the ones shown for the pre-trained networks. Besides the 'Highway' class the accuracy increased for all classes. The training was overall less stable without pre-trained weights. Also this training modality is expected to be adapted more to the training data, as the feature maps are generated solely for this less general purpose compared to the pre-trained feature maps generated on data that differs from remote sensing imagery.

Depending on the training modality regarded, the classification accuracy on the SAR datasets is 20% to 30% worse compared to the networks trained on the optical dataset.

Three possible reasons for this difference come into consideration. First of all, optical and SAR imaging represents two different physical measurements. This general

difference can lead to one imaging technique yielding data that allows simpler classification. It is for example plausible that the existence of speckle noise is hindering the classification. The influence of this factor can not directly be estimated.
Secondly, the information content between the datasets differs. While the optical data includes 3 bands (i.e. RGB), only single polarization SAR data is available here. Even if the different imaging techniques would be similar in terms of the possible classification accuracy, the optical dataset potentially contains thrice the information of the SAR data. A feasible measure of the impact of the information content is the comparison of RGB and gray-scale optical data. Training the VGG19 network resp. ResNet50 with the same parameters that led to the outcome shown in tab. 5.1, results in an accuracy of $(85.3 \pm 0.4)\%$ resp. $(87.0 \pm 0.2)\%$. Using only one channel of information thus impairs the accuracy by roughly 4%, which is a relevant but still small part of the difference between optical and SAR classification.
Finally the translation process is a probable reason for the worse performance. As already discussed, differently trained generator networks translate (spatial) features better, i.e. more clear. Also the analysis in section 4.2 and 4.3 did show that small structures like buildings and other objects with straight edges are not translated perfectly.

The Pix2Pix paper was one of the primary elaborations on the usage of conditional GANs as image translators. One of its main contributions was to present a framework that can be used on a wide variety of tasks. This comes along with the drawback of the generator network struggling to produce sharp SAR images.
As this can possibly hinder the classification trained on top of the translated dataset, a framework that focuses more on the details when generating images is desirable. Since algorithms utilized in image super-resolution are build exactly for this purpose, the Enhanced Super-Resolution Generative Adversarial Network (ESRGAN) framework [15] is chosen to analyze if the translation can preserve more details and if that improves the classifiers performance.

The framework is briefly discussed with focus on the differences to the Pix2Pix GAN. Afterwards the quality of translation and performance of the classification tasks are presented and compared to the former results.

5.2 The ESRGAN Framework for Image Super-Resolution

In image super-resolution the objective is to build an algorithm capable of rendering an image with higher resolution than the given input image possesses. Typically the number of pixels is doubled for each spatial dimension. The application of neural networks i.e. GANs in this domain makes use of the same basic methods like supervised training, gradient descent and back-propagation described earlier. While high-resolution images and their resolution decreased counterparts are utilized for the training in super-resolution, only small structural changes have to be carried out to apply the framework to image translation.

The ESRGAN framework is an improved version of the Super-Resolution (SR)GAN [45]. As fundamental units the generator and discriminator are working in an adversarial manner again.
In its basic configuration the discriminator is built analogically to the Pix2Pix discriminator. Convolutional layers that incorporate batch normalization and leaky ReLU functions are stacked together. Roughly double the number of these basic layers are used compared to the previous discriminator with spatial downsampling applied only every second layer. The exact specifications can be viewed in [45].
The contribution of [15] to the discriminator is the introduction of the relativistic generator based on the work in [46]. A relativistic discriminator estimates the probability that a real image is relatively more realistic than a fake image and a fake image is less realistic than a real image in contrast to the standard discriminator that only predicts if an input image is real. In the case of the ESRGAN the relativistic average discriminator [46] is used which relates the input image to the average of the current batch instead of one random sample.

An essential difference is the generator architecture. While the Pix2Pix generator represents a U-Net structure, the SRGAN generator is a residual network, the ESRGAN framework even uses a residual in residual block as basic unit, as can be seen in fig. 5.4.
Besides some initial and closing convolutional layers and the upsampling layer, the network consists of stacked basic units. For the SRGAN residual blocks similar to the one shown in fig. 3.4 are proposed. By contrast in [15] is argued that batch normalization is not improving the performance but introducing artifacts to the training, which was also shown in the context of fig. 5.1. Therefore they propose to

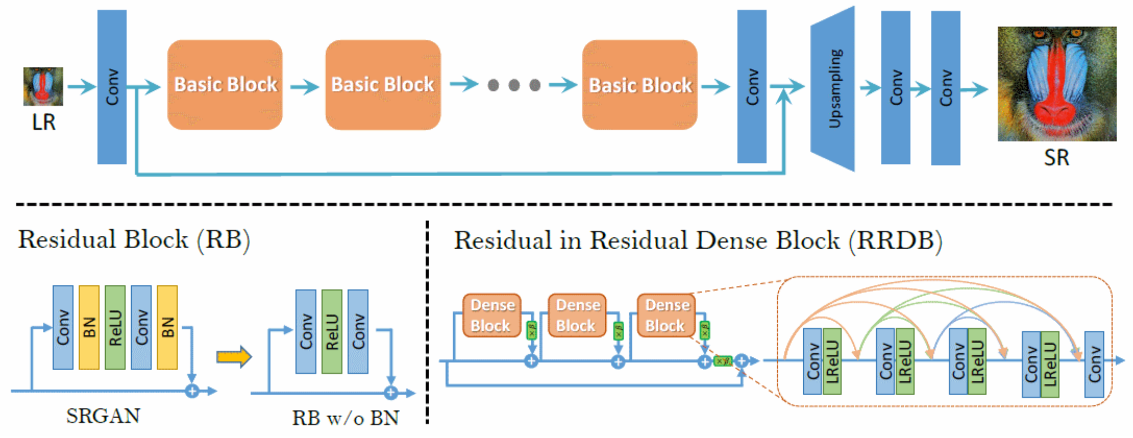


Figure 5.4: The ESRGAN generator structure; top: the basic architecture as proposed in [45]; bottom left: the basic residual block with and without batch normalization; bottom right: the residual in residual block as proposed in [15].

omit batch normalization and also introduce a deeper structure for the basic block shown in the bottom right part of fig. 5.4.
As no upsampling is required for the image translation task, the corresponding layer is simply omitted.
The details of the implementation correspond to the descriptions in [45] and [15] and are therefore not shown in detail here.

Finally the loss function differs from eq. 4.2 as an additional term referred to as perceptual loss is added. This loss corresponds to the activations of the final convolutional layer of an VGG19 network which are compared for real and fake images. As per [45] this measure is closer to perceptual similarity compared e.g. to a pixel-wise loss. The objective function therefore contains three summands, the relativistic averaged adversarial loss L_{adv}^{rel}, the L1 loss and the perceptual loss L_{perc} which are balanced by weighting factors as shown in eq. 5.2.

$$L_{p2p} = \alpha L_{perc} + \beta L_{adv}^{rel} + \gamma L_{L1} \tag{5.2}$$

The primary drawback of the ESRGAN architecture is the size of the model, mainly introduced by the new generator structure and the objective function forcing the algorithm to include a large part of the VGG19 structure. This limits the training on the Nvidia Tesla P100 GPU. For applying the network to the same number of images (compare section 4.3) the image size has to be reduced to 128×128 pixels for 2000 training images (10 scenes) or 64×64 pixels for 4000 training images (20 scenes). As already discussed, limiting the GAN to train on smaller image patches can be expected to decrease the translation quality slightly. The same was found in [15] where images of 128 × 128 pixels are fed into the network.

Performing the first experiments and using $\alpha = 1$, $\beta = 5 \cdot 10^{-3}$ and $\gamma = 1 \cdot 10^{-2}$ as proposed in [15] did not yield visual pleasant results. Inspecting the training procedure revealed that the weighted losses differed by several numbers of magnitude. While the weighted perceptual loss was greater than 100, the weighted adversarial loss roughly equaled 1 and the weighted L1-loss was smaller than 10^{-2}. Choosing $\alpha = 1$, $\beta = 100$ and $\gamma = 500$ led to balanced losses for the whole training and improved the perceptual quality.
Several experiments were carried out to find the best network and training configuration based on the visual quality of the generated compared to the real SAR images. The best results were achieved by applying pre-training with only L1-loss for 5 epochs as proposed in the paper. Thereafter the GAN is trained for 40 epochs with the weights as given above and the learning rate of the discriminator resp. generator set to 10^{-5} resp. 10^{-4}. Also the image quality was found to be better when training the network with 128 × 128 pixels and accordingly images from 10 different scenes. Fig. 5.5 shows example translation results.

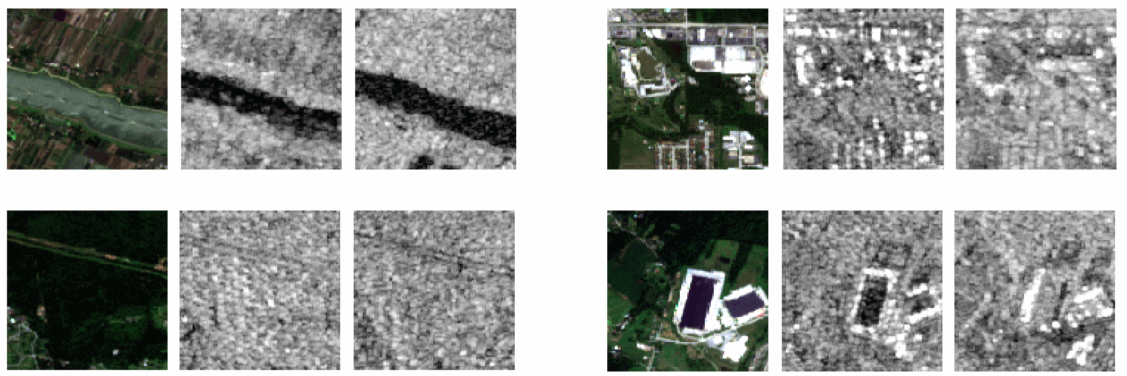


Figure 5.5: Four example results of the ESRGAN generator trained on 10 Sen1-2 scenes as described above (condition: left, generated: middle, original: right)

Overall the translation works well. The speckle noise is present and the generated images are mostly similar to the original ones. The bottom left image shows that even small structures are captured well. In contrast, the top left triplet depicts an example for differences in the clarity of fundamental features, i.e. the river being not as straight as in the original image. Additionally the crops above the river are displayed differently. Since they are not present in the real image, the generated sample appears to be closer to the condition. The examples on the right side indicate that small structures are rendered more accurate than for the Pix2Pix framework.

This is valid for both examples, the translated images are again more similar to the conditions than the originals which is also amplified by an increased contrast. Although this seems to be better regarding the clarity of the image, it is attended by the possible drawback of not representing real SAR images.

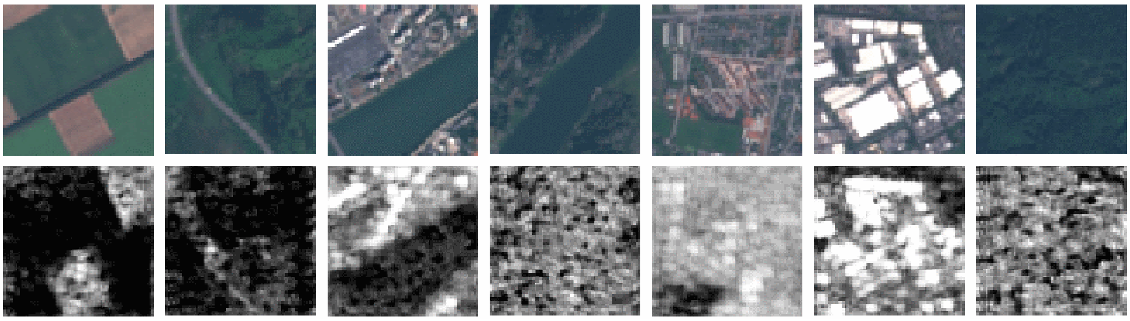


Figure 5.6: Example images of the EuroSAT dataset translated with the ESRGAN generator; top: original images, bottom: translated images.

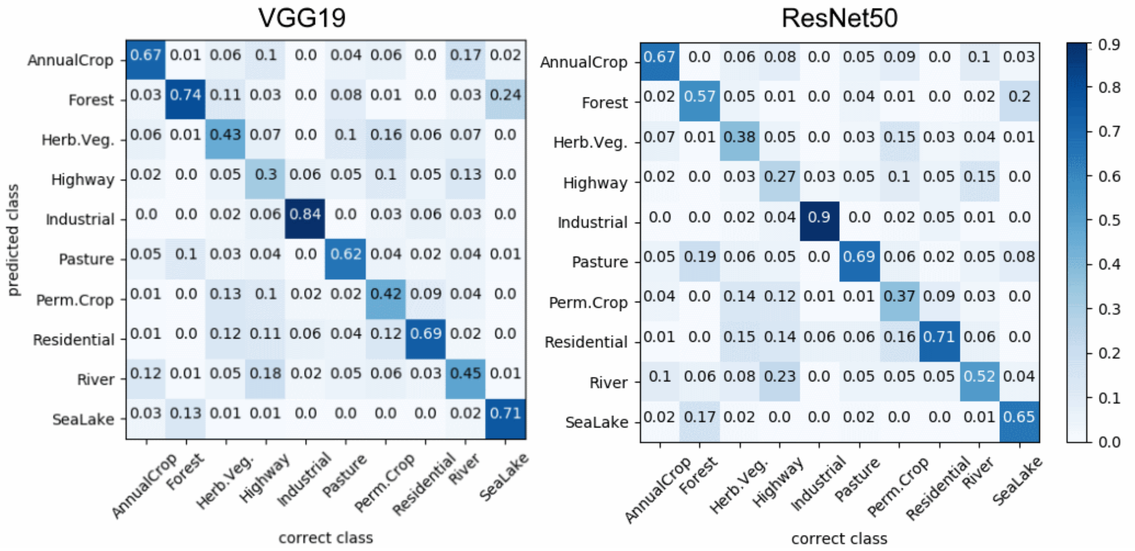
Fig. 5.6 shows example images of the translated EuroSAT dataset alongside their optical pendants. The visual quality of the images is diverse. In the left image it can be seen that edges are clearly observable but the image is definitely too rich in contrast. Small structures are not as distinguishable as expected which can be seen in the second image depicting a highway. Also the urban areas are more blurry than in fig. 5.5. For the river class two examples are shown to illustrate the quality discrepancies observable in the whole dataset. It seems as if the right river is too low in contrast and therefore translated like a forest which can be seen on the right side.

Although the quality of translation is not as good as expected, fig. 5.6 shows more distinguishable features than fig. 5.1.

To investigate if the visual more distinguishable images enhance the classification performance the experiments of section 5.1 are repeated with the newly created dataset.

They show that in terms of the accuracy the ESRGAN translation is worse than the Pix2Pix translation. Using the pre-trained ResNet50 classifier results in an accuracy of roughly 50%. Without pre-trained weights it accomplishes $(57.3 \pm 3.7)\%$ accuracy on the test set, which is slightly lower than the accuracy of $(59.0 \pm 0.8)\%$ achieved by the pre-trained VGG19 network.

The confusion matrices shown in fig. 5.7 differ from the previously presented ones. Highways and permanent crop are again difficult to identify for the networks, but



VGG19

predicted class \ correct class	AnnualCrop	Forest	Herb.Veg.	Highway	Industrial	Pasture	Perm.Crop	Residential	River	SeaLake
AnnualCrop	0.67	0.01	0.06	0.1	0.0	0.04	0.06	0.0	0.17	0.02
Forest	0.03	0.74	0.11	0.03	0.0	0.08	0.01	0.0	0.03	0.24
Herb.Veg.	0.06	0.01	0.43	0.07	0.0	0.1	0.16	0.06	0.07	0.0
Highway	0.02	0.0	0.05	0.3	0.06	0.05	0.1	0.05	0.13	0.0
Industrial	0.0	0.0	0.02	0.06	0.84	0.0	0.03	0.06	0.03	0.0
Pasture	0.05	0.1	0.03	0.04	0.0	0.62	0.04	0.02	0.04	0.01
Perm.Crop	0.01	0.0	0.13	0.1	0.02	0.02	0.42	0.09	0.04	0.0
Residential	0.01	0.0	0.12	0.11	0.06	0.04	0.12	0.69	0.02	0.0
River	0.12	0.01	0.05	0.18	0.02	0.05	0.06	0.03	0.45	0.01
SeaLake	0.03	0.13	0.01	0.01	0.0	0.0	0.0	0.0	0.02	0.71

ResNet50

predicted class \ correct class	AnnualCrop	Forest	Herb.Veg.	Highway	Industrial	Pasture	Perm.Crop	Residential	River	SeaLake
AnnualCrop	0.67	0.0	0.06	0.08	0.0	0.05	0.09	0.0	0.1	0.03
Forest	0.02	0.57	0.05	0.01	0.0	0.04	0.01	0.0	0.02	0.2
Herb.Veg.	0.07	0.01	0.38	0.05	0.0	0.03	0.15	0.03	0.04	0.01
Highway	0.02	0.0	0.03	0.27	0.03	0.05	0.1	0.05	0.15	0.0
Industrial	0.0	0.0	0.02	0.04	0.9	0.0	0.02	0.05	0.01	0.0
Pasture	0.05	0.19	0.06	0.05	0.0	0.69	0.06	0.02	0.05	0.08
Perm.Crop	0.04	0.0	0.14	0.12	0.01	0.01	0.37	0.09	0.03	0.0
Residential	0.01	0.0	0.15	0.14	0.06	0.06	0.16	0.71	0.06	0.0
River	0.1	0.06	0.08	0.23	0.0	0.05	0.05	0.05	0.52	0.04
SeaLake	0.02	0.17	0.02	0.0	0.0	0.02	0.0	0.0	0.01	0.65

Figure 5.7: Confusion matrices for the VGG19 (left) and ResNet50 (right) networks trained on the ESRGAN dataset.

this time water bodies that were formerly classified nearly perfectly cause problems. On the other hand urban areas, especially industrial areas are classified very well and also pasture is recognized above average while previously causing difficulties. Conspicuous is also that water bodies are mixed up with forest relatively often.

Regarding the differences in the confidence matrices it seems to be possible that the combination of the datasets translated by the Pix2Pix and the ESRGAN frameworks improves the overall performance. The corresponding experiment is performed applying the VGG19 network with and the ResNet50 without pre-training.
Training on the merged dataset results in an accuracy of (58.0 ± 1.3)% resp. (63.6 ± 1.0)% for the VGG19 resp. ResNet50. For the VGG19 network the accuracy is worse than on the single datasets but for the ResNet50 it improved the accuracy. As expected, the values of the confusion matrix qualitatively range between those of the confusion matrices of the Pix2Pix resp. ESRGAN datasets.

To finally evaluate the created classifiers, their performance on a different dataset is tested.

5.3 Testing the Classifiers on a different Dataset

Since there are only few available datasets for training deep neural networks in remote sensing, testing a deep learning algorithm not only with the regarded, but also different datasets is often omitted (e.g. [17][2]). This corresponds to the expectation that the data used for training and testing is of the same distribution as the data fed into the algorithm in the application phase. Especially in the area of remote sensing with different imaging devices and pre-processing methods on the one hand and a large variety of the data on the other hand, this can not be assumed to be true in many cases.
Therefore the previously trained classifiers are now applied to data of the Sen1-2 spring subset, which has neither be used to train the classifiers, nor to train the GAN translators. Besides being unfamiliar to the classifiers, the Sen1-2 spring subset was selected since no labeled datasets categorized into the same classes as the EuroSAT dataset are available. As the dataset has to be created manually an unused subset of the Sen1-2 dataset is selected due to its availability.

The dataset consists of 150 images, 15 for every class. Although this is a very small number, it is a compromise between the informative value provided by the dataset and the effort to create it. For the selection of samples the spring scenes up to number 27 are utilized. Images were selected such that for any class at most three images per scene are picked and that no overlapping images are chosen. The selection was based on the optical images as those can be interpreted more reliably. After selecting the optical image a subimage of size 64 × 64 depicting the specific class was extracted and added to the dataset. The same procedure was repeated for the SAR image using the exact same area as for the optical image. Thereby an optical and a SAR version of the dataset was created, the first being used for baseline testing again. In fig. 5.8 example images for every class are depicted.

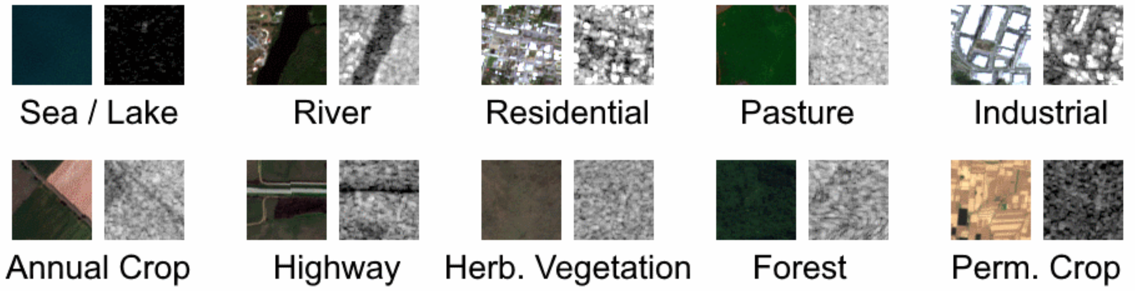


Figure 5.8: Example optical / SAR images from the manually created test set

Even though the images were selected carefully, errors, especially for the annual resp. permanent crop classes which are difficult to distinguish, can not be ruled out. Tab. 5.3 shows the results for the classifiers previously trained.

training dataset / modality	accuracy: ResNet50	accuracy: VGG19
optical dataset	(37.7 ± 0.9) %	(55.6 ± 0.9) %
SAR $G^{(20)}$, pre-trained classifiers	(24.5 ± 1.0) %	(34.5 ± 1.3) %
SAR $G^{(20)}$, no pre-training	(25.7 ± 2.6) %	-
SAR $G^{(20)}$, no pre-training, dropout	(28.9 ± 0.9) %	-
ESRGAN dataset	(21.7 ± 3.0) %	(24.9 ± 2.2) %
merged dataset	(29.6 ± 3.4) %	(33.3 ± 2.4) %

Table 5.3: Test accuracy for all classifiers on the manually created dataset

Apparently all classifiers achieve poor results. The classifiers trained on the optical dataset perform extremely bad compared to their initial testing results of roughly 90% accuracy, which is especially true for the ResNet50. Regarding the confidence matrices for both cases (fig. A1, appendix), fundamental difficulties with the classes pasture, river, permanent crop and annual crop are observable. While the confusion of annual and permanent crop can potentially be a consequence of the dataset creation, the other classes were selected with focus on the clarity of the main features. Moreover the confusion of the agricultural land only represents a small part of the misclassification for these two classes.

Besides the problematic cases the VGG19 network performs well on the handcrafted dataset whereas the ResNet50 seems to lack generalization capability. This is valid for the classifiers trained on SAR data too, even though the ResNet50 performance on the original test set was superior for every training modality.

The classifiers trained on the ESRGAN dataset achieved the worst accuracy which corresponds to the perceptual impression that characteristic features are more present in the $G^{(20)}$ dataset. Regarding the confusion matrices of the SAR classifiers (fig. A1, appendix) it becomes apparent that similar classes cause problems. Industrial and residential areas are recognized comparatively well which was also seen on the initial test set. However, rivers are one of the best classified classes as well as highways for the VGG19 classifiers in contrast to the former test set. Some classes like pasture, crops and herbaceous vegetation are completely missclassified. The forest and sea/lake classes are also classified poorly, although they always yielded above-average results on the former test set. Finally it is conspicuous that many classes are erroneously considered to be in the residential class.

Chapter 6

Conclusion

The objective of this thesis is the implementation and evaluation of a SAR classifier without the existence of a labeled dataset. To accomplish this, the similarity between the optical and SAR imaging modalities was exploited, utilizing the availability of both, a labeled optical and a co-registered optical / SAR dataset. The task was split into two steps, the implementation of an image translator with conditional GANs with its application to the optical dataset and the training of a DNN classifier on the artificial SAR dataset. While the previous parts of this thesis were devoted to background, data acquisition, implementation and description of the results, this chapter focuses on the evaluation of the approach.

For the first goal of setting up an algorithm capable of translating optical to SAR imagery, the Sen1-2 dataset was used to train two cGAN frameworks. Before these frameworks and their achievements are evaluated, the datasets suitability for this task shall be reviewed.
The main benefits of the Sen1-2 dataset are the free access, the huge number of samples, its variety and the increased similarity of the imaging geometries compared e.g. to the SARptical dataset. While deep learning algorithms usually scale with the dataset size, a high variety in the data increases the tasks complexity. Regarding the manually created subset utilized in the previous section illustrates this variety. Fig. 6.1 shows different optical images for the river class as an example.

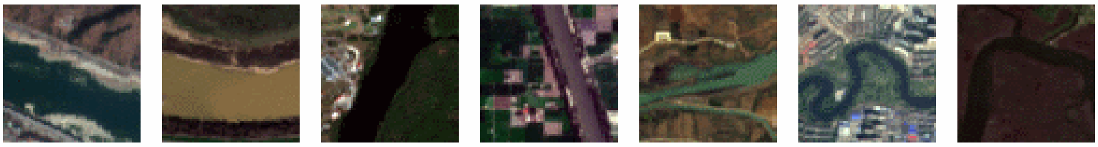


Figure 6.1: Example optical images from the manually created Sen1-2 subset

When thinking of a river observed with an air- or spaceborne device usually a lengthy blue water body surrounded by another land cover type comes into mind which might be of different sizes or oriented in different directions. Besides this spatial variety a spectral variety (i.e. color) and, more present also for the SAR images, a contrast variety between the river and its surroundings are very distinctive. With respect to the categorization of the EuroSAT dataset the examples thus show a high intra-class variety. This increases the difficulty of creating a sufficient translation algorithm. On the other hand the variety of the dataset potentially allows to build a very general translation algorithm.

Furthermore it was already mentioned that the consistency of the Sen1-2 and EuroSAT datasets is essentially. Sticking to the river example, having a very broad distribution in terms of spatial, spectral and contrast variety for the Sen1-2 dataset, but a narrow or very different distribution in the EuroSAT dataset, could cause problems even if the translator network works well on the dataset it was trained on. Utilizing both datasets inside the framework corresponds to the assumption that their distributions are similar, i.e. the already mentioned fundamental assumption of deep learning [33]. Obviously this is only an approximation and measuring the difference between the distributions is difficult. Collecting the images with the same device (Sentinel 2) is advantageous for the similarity. From the data gathering perspective the EuroSAT dataset includes only a subset of the Sen1-2 dataset as it was acquired using only scenes recorded over Europe, whereas the Sen1-2 dataset includes scenes from the Earth's complete landmass. Lastly the pre-processing of the data may differ. For the creation of the EuroSAT dataset no atmospheric correction was carried out for example. No information referring to this are provided by the description of the Sen1-2 dataset.

While the similarity of the distributions is an oversimplification, in practise EuroSAT and Sen1-2 are the most similar datasets available for the task and their suitability can only be evaluated in conjunction with the translation algorithms.

The applicability of the Pix2Pix framework to different tasks was showcased already in [13]. Baseline tests did show that the generator structure was adaptable to the demand of a smaller input size. Applying it to the Sen1-2 dataset caused problems in the beginning since the inherent speckle noise of the SAR images was not captured. Readjusting the weighting of the two objectives (L1 and adversarial, eq. 4.2) substantially increased both, the perceptual quality and the training behaviour. Still the experiments were carried out on a very small subset of the Sen1-2 dataset. To test the scaling of the algorithm, the size of the subset was doubled which further increased the variety. Although the training was less stable the perceptual quality did not change noticeably. Multiple examples showcased that the translation generally works, but sometimes appears to produce results worse than the original SAR images.

One issue of the generated images, especially for urban areas was their fine structure. It often appeared blurry to a certain degree i.e. the edges are less defined than in the original images. The ESRGAN framework was then introduced to improve the sharpness of the translated images. Again, the proposed weighting values of the objective function and the learning rate had to be changed in order to effectively apply the framework to the SAR translation task. Although small structures in the Sen1-2 test set were translated better than before (fig. 5.5) the drawback of the ESRGAN is its memory requirement which inhibits the usage of the full sizes 256×256 images for the training. Depending on the size of training set only 64×64 or 128×128 patches are usable for the given data processing system. The superiority of the translator trained on the bigger patches indicates that the usage of smaller images is disadvantageous in general which corresponds to the findings in [15].
Regarding the small structures, e.g. in urban areas the ESRGAN framework performs better than the Pix2Pix framework. For the overall quality of the translated images this can not be decided certainly, as some land-cover types like farmland and vegetation seem to be depicted more clear compared to the original images. The problem of subjectivity and a missing metric was already discussed in section 4.3. Summarizing the findings for both frameworks on the Sen1-2 dataset, their results on optical to SAR image translation is far from being perfect. Both perform well overall, some images basically resemble the original SAR images while in other cases obvious differences become apparent. The perceptual quality (i.e. if the observer can differentiate real and generated samples) is also good, with urban areas sometimes being the weakness of the Pix2Pix framework as they are distinguishable due to their blurriness.

Building the translator network was only a preliminary goal. The primal objective requires the translator to generate "reasonable" SAR images from the EuroSAT dataset that allow to successfully train a classification algorithm. For the translated images to be reasonable it would be desirable to correspond to the perceptual quality of the Sen1-2 test set as no better results can be expected. Evaluating this is hindered by the lack of SAR data to compare on the one hand and by the small image size of 64×64 pixels.

Three different artificial datasets exemplary depicted in fig. 5.1 and fig. 5.6 were created using two Pix2Pix generators trained on 10 resp. 20 scenes and the ESRGAN. Employing the visibility of important features in the images as indicator, the Pix2Pix translator trained with 20 scenes appeared to be superior to the two other translators. The ESRGAN translator often generated sharper edges but tends to generate images too rich in contrast. Additionally, it sometimes produced similar results for different classes e.g. river and forest in fig. 5.6. This also appeared for the other translators but fewer in the case of the Pix2Pix generators, especially the one trained on 20 scenes. Putting it together, the Pix2Pix translator trained on more data generated the images of the highest perceptual quality for the EuroSAT dataset.

Although it is not possible to quantify how much worse the translation works on the EuroSAT dataset compared to the Sen1-2 dataset, two types of failures appeared only for the EuroSAT dataset. Overlooking primary features is the first one and visible for the examples of highway and river in fig. 5.1 and the second river example in fig. 5.6. This is most probably caused by the previously discussed differences in the datasets. It appears more frequently when there is a low contrast or unusual colors for these features like a brown river or a highway between crop of different colors. The second failure case can be seen in the bottom left image in fig. 5.1. In the bottom right corner an artifact appears. It is not limited to the image edges and corresponds to artifacts described in [15]. According to their findings it is induced by the usage of batch normalization and in fact did not appear when applying the ESRGAN translator.

Utilizing the three artificial SAR datasets, two different classification networks, i.e. ResNet50 and VGG19, were trained and compared. Subsequent to the comparison of the achievements on the EuroSAT test set, a subset of unseen Sen1-2 images was manually selected and labeled to test the performance of these classifiers on different data. To compare the results the networks were also trained and tested on the corresponding optical data.

The 98.57% accuracy reported in [17] have not been replicated on the optical dataset, instead roughly 90% were achieved with the ResNet50 and VGG19 network. These results together with the corresponding confusion matrices were utilized as baseline tests for comparison with the results of the SAR classifiers. Tab. 6.1 lists all classifiers and their achieved accuracies for the EuroSAT test set.

training dataset / modality	accuracy: ResNet50	accuracy: VGG19
optical dataset	(90.8 ± 0.3) %	(89.2 ± 0.2) %
SAR $G^{(20)}$, pre-trained classifiers	(56.4 ± 0.3) %	(62.5 ± 0.5) %
SAR $G^{(20)}$, no pre-training	(66.6 ± 0.7) %	-
SAR $G^{(20)}$, no pre-training, dropout	(68.2 ± 1.1) %	-
ESRGAN dataset	(57.3 ± 3.7) %	(59.0 ± 0.8) %
merged dataset	(63.6 ± 1.0) %	(58.0 ± 1.3) %

Table 6.1: Test accuracy for all classifiers on the EuroSAT test set.

The best results for SAR data were achieved with the ResNet50 without pre-training on the dataset translated with the Pix2Pix framework. Besides omitting the pre-training, the results could not be improved. Neither the usage of the ESRGAN framework that enhanced the translation of small details and edges, nor the combination of the datasets improved the classification accuracy. As already discussed, specific classes caused problems. Highways and rivers were repeatedly misclassified, which resulted either from the specific features that are only present on a small part of the image or not even translated correctly. Also some vegetation types like permanent crop, herbaceous vegetation and pasture were difficult to classify for most of the networks. The classifiers trained on the ESRGAN dataset achieved better results for industrial area, which was expected due to the improved translation of small structures, and pasture but apart from that performed worse.

Although it appeared plausible, the fewer spectral information of the SAR data (one channel compared to three channel optical data) is not the explanation for the worse performance of the SAR classifiers. This was shown by training the same classifiers with single channel (i.e. gray-scale) optical images which resulted in only slightly worse accuracies (85% resp. 87%). As discussed, the remaining reasons for the inferior performance are either the differences in imaging modalities, i.e. SAR data can not be classified as well as optical data for this task, or the translation process. Based on the previous evaluation the open problems of the translation process appear more plausible.

While the results are not amazing, achieving nearly 70% accuracy in SAR classification without the usage of labeled data can be regarded as success.

The disadvantage of employing an artificially created dataset for training a classifier is that the assumption of similar distributions for the training data and the real data is even more vulnerable. Of course the test set contains samples that have not been utilized while training the classifier, but they where created in the same way as the training set. As already argued the dataset creation impacts the distribution of the data. While it was questionable if the optical Sen1-2 images and the EuroSAT images are of a similar distribution, it is more unlikely that the artificial SAR data and real SAR data are of resembling distributions.

Therefore a "real" SAR dataset was created utilizing the autumn subset of the Sen1-2 dataset and manually selecting and labeling the images. The informative value of a dataset consisting of only 150 samples is obviously limited but it allows to answer the question if the created classifiers are working at all.

Regarding the results in tab. 5.3 the answer seems to be no. With the accuracies of all classifiers lying between 20% and 35%, the distance to random guessing (i.e. 10% for ten different classes) is small. The inspection of the confusion matrices did again show a high variance in the accuracy for different classes. Here the well discriminated classes were urban areas on the one hand but also rivers and highways which is the exact opposite behaviour of the previous test results. This indicates that the more homogeneous classes were differently represented in the artificial datasets and the real SAR dataset which is in term a sign of the distributions not being similar. Interestingly the optical classifiers also performed considerably worse. As no source training a classifier on aerial imagery and testing it on a different dataset is known, the generality of this finding can not be confirmed. However it appears likely considering the perceptual similarity of the EuroSAT and Sen1-2 datasets and the popularity of the utilized classification algorithms.

Overall the VGG19 network generalizes better than the ResNet50 and can therefore be declared the better DNN for these specific task and datasets.

Summarizing the findings, the objective of the thesis was achieved partially. The proposed framework was successfully implemented and evaluated, showing its strengths and weaknesses and comparing different configurations i.e. translating and classification algorithms, different Sen1-2 subsets etc. Evaluating different training modalities on the contrary did not yield clear trends. For the translation algorithm the lack of a sufficient metric denied a quantitative analysis. Although the VGG19 network generalized better than the ResNet50, the influence of pre-training could not be clarified i.e. it could not be shown that pre-trained algorithms generalize better. Also the achieved accuracy is not satisfactory. Remaining below 70% accuracy on the test set does not allow the application of the algorithm even if the target distribution would exactly correspond to the training distribution. Moreover a semi-significant test on real data demonstrated that apart from a few classes the transferability of the classification algorithms, including the optical classifiers, is not given.

Although a well working classification algorithm could not be provided in the course of this thesis, the applicability of the framework was demonstrated.

As the two primary problems were the translation on different distributions and the classification of difficult classes, few simplifications could lead to an employable classifier. Firstly more elementary classes could be defined which allows to continue the usage of the EuroSAT dataset. Starting from four classes like water bodies, forest, urban area and agricultural land and expanding when a sufficient classification accuracy is achieved would be a possible guideline. Additionally the training and application of specialized translator algorithms e.g. for urban area, for vegetation and for agricultural land could be promising.

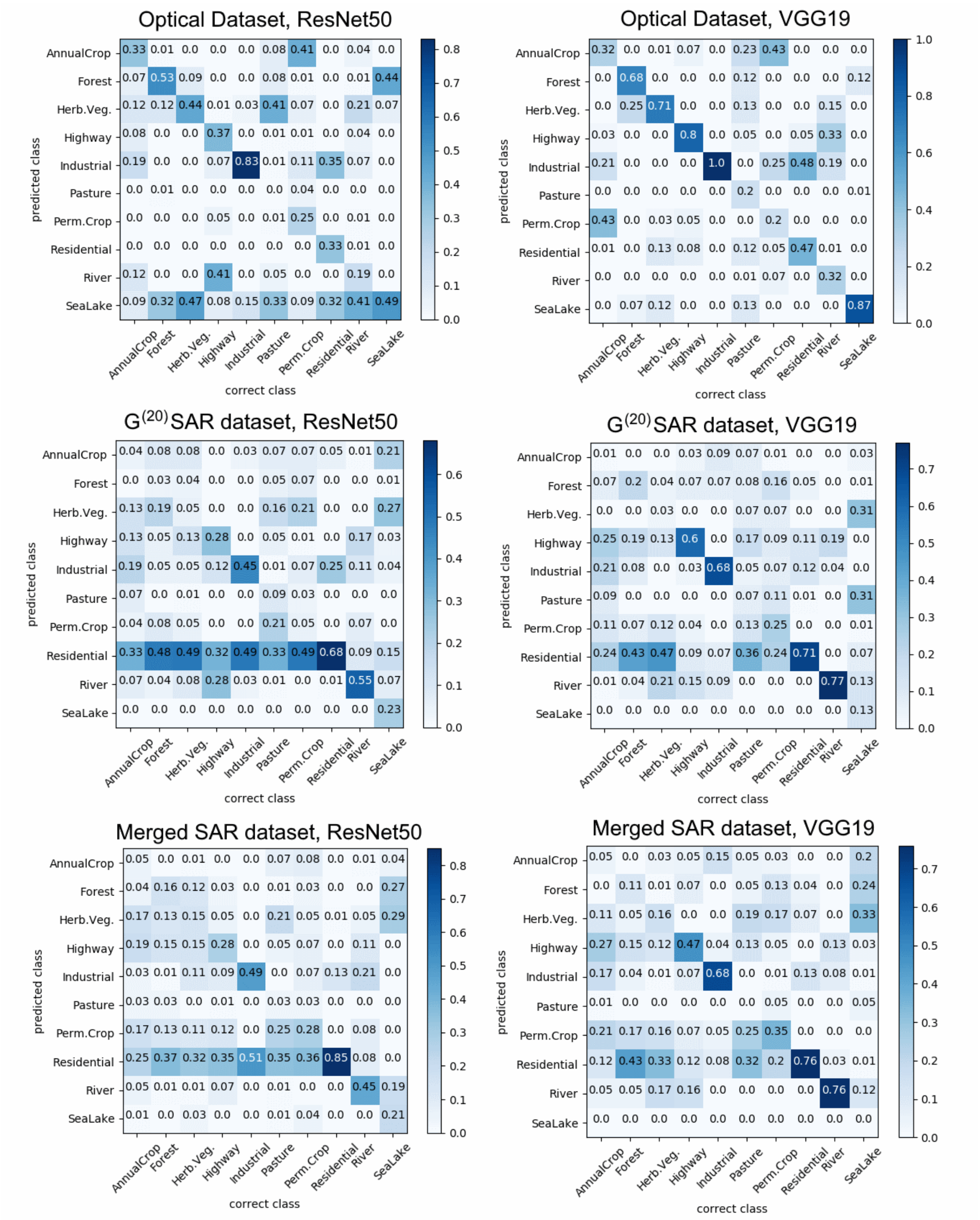
In the long term the different distributions of the datasets need to be unified. Since the Sen1-2 dataset is very large and could only be partially exploited here, a labeled subset which could be used instead of the EuroSAT dataset would probably resolve a substantial part of the problems described here.

Also more complex data has the potential to further improve the framework. The EuroSAT dataset is available with multispectral data but as the Sen1-2 dataset only incorporates the RGB channels this could not be utilized. Furthermore [16] states that single-polarization SAR data was used for simplicity which indicates that the extension of the Sen1-2 dataset to multi-polarization data is possible.

From a theoretical perspective an improved metric for the evaluation of image translation tasks is required. Not requiring a human observer for the evaluation of a translation network would most likely not only speed up the creation and testing of multiple configurations but also improve the results by reducing the subjectivity.

Finally the question of the frameworks usability apart from remote sensing is to be answered. [13] showcased the various existing translation tasks and existing co-registered datasets for these problems. Possible applications thus may arise from completely different fields. As the existence of an equal classification for both datasets is required to utilize the proposed framework, different representations or measurements of the same objects or scenes are the most probable field of application. Examples may be night vision where a lack of labeled datasets is likely and the translation of visual datasets like ImageNet might be promising. Deep learning also enables 2D to 3D object translation which could also allow the usage of existing labeled datasets in a domain with a deficit thereof. This study did however show that a high similarity of the datasets is required to successfully apply the framework.

Appendix



Optical Dataset, ResNet50

	AnnualCrop	Forest	Herb.Veg.	Highway	Industrial	Pasture	Perm.Crop	Residential	River	SeaLake
AnnualCrop	0.33	0.01	0.0	0.0	0.0	0.08	0.41	0.0	0.04	0.0
Forest	0.07	0.53	0.09	0.0	0.0	0.08	0.01	0.0	0.01	0.44
Herb.Veg.	0.12	0.12	0.44	0.01	0.03	0.41	0.07	0.0	0.21	0.07
Highway	0.08	0.0	0.0	0.37	0.0	0.01	0.01	0.0	0.04	0.0
Industrial	0.19	0.0	0.0	0.07	0.83	0.01	0.11	0.35	0.07	0.0
Pasture	0.0	0.01	0.0	0.0	0.0	0.0	0.04	0.0	0.0	0.0
Perm.Crop	0.0	0.0	0.0	0.05	0.0	0.01	0.25	0.0	0.01	0.0
Residential	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.33	0.01	0.0
River	0.12	0.0	0.0	0.41	0.0	0.05	0.0	0.0	0.19	0.0
SeaLake	0.09	0.32	0.47	0.08	0.15	0.33	0.09	0.32	0.41	0.49

Optical Dataset, VGG19

	AnnualCrop	Forest	Herb.Veg.	Highway	Industrial	Pasture	Perm.Crop	Residential	River	SeaLake
AnnualCrop	0.32	0.0	0.01	0.07	0.0	0.23	0.43	0.0	0.0	0.0
Forest	0.0	0.68	0.0	0.0	0.0	0.12	0.0	0.0	0.0	0.12
Herb.Veg.	0.0	0.25	0.71	0.0	0.0	0.13	0.0	0.0	0.15	0.0
Highway	0.03	0.0	0.0	0.8	0.0	0.05	0.0	0.05	0.33	0.0
Industrial	0.21	0.0	0.0	0.0	1.0	0.0	0.25	0.48	0.19	0.0
Pasture	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.01
Perm.Crop	0.43	0.0	0.03	0.05	0.0	0.0	0.2	0.0	0.0	0.0
Residential	0.01	0.0	0.13	0.08	0.0	0.12	0.05	0.47	0.01	0.0
River	0.0	0.0	0.0	0.0	0.0	0.01	0.07	0.0	0.32	0.0
SeaLake	0.0	0.07	0.12	0.0	0.0	0.13	0.0	0.0	0.0	0.87

$G^{(20)}$SAR dataset, ResNet50

	AnnualCrop	Forest	Herb.Veg.	Highway	Industrial	Pasture	Perm.Crop	Residential	River	SeaLake
AnnualCrop	0.04	0.08	0.08	0.0	0.03	0.07	0.07	0.05	0.01	0.21
Forest	0.0	0.03	0.04	0.0	0.0	0.05	0.07	0.0	0.0	0.01
Herb.Veg.	0.13	0.19	0.05	0.0	0.0	0.16	0.21	0.0	0.0	0.27
Highway	0.13	0.05	0.13	0.28	0.0	0.05	0.01	0.0	0.17	0.03
Industrial	0.19	0.05	0.05	0.12	0.45	0.01	0.07	0.25	0.11	0.04
Pasture	0.07	0.0	0.01	0.0	0.0	0.09	0.03	0.0	0.0	0.0
Perm.Crop	0.04	0.08	0.05	0.0	0.0	0.21	0.05	0.0	0.07	0.0
Residential	0.33	0.48	0.49	0.32	0.49	0.33	0.49	0.68	0.09	0.15
River	0.07	0.04	0.08	0.28	0.03	0.01	0.0	0.01	0.55	0.07
SeaLake	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.23

$G^{(20)}$SAR dataset, VGG19

	AnnualCrop	Forest	Herb.Veg.	Highway	Industrial	Pasture	Perm.Crop	Residential	River	SeaLake
AnnualCrop	0.01	0.0	0.0	0.03	0.09	0.07	0.01	0.0	0.0	0.03
Forest	0.07	0.2	0.04	0.07	0.07	0.08	0.16	0.05	0.0	0.01
Herb.Veg.	0.0	0.0	0.03	0.0	0.0	0.07	0.07	0.0	0.0	0.31
Highway	0.25	0.19	0.13	0.6	0.0	0.17	0.09	0.11	0.19	0.0
Industrial	0.21	0.08	0.0	0.03	0.68	0.05	0.07	0.12	0.04	0.0
Pasture	0.09	0.0	0.0	0.0	0.0	0.07	0.11	0.01	0.0	0.31
Perm.Crop	0.11	0.07	0.12	0.04	0.0	0.13	0.25	0.0	0.0	0.01
Residential	0.24	0.43	0.47	0.09	0.07	0.36	0.24	0.71	0.0	0.07
River	0.01	0.04	0.21	0.15	0.09	0.0	0.0	0.0	0.77	0.13
SeaLake	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.13

Merged SAR dataset, ResNet50

	AnnualCrop	Forest	Herb.Veg.	Highway	Industrial	Pasture	Perm.Crop	Residential	River	SeaLake
AnnualCrop	0.05	0.0	0.01	0.0	0.0	0.07	0.08	0.0	0.01	0.04
Forest	0.04	0.16	0.12	0.03	0.0	0.01	0.03	0.0	0.0	0.27
Herb.Veg.	0.17	0.13	0.15	0.05	0.0	0.21	0.05	0.01	0.05	0.29
Highway	0.19	0.15	0.15	0.28	0.0	0.05	0.07	0.0	0.11	0.0
Industrial	0.03	0.01	0.11	0.09	0.49	0.0	0.07	0.13	0.21	0.0
Pasture	0.03	0.03	0.0	0.01	0.0	0.03	0.03	0.0	0.0	0.0
Perm.Crop	0.17	0.13	0.11	0.12	0.0	0.25	0.28	0.0	0.08	0.0
Residential	0.25	0.37	0.32	0.35	0.51	0.35	0.36	0.85	0.08	0.0
River	0.05	0.01	0.01	0.07	0.0	0.01	0.0	0.0	0.45	0.19
SeaLake	0.01	0.0	0.03	0.0	0.0	0.01	0.04	0.0	0.0	0.21

Merged SAR dataset, VGG19

	AnnualCrop	Forest	Herb.Veg.	Highway	Industrial	Pasture	Perm.Crop	Residential	River	SeaLake
AnnualCrop	0.05	0.0	0.03	0.05	0.15	0.05	0.03	0.0	0.0	0.2
Forest	0.0	0.11	0.01	0.07	0.0	0.05	0.13	0.04	0.0	0.24
Herb.Veg.	0.11	0.05	0.16	0.0	0.0	0.19	0.17	0.07	0.0	0.33
Highway	0.27	0.15	0.12	0.47	0.04	0.13	0.05	0.0	0.13	0.03
Industrial	0.17	0.04	0.01	0.07	0.68	0.0	0.01	0.13	0.08	0.01
Pasture	0.01	0.0	0.0	0.0	0.0	0.0	0.05	0.0	0.0	0.05
Perm.Crop	0.21	0.17	0.16	0.07	0.05	0.25	0.35	0.0	0.0	0.0
Residential	0.12	0.43	0.33	0.12	0.08	0.32	0.2	0.76	0.03	0.01
River	0.05	0.05	0.17	0.16	0.0	0.0	0.0	0.0	0.76	0.12
SeaLake	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure A1: Confusion matrices for classification algorithms trained on different datasets and tested on the manually created dataset (section 5.3)

Bibliography

[1] Schwegmann, Colin P. ; Kleynhans, Waldo ; Salmon, BP: The development of deep learning in synthetic aperture radar imagery. In: *2017 International Workshop on Remote Sensing with Intelligent Processing (RSIP)* IEEE, 2017, S. 1–2

[2] Marmanis, Dimitrios ; Yao, Wei ; Adam, Fathalrahman ; Datcu, Mihai ; Reinartz, Peter ; Schindler, Konrad ; Wegner, Jan D. ; Stilla, Uwe: Artificial generation of big data for improving image classification: a generative adversarial network approach on SAR data. In: *arXiv preprint arXiv:1711.02010* (2017)

[3] Krizhevsky, Alex ; Sutskever, Ilya ; Hinton, Geoffrey E.: Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*, 2012, S. 1097–1105

[4] Zeiler, Matthew D. ; Fergus, Rob: Visualizing and understanding convolutional networks. In: *European conference on computer vision* Springer, 2014, S. 818–833

[5] Simonyan, Karen ; Zisserman, Andrew: Very deep convolutional networks for large-scale image recognition. In: *arXiv preprint arXiv:1409.1556* (2014)

[6] Szegedy, Christian ; Liu, Wei ; Jia, Yangqing ; Sermanet, Pierre ; Reed, Scott ; Anguelov, Dragomir ; Erhan, Dumitru ; Vanhoucke, Vincent ; Rabinovich, Andrew: Going deeper with convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, S. 1–9

[7] He, Kaiming ; Zhang, Xiangyu ; Ren, Shaoqing ; Sun, Jian: Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, S. 770–778

[8] Girshick, Ross ; Donahue, Jeff ; Darrell, Trevor ; Malik, Jitendra: Region-based convolutional networks for accurate object detection and segmen-

tation. In: *IEEE transactions on pattern analysis and machine intelligence* 38 (2016), Nr. 1, S. 142–158

[9] Redmon, Joseph ; Divvala, Santosh ; Girshick, Ross ; Farhadi, Ali: You only look once: Unified, real-time object detection. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, S. 779–788

[10] Long, Jonathan ; Shelhamer, Evan ; Darrell, Trevor: Fully convolutional networks for semantic segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, S. 3431–3440

[11] Schmitt, M ; Hughes, L H. ; Körner, M ; Zhu, X X.: Colorizing sentinel-1 sar images using a variational autoencoder conditioned on sentinel-2 imagery. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLII (2018), S. 1045 – 1051

[12] Goodfellow, Ian ; Pouget-Abadie, Jean ; Mirza, Mehdi ; Xu, Bing ; Warde-Farley, David ; Ozair, Sherjil ; Courville, Aaron ; Bengio, Yoshua: Generative adversarial nets. In: *Advances in neural information processing systems*, 2014, S. 2672–2680

[13] Isola, Phillip ; Zhu, Jun-Yan ; Zhou, Tinghui ; Efros, Alexei A.: Image-to-image translation with conditional adversarial networks. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* IEEE, 2017, S. 5967–5976

[14] Doersch, Carl: Tutorial on variational autoencoders. In: *arXiv preprint arXiv:1606.05908* (2016)

[15] Wang, Xintao ; Yu, Ke ; Wu, Shixiang ; Gu, Jinjin ; Liu, Yihao ; Dong, Chao ; Loy, Chen C. ; Qiao, Yu ; Tang, Xiaoou: ESRGAN: Enhanced super-resolution generative adversarial networks. In: *arXiv preprint arXiv:1809.00219* (2018)

[16] Schmitt, Michael ; Hughes, Lloyd H. ; Zhu, Xiao X.: The SEN1-2 Dataset for Deep Learning in SAR-Optical Data Fusion. In: *arXiv preprint arXiv:1807.01569* (2018)

[17] Helber, Patrick ; Bischke, Benjamin ; Dengel, Andreas ; Borth, Damian: Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. In: *arXiv preprint arXiv:1709.00029* (2017)

[18] Schmitt, M ; Hughes, L H. ; Körner, M ; Zhu, X X.: Colorizing sentinel-1 sar images using a variational autoencoder conditioned on sentinel-2 imagery. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLII (2018), S. 1045 – 1051

[19] Moreira, Alberto ; Prats-Iraola, Pau ; Younis, Marwan ; Krieger, Gerhard ; Hajnsek, Irena ; Papathanassiou, Konstantinos: A Tutorial on Synthetic Aperture Radar. In: *IEEE Geoscience and Remote Sensing Magazine* (2013), S. 1–43

[20] Skolnik, Merrill I.: *Radar handbook*. 2. McGraw-Hill, 1990. – ISBN 0–07–057913–X

[21] Borengasser, Marcus ; Hungate, William S. ; Watkins, Russell: *Hyperspectral Remote Sensing: Principles and Applications (Remote Sensing Applications Series)*. CRC Press, 2007. – ISBN 1566706548

[22] Liew, S C.: Optical Remote Sensing. In: *Principles of Remote Sensing* (2001). `https://crisp.nus.edu.sg/ research/tutorial/optical.htm`, Abruf: 2018-01-07

[23] Wang, Yuanyuan ; Zhu, Xiao X.: The SARptical Dataset for Joint Analysis of SAR and Optical Image in Dense Urban Area. In: *arXiv preprint arXiv:1801.07532* (2018)

[24] Cybenko, George: Approximation by superpositions of a sigmoidal function. In: *Mathematics of Control, Signals, and Systems (MCSS)* 2 (1989), Nr. 4, S. 303–314

[25] Rumelhart, David E. ; Hinton, Geoffrey E. ; Williams, Ronald J. u. a.: Learning representations by back-propagating errors. In: *Cognitive modeling* 5 (1988), Nr. 3, S. 1

[26] Nielsen, Michael: Neural Networks and Deep Learning. (2017). `http://neuralnetworksanddeeplearning.com/`, Abruf: 2018-01-05

[27] Rosenblatt, Frank: The perceptron: A probabilistic model for information storage and organization in the brain. In: *Psychological review* 65 (1958), Nr. 6, S. 386

[28] Tieleman, Tijmen ; Hinton, Geoffrey: Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. In: *COURSERA: Neural networks for machine learning* 4 (2012), Nr. 2, S. 26–31

[29] DUCHI, John ; HAZAN, Elad ; SINGER, Yoram: Adaptive subgradient methods for online learning and stochastic optimization. In: *Journal of Machine Learning Research* 12 (2011), Nr. Jul, S. 2121–2159

[30] KINGMA, Diederik ; BA, Jimmy: Adam: A method for stochastic optimization. In: *arXiv preprint arXiv:1412.6980* (2014)

[31] LECUN, Yann ; BOTTOU, Léon ; BENGIO, Yoshua ; HAFFNER, Patrick: Gradient-based learning applied to document recognition. In: *Proceedings of the IEEE* 86 (1998), Nr. 11, S. 2278–2324

[32] SRIVASTAVA, Nitish ; HINTON, Geoffrey ; KRIZHEVSKY, Alex ; SUTSKEVER, Ilya ; SALAKHUTDINOV, Ruslan: Dropout: a simple way to prevent neural networks from overfitting. In: *The Journal of Machine Learning Research* 15 (2014), Nr. 1, S. 1929–1958

[33] PAN, Sinno J. ; YANG, Qiang u. a.: A survey on transfer learning. In: *IEEE Transactions on knowledge and data engineering* 22 (2010), Nr. 10, S. 1345–1359

[34] LECUN, Yann A. ; BOTTOU, Léon ; ORR, Genevieve B. ; MÜLLER, Klaus-Robert: Efficient backprop. In: *Neural networks: Tricks of the trade.* Springer, 2012, S. 9–48

[35] IOFFE, Sergey ; SZEGEDY, Christian: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: *arXiv preprint arXiv:1502.03167* (2015)

[36] RADFORD, Alec ; METZ, Luke ; CHINTALA, Soumith: Unsupervised representation learning with deep convolutional generative adversarial networks. In: *arXiv preprint arXiv:1511.06434* (2015)

[37] SALIMANS, Tim ; GOODFELLOW, Ian ; ZAREMBA, Wojciech ; CHEUNG, Vicki ; RADFORD, Alec ; CHEN, Xi: Improved techniques for training gans. In: *Advances in Neural Information Processing Systems*, 2016, S. 2234–2242

[38] ARJOVSKY, Martin ; BOTTOU, Léon: Towards principled methods for training generative adversarial networks. In: *arXiv preprint arXiv:1701.04862* (2017)

[39] ARJOVSKY, Martin ; CHINTALA, Soumith ; BOTTOU, Léon: Wasserstein gan. In: *arXiv preprint arXiv:1701.07875* (2017)

[40] MIRZA, Mehdi ; OSINDERO, Simon: Conditional generative adversarial nets. In: *arXiv preprint arXiv:1411.1784* (2014)

[41] HINTON, Geoffrey E. ; SALAKHUTDINOV, Ruslan R.: Reducing the dimensionality of data with neural networks. In: *science* 313 (2006), Nr. 5786, S. 504–507

[42] RONNEBERGER, Olaf ; FISCHER, Philipp ; BROX, Thomas: U-net: Convolutional networks for biomedical image segmentation. In: *International Conference on Medical image computing and computer-assisted intervention* Springer, 2015, S. 234–241

[43] ABADI, Martín ; AGARWAL, Ashish ; BARHAM, Paul ; BREVDO, Eugene ; CHEN, Zhifeng ; CITRO, Craig ; CORRADO, Greg S. ; DAVIS, Andy ; DEAN, Jeffrey ; DEVIN, Matthieu ; GHEMAWAT, Sanjay ; GOODFELLOW, Ian ; HARP, Andrew ; IRVING, Geoffrey ; ISARD, Michael ; JIA, Yangqing ; JOZEFOWICZ, Rafal ; KAISER, Lukasz ; KUDLUR, Manjunath ; LEVENBERG, Josh ; MANÉ, Dandelion ; MONGA, Rajat ; MOORE, Sherry ; MURRAY, Derek ; OLAH, Chris ; SCHUSTER, Mike ; SHLENS, Jonathon ; STEINER, Benoit ; SUTSKEVER, Ilya ; TALWAR, Kunal ; TUCKER, Paul ; VANHOUCKE, Vincent ; VASUDEVAN, Vijay ; VIÉGAS, Fernanda ; VINYALS, Oriol ; WARDEN, Pete ; WATTENBERG, Martin ; WICKE, Martin ; YU, Yuan ; ZHENG, Xiaoqiang: *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems.* `https://www.tensorflow.org/`. Version: 2015. – Software available from tensorflow.org

[44] CHOLLET, François u. a.: *Keras.* `https://keras.io`, 2015

[45] LEDIG, Christian ; THEIS, Lucas ; HUSZÁR, Ferenc ; CABALLERO, Jose ; CUNNINGHAM, Andrew ; ACOSTA, Alejandro ; AITKEN, Andrew ; TEJANI, Alykhan ; TOTZ, Johannes ; WANG, Zehan u. a.: Photo-realistic single image super-resolution using a generative adversarial network. In: *arXiv preprint* (2017)

[46] JOLICOEUR-MARTINEAU, Alexia: The relativistic discriminator: a key element missing from standard GAN. In: *arXiv preprint arXiv:1807.00734* (2018)

List of Figures

1.1 Deep Learning and its classification in the area of Artificial Intelligence 2
1.2 The framework of the thesis (unsupervised training of a SAR classifier) 3

2.1 Geometry of a SAR device [19] 6
2.2 Required processing steps to visualize a SAR image [19] 8
2.3 Example of co-registered optical and SAR images from the SARptical dataset [23]; large differences in the image geometries become apparent 9
2.4 Examples of each class of the EuroSAT dataset [17] 11
2.5 Representation of the scenes utilized to create the Sen1-2 dataset [16] 12
2.6 Example patch pairs of the Sen1-2 dataset [16]; top: single-polarization SAR images, bottom: RGB optical images 12

3.1 The structure of a standard artificial neuron, depicting how the output is calculated, given the input. 14
3.2 A sketch of the layer-wise structure of a feed-forward neural network. It consists of input-, output- and hidden layers. A neuron in a specific layer is connected to all neurons of the neighbouring layers. 15
3.3 Visualization of a standard CNN structure, starting with convolutional and pooling layers and concluding with fully connected layers. . 16
3.4 The basic building block of a residual network as proposed by [7] . . . 18
3.5 The principle structure of a generative adversarial network 21
3.6 The (ideal) training scenario for a GAN (adapted from [12]) 22

4.1 Example applications of the Pix2Pix framework [13] 24
4.2 Principle structure of an autoencoder with skip connections (U-net) . 26
4.3 Example image pair of the Google maps dataset 28
4.4 Two examples of optical satellite images translated to maps from [13] 29
4.5 Four examples from the last epochs of the training. Each example consists of the condition (left), the generated image (middle) and the original image from the dataset (right). 29

4.6 Exemplary courses of the training; left: the input image; right: the original image from the dataset; middle: results at different stages of the training. 31
4.7 Four example results for the generator trained with a subset of the Sen1-2 dataset in run configuration (4). Each triplet consists of condition (left), generated image (middle) and original image (right). . . 33
4.8 Discriminator accuracy while training the GAN with $\lambda = 10$. 33
4.9 Four example results for the generator trained with a Sen 1-2 subset with λ changed to 10 from 100, i.e. the adversarial objective becoming more important. (condition: left, generated: middle, original: right) . 34
4.10 Results of increasing the amount of training data. Left: discriminator accuracy while training the GAN. Right: Example translation results (condition: left, generated: middle, original: right) 35
4.11 Test of smaller input size and comparison to the standard input size. From left to right: condition, 64×64 input (rearranged), 256×256 input, real image 37

5.1 Example images from the translated EuroSAT dataset; top: original images; middle: translation with generator trained on 10 Sen1-2 scenes; bottom: translation with generator trained on 20 scenes. . . . 39
5.2 Confusion matrices for both classifiers trained on the optical dataset. 43
5.3 Confusion matrices for both classifiers trained on the $G^{(20)}$ dataset. . 44
5.4 The ESRGAN generator structure; top: the basic architecture as proposed in [45]; bottom left: the basic residual block with and without batch normalization; bottom right: the residual in residual block as proposed in [15]. 48
5.5 Four example results of the ESRGAN generator trained on 10 Sen1-2 scenes as described above (condition: left, generated: middle, original: right) 49
5.6 Example images of the EuroSAT dataset translated with the ESRGAN generator; top: original images, bottom: translated images. . . 50
5.7 Confusion matrices for the VGG19 (left) and ResNet50 (right) networks trained on the ESRGAN dataset. 51
5.8 Example optical / SAR images from the manually created test set . . 52

6.1 Example optical images from the manually created Sen1-2 subset . . 54
A1 Confusion matrices for classification algorithms trained on different datasets and tested on the manually created dataset (section 5.3) . . 61

List of Tables

2.1 Commonly used bands and corresponding frequencies and wavelengths 8

5.1 Test accuracy for both classifiers trained on the optical EuroSAT dataset 42
5.2 Test accuracy for both classifiers trained on the translated datasets . 43
5.3 Test accuracy for all classifiers on the manually created dataset . . . 53

6.1 Test accuracy for all classifiers on the EuroSAT test set. 58

Selbstständigkeitserklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbstständig verfasst und noch nicht für andere Prüfungen eingereicht habe. Sämtliche Quellen einschließlich Internetquellen, die unverändert oder abgewandelt wiedergegeben werden, insbesondere Quellen für Texte, Grafiken, Tabellen und Bilder, sind als solche kenntlich gemacht. Mir ist bekannt, dass bei Verstößen gegen diese Grundsätze ein Verfahren wegen Täuschungsversuchs bzw. Täuschung eingeleitet wird.

Berlin,