

# Single Image Super-Resolution for SAR Images

**DIPLOMARBEIT**

zur Erlangung des akademischen Grades

**Diplom-Ingenieur**

im Rahmen des Studiums

**Data Science**

eingereicht von

**Philip Dimitrov, BSc**

Matrikelnummer 01025609

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Robert Sablatnig

Mitwirkung: Dipl.-Ing. Dr.techn. Sebastian Zambanini

Univ.Prof. Dipl.-Ing. Dr.techn. Wolfgang Wagner

Univ.Ass. Felix David Reuß, MSc

Wien, 7. Oktober 2021

---

Philip Dimitrov

---

Robert Sablatnig

---

Technische Universität Wien

A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ [www.tuwien.at](http://www.tuwien.at)



# Single Image Super-Resolution for SAR Images

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

**Diplom-Ingenieur**

in

**Data Science**

by

**Philip Dimitrov, BSc**

Registration Number 01025609

to the Faculty of Informatics

at the TU Wien

Advisor: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Robert Sablatnig

Assistance: Dipl.-Ing. Dr.techn. Sebastian Zambanini

Univ.Prof. Dipl.-Ing. Dr.techn. Wolfgang Wagner

Univ.Ass. Felix David Reuß, MSc

Vienna, 7<sup>th</sup> October, 2021

\_\_\_\_\_  
Philip Dimitrov

\_\_\_\_\_  
Robert Sablatnig





# Erklärung zur Verfassung der Arbeit

Philip Dimitrov, BSc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 7. Oktober 2021

---

Philip Dimitrov



# Danksagung

Hiermit will ich mich bei meinen Betreuern für ihre Unterstützung und die gute interdisziplinäre Zusammenarbeit bedanken. Großer Dank gilt auch meiner Freundin, meinen Freunden und meiner Familie.



# Acknowledgements

The computational results presented have been achieved, in part, using the Vienna Scientific Cluster (VSC).



# Kurzfassung

Die Einzelbild-Superauflösung (SA) ist eine Methode, um aus einem einzigen Bild mit niedriger Auflösung ein hochauflösendes Bild zu erzeugen. Die SA wird in verschiedenen Bereichen eingesetzt, z. B. in der medizinischen Bildgebung, der Satelliten- und der Sicherheitsbildgebung. Die Verwendung von superaufgelösten Bildern beschleunigt die Trainingskonvergenz und erhöht die Erkennungs- und Segmentierungsgenauigkeit im Vergleich zu niedrig aufgelösten Bildern. Neben der Erhöhung der Auflösung kann die SA auch zur Rauschunterdrückung eingesetzt werden. Es gibt physikalische Beschränkungen für die Bildqualität und -auflösung aufgrund der Bilderfassungshardware. Die SA hilft, diese Beschränkungen zu überwinden. Diese Arbeit konzentriert sich auf die Anwendung von SA für C-Band Synthetic Aperture Radar (SAR)-Bilder, die von den Sentinel-1-Satelliten der Copernicus-Mission der Europäischen Weltraumorganisation aufgenommen wurden. Erdbeobachtungsaufgaben wie die Klassifizierung der Bodenbedeckung, die Erkennung von Ölverschmutzungen, Bodenoberflächentemperatur und Bodenfeuchtigkeit hängen von der Qualität der Fernerkundungsbilder ab. Geringe Auflösung und Rauschen beeinträchtigen die zugrundeliegenden Modelle, daher ist eine hohe Auflösung für die Geowissenschaften von großer Bedeutung. In dieser Arbeit werden modernste SA-Ansätze auf der Grundlage von tiefen neuronalen Netzen untersucht. Eine Erdbeobachtungsaufgabe zur Segmentierung der Bodenbedeckung wird verwendet, um die Eignung der SA für SAR-Bilder zu bewerten. Die Ergebnisse der Hochskalierung von SAR-Bildern um einen Faktor von 2 oder 4 werden anhand von Bildqualitätsmetriken (PSNR, SSIM) und einem Erdbeobachtungssegmentierungsmodell bewertet. Darüber hinaus wird untersucht, ob die SA-Modelle mit ungesesehenen zeitlichen und räumlichen Konditionen zurechtkommen und ob ein adversarisches Training die Ergebnisse weiter verbessern kann. Die abschließende Bewertung zeigt, dass SA für C-Band SAR-Bilder bei einer Hochskalierung um den Faktor 2 geeignet ist und dass ungesehene zeitliche und räumliche Konditionen keine Probleme verursachen. Im Gegensatz dazu ist für eine SA um den Faktor 4 zur Bewältigung ungesehener zeitlicher und räumlicher Konditionen ein zusätzlicher Aufwand (erneutes Trainieren des Erdbeobachtungsmodells auf den SA-Bildern) erforderlich. Die Ergebnisse deuten darauf hin, dass das adversarische Training sowohl die Klassifizierung als auch die Bildqualitätsmetriken verbessern kann. Wenn nur die niedrig aufgelösten Bilder gesichert werden, kann die SA um den Faktor 2 oder 4 den erforderlichen Speicherplatz um den Faktor 4 bzw. 16 verringern. Diese Arbeit legt den Grundstein für zukünftige Forschung im Bereich der Einzelbild-SA für C-Band SAR-Bilder.





# Abstract

Single image Super-Resolution (SR) is a method to get a high-resolution image out of a single Low-Resolution (LR) image. SR is used in different domains, such as medical imaging, satellite imaging, and security imaging. Using SR compared to LR images speeds up training convergence and boosts recognition and segmentation accuracy. Apart from increasing the resolution of LR images, SR is able to denoise. Given the image acquisition hardware, there are physical restrictions on the image quality and resolution. SR helps overcome those limitations. This work focuses on the application of SR for Synthetic Aperture Radar (SAR) C-Band images captured by the Sentinel-1 satellites of the Copernicus Mission conducted by the European Space Agency. Earth Observation (EO) tasks, such as land cover estimation, detection of oil spills, land surface temperature, and soil moisture depend on the quality of the given remote sensing images. LR and noise impairs the underlying models, therefore SR is significant for earth science. This thesis investigates state-of-the-art SR approaches on SAR C-band images based on deep neural networks. An EO task for pixel-wise land cover segmentation is proposed in order to assess the suitability of SR for SAR images. Results of upscaling SAR images by a factor of 2 or 4 are evaluated based on image quality metrics (PSNR, SSIM) and an EO segmentation model. Furthermore, it is assessed if the SR methods can handle unseen temporal and spatial conditions and if adversarial training can further enhance the results. The final evaluation shows that SR for SAR C-band images is viable for upscaling by a factor of 2 and that unseen temporal and spatial conditions are manageable. In contrast, for SR by a factor of 4 to handle unseen temporal and spatial conditions, additional effort (re-training the EO model on the SR images) is required. Results indicate that adversarial training can improve both classification and image quality metrics. By keeping only the LR images, SR by a factor of 2 or 4 can reduce the necessary storage by a ratio of 4 or 16, respectively. This work lays the ground for future research in the field of single image SR for SAR C-band images.



# Contents

<b>Kurzfassung</b>	<b>xi</b>
<b>Abstract</b>	<b>xiii</b>
<b>Contents</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation and Problem Statement . . . . .	1
1.2 Aim of the Thesis . . . . .	3
1.3 Structure of the Thesis . . . . .	4
<b>2 Related Work</b>	<b>5</b>
2.1 Remote Sensing and Sensor Types . . . . .	5
2.2 Super Resolution . . . . .	8
2.3 Semantic Image Segmentation . . . . .	10
2.4 Summary . . . . .	12
<b>3 Foundations of Deep Learning Based Super Resolution and Semantic Segmentation</b>	<b>13</b>
3.1 Training . . . . .	13
3.2 Activation Functions . . . . .	15
3.3 Sub-Pixel Convolution Layer . . . . .	15
3.4 Residual-Learning . . . . .	16
3.5 Models . . . . .	19
3.6 Summary . . . . .	30
<b>4 Methodology</b>	<b>33</b>
4.1 Data . . . . .	33
4.2 Study Site . . . . .	34
4.3 Data Pre-processing and Splitting . . . . .	36
4.4 Data Augmentation . . . . .	38
4.5 VGG Loss . . . . .	41
4.6 Model & Training Configurations . . . . .	41
4.7 Metrics . . . . .	44
	xv

4.8 Summary . . . . .	46
<b>5 Experiments</b>	<b>49</b>
5.1 Experimental Design . . . . .	49
5.2 Implementation Environment . . . . .	51
5.3 Results . . . . .	52
5.4 Summary . . . . .	78
<b>6 Conclusion</b>	<b>79</b>
6.1 Summary . . . . .	79
6.2 Future Work . . . . .	80
<b>List of Figures</b>	<b>83</b>
<b>List of Tables</b>	<b>87</b>
<b>List of Acronyms</b>	<b>89</b>
<b>Appendix</b>	<b>93</b>
Figures & Tables . . . . .	93
<b>Bibliography</b>	<b>97</b>

# CHAPTER 1

## Introduction

Given an image, Super-Resolution (SR) is a technique to create a new image with a higher resolution [1]. This is achieved using a single image or multiple images [2]. This work will focus on using single images to achieve the increase in resolution. Such methods are termed Single-Image Super-Resolution (SISR).

### 1.1 Motivation and Problem Statement

This thesis focuses on Synthetic Aperture Radar (SAR) remote sensing imagery. Remote sensing is the acquisition of information about an area or object, typically from aircraft or satellites [3]. The SAR instrument uses radio waves to create the image and operates in different acquisition modes depending on the application requirement, e.g. vegetation classification, oceanography or archaeology [4]. An example for such images can be seen in Figure 1.1, which is captured by the Sentinel-1 satellite of the Copernicus Mission by the European Space Agency.

High-Resolution (HR) images have a high level of detail [5], allow a precise image interpretation, and enable new remote sensing applications [6]. Low-Resolution (LR) images have less visible details, which hinder remote sensing applications like road extraction and target identification [7]. However, available instruments make a trade-off between high spatial and high temporal resolution [8, 9]. The goal of SR is to generate a HR image from a given LR input [10]. A comparative illustration between LR and HR images of the same geographic region are exemplified in Figure 1.2. In this example, the LR has sixteen times less pixels than the HR counterpart. The task of SR is to use the available information of the LR image to recreate an image in HR resolution. Furthermore, SR is able to reduce the inherent noise in the LR images [11, 12].

Earth Observation (EO) tasks like land cover estimation, detection of oil spills, land surface temperature, and soil moisture depend on remote sensing images [13]. There

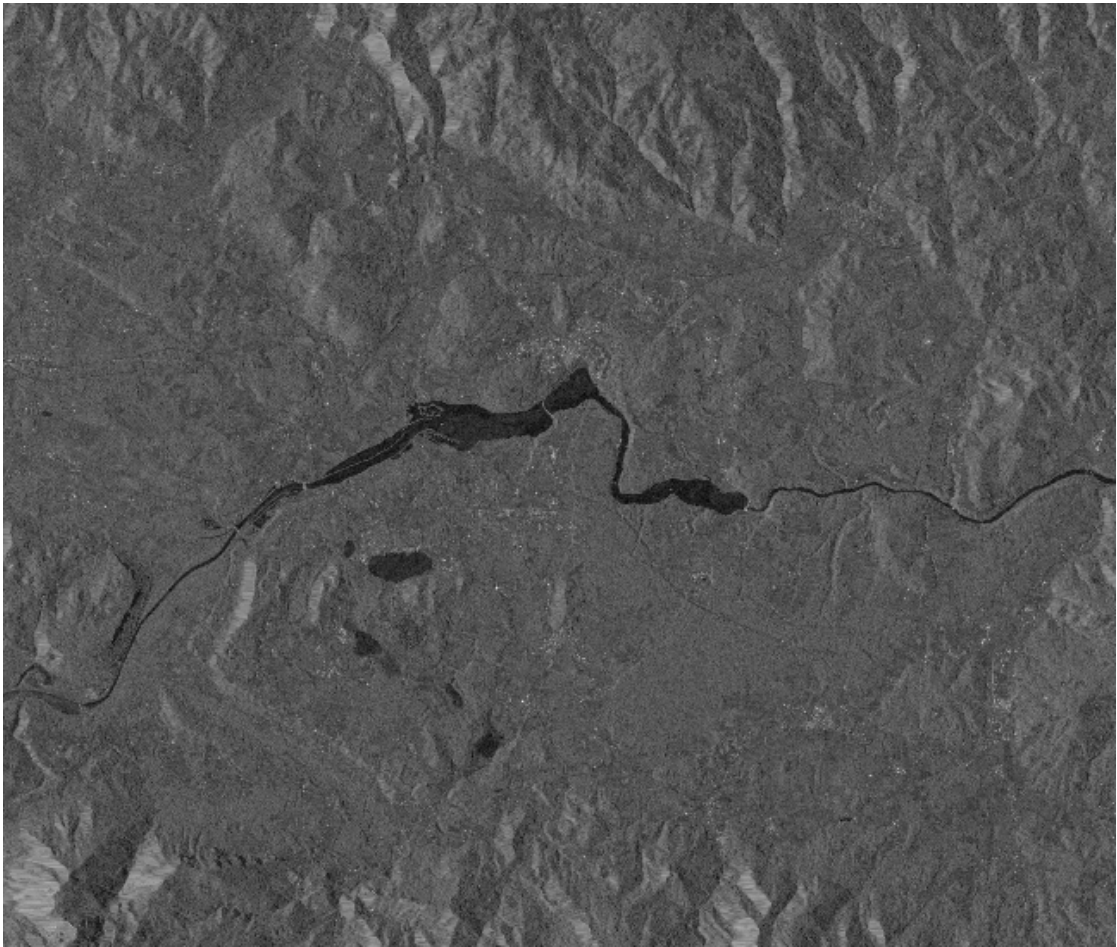


Figure 1.1: SAR image over Carinthia, Austria. Source: TU Wien Sentinel-1 datacube

is evidence for faster training convergence [14] and increased recognition [15] [16] and segmentation [17] accuracy, when using SR compared to the original LR images. Applying SR to increase the resolution and reduce the noise can consequently enhance the underlying models and is therefore significant for earth science.

Another motivation of SR, besides enhancing the quality of the given image, is the ability to cut down processing and storage capacity [18] through compression [19]. By using SR models, only LR images - compressed data - need to be stored, which can then be used to reconstruct the original HR image. This task is especially relevant to TU Wien as part of the Earth Observation Data Center (EODC) collaboration, as the data volume is expected to reach Peta-byte scale during a satellite lifespan [20]. This is relevant for conserving the data.

There are physical constraints on the image acquisition technology and hence on the image resolution [21]. Therefore, achieving SR reduces the limitations of the sensors

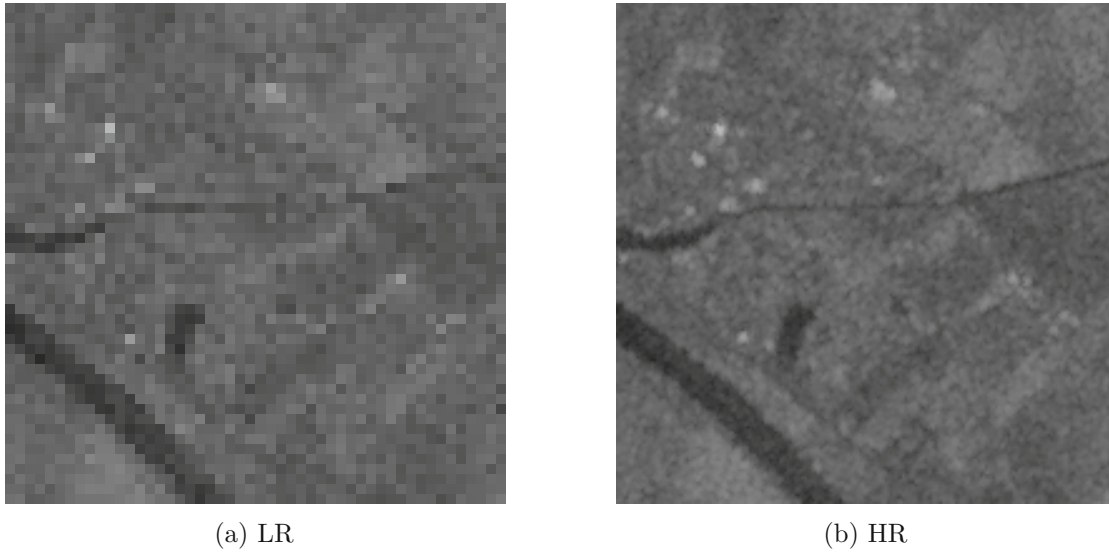


Figure 1.2: Same geographic area in (a) LR (40m GSD) and (b) HR (10m GSD).

currently available or deployed [22]. It can enhance the historical data available, and make it more comparable to the new (and better) remote sensing images.

## 1.2 Aim of the Thesis

This thesis aims to answer the following question: *Is a Convolutional Neural Network (CNN) model suitable for increasing the resolution of a given SAR C-band image by a factor of 2 or 4?* In detail, the suitability of the CNN model will be evaluated on the basis of semantic image segmentation for land cover, i.e. assigning a land cover class to each image pixel. The EO task assessment is described in Section 5. Furthermore, the goal of this work is to provide a model for upscaling lower resolution SAR images based on State-Of-The-Art (SOTA) techniques for SR.

While answering the main research question, the following additional sub-questions will be addressed:

1. How does the CNN handle unseen temporal conditions, e.g. can the model handle autumn conditions without being trained on such?
2. How does the CNN handle unseen spatial conditions, e.g. can the model handle mountainous regions without being trained on such?
3. Can a Generative Adversarial Network (GAN) [23] improve the results of the CNN?

This work refers to SISR with scale factors of 2 and 4 to up-scale SAR images with 20 meter to 10 meter, and 40 meter to 10 meter spatial resolution, respectively. The images with 10 meters pixel spacing are termed *high resolution*, the rest *low resolution*.

The goal is to assess if SR is suitable for SAR C-band images on the grounds of a task-based evaluation. Analyzing state-of-the-art approaches with and without deep learning is crucial for SAR SR. To serve as a basis for future research, the models will be also evaluated with the standard SR metrics PSNR and SSIM [24], as this offers more comparability than the task-specific assessment.

### 1.3 Structure of the Thesis

The remainder of this thesis is structured as follows: Chapter 2 presents an overview of existing literature in the field. Additionally, it formally defines remote sensing, SR, and semantic image segmentation.

Chapter 3 presents the data at hand and introduces key concepts of neural networks for the SR tasks. Neural Network (NN) components for SR are preluded and discussed. Furthermore, the state-of-the-art models which are going to be used in the experiments are presented, i.e. SRCNN, VDSR, SRResNet, SRGAN, ESRGAN, and EESRGAN. Chapter 3 concludes with introducing U-Nets as used for the task-based evaluation.

Chapter 4 introduces the datasets and the required data pre-processing steps. The importance and usage of data augmentation is showcased. Training and implementation details are presented.

Chapter 5 depicts how the experiments are set up to answer the research question. Important classification and image quality metrics are defined. The results of the experiments are presented based on the models introduced in Chapter 4. Next to presenting the results, this chapter focuses on discussing and interpreting the results. The research question and sub-questions are evaluated.

Section 6 concludes with what can be learned from this work and what the future prospects for SR techniques in deep learning for spatial data are.



# Related Work

Section 2.1 gives an introduction to the topic of remote sensing. For a better understanding of the data at hand, the underlying sensor types are specified and discussed.

An overview of the SOTA SR methods is given in Section 2.2. Furthermore, Section 2.2 formally defines the task of SR.

Section 2.3 focuses on semantic segmentation, the goal of which is to label each pixel with a corresponding class. Semantic segmentation is a task that is used in this thesis to evaluate the different SR techniques.

## 2.1 Remote Sensing and Sensor Types

Remote sensing has no universally accepted definition [25]. A general definition is given by Sabins and Floyd [26]: "*Remote sensing is the science of acquiring, processing, and interpreting images and related data, acquired from aircraft and satellites, that record the interaction between matter and electromagnetic energy*".

Lush [27] divides remote sensing into two categories, depending on the electromagnetic spectrum of the recorded radiation. Optical sensors are able to record the reflected energy of the visible and infrared bands (wavelength less than 1 mm). Microwave sensors measure the microwave portion of the electromagnetic spectrum (wavelength larger than 1 mm). Furthermore, Joshi et al. [28] divide microwave remote sensing into passive and active radars. Passive radars measure the emitted energy of the observed surface. Whereas, active radars emit electromagnetic waves and measure the reflected signal.

In its basic form, the radar radiates electromagnetic energy and detects the reflected echoes [29]. The sensor is used to generate electromagnetic power, which is directed over the switch to the antenna. The antenna and switch are simultaneously used to collect the returned radio waves and to direct the pulse to the receiver. The receiver converts the power into digital numbers, which are stored by the data recorder for later processing [4].

## 2. RELATED WORK

For a radar to achieve high resolution, an antenna of sufficient size is necessary [30]. However, SAR is able to achieve high spatial resolution with smaller antennas by being in motion and by combining the received signals in a sequential way [30, 31]. Therefore, SAR imagery is one of the main sources for change detection studies [32].

SAR is a sensor capable of measuring the reflectivity of a surface [33]. Each pixel of the SAR data at hand contains the value of the backscatter coefficient,  $\sigma^0$ , expressed in Decibel (dB). Simply put,  $\sigma^0$  indicates what proportion of the initially transmitted power is returned, normalized by the illuminated area [34].

Radars are capable of transmitting and receiving electromagnetic waves in horizontal and vertical polarizations [35]. Figure 2.1 illustrates a pulse transmission with horizontal and vertical polarization. The arrow depicts the direction of the wave.

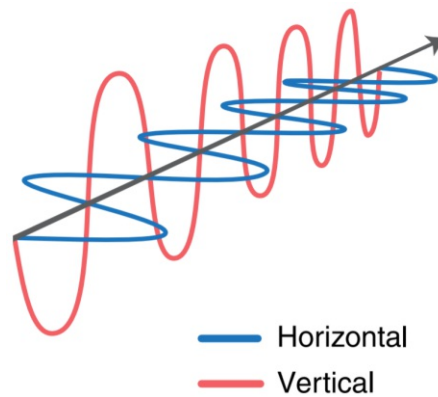


Figure 2.1: Electromagnetic wave with horizontal and vertical polarization. Source: [35]

There are four types of polarization depending on the polarization received and transmitted - Vertical on transmit, Vertical on receive (VV), Vertical on transmit, Horizontal on receive (VH), Horizontal on transmit, Horizontal on receive (HH), and Horizontal on transmit, Vertical on receive (HV) [36]. VV and HH are categorized as co-polarized, whereas VH and HV as cross-polarized [37]. The selection of polarization depends on the application. For instance, when monitoring vegetation, co-polarization is a better measure for early vegetation, whereas cross-polarization is better suited for advanced vegetative stages [38].

The intensity of the backscatter data depends on multiple factors. On the one hand, it is dependent on the object, i.e. surface (smooth, rough) [39], moisture conditions [40], soil conditions [41], topography (flat area, relief area) [42], soil texture (sand, silt, clay) [43]. On the other hand, it is dependent on the sensor, i.e. frequency (C-, X-, L-band) and polarization (VH, VV, HH, HV), while being independent of daylight and cloud cover [44].

Radars operate on different frequencies. The wavelength of the microwave is inversely proportional on the frequency, and vice versa, obeying the equation  $\text{Wavelength} =$

Speed of Light/Frequency [45]. Backscatter is dependent on the wavelength [46], hence the emitted microwave frequency is of importance. The frequencies are bundled in different bands. For convenience, each band is designated with specific letters, e.g. L (1-2 GHz), C (4-8 GHz), X (8-12 GHz) [29].

SAR images depicted in this work are based on the intensity values of the backscatter coefficient - high backscatter values are plotted in white, low in black. A general guideline to better understand the images is that more backscatter bounces lead to less backscattering response [47]. An example of single, double, and triple bounce can be seen in Figure 2.2. Furthermore, trees with larger leaves produce stronger single- and double bounce scattering than ones with smaller leaves, e.g. croplands [48]. Consequently, urban areas and man-made objects have high backscatter [49]. Compared to urban areas, forests have medium backscatter [50], yet distinguishably more than crop fields [51]. Water bodies have the lowest backscatter, due to the smooth surface. However, backscatter increases when the surface is rougher (wind, currents, waves) [52], which can be explained by the Bragg resonance as a dominant mechanism of radar backscatter from water surface [53].

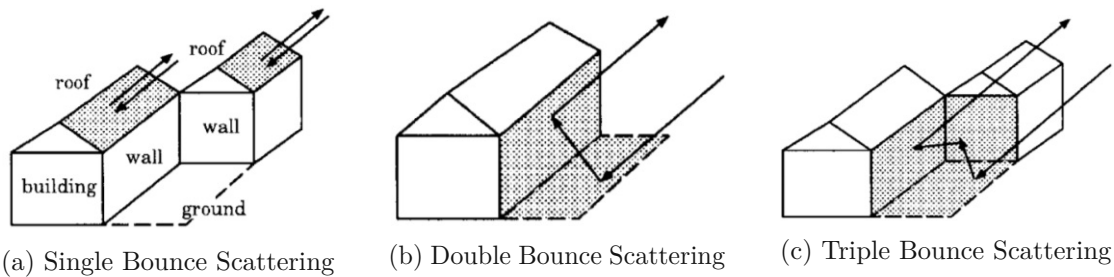


Figure 2.2: Image showcasing the backscattering mechanisms in urban areas: (a) Single Bounce Scattering, (b) Double Bounce Scattering, and (c) Triple Bounce Scattering. Source: [47]

Different applications have been developed which make use of the backscatter properties of different surfaces. Some examples include the detection of windthrow [44], monitoring of vegetation dynamics [54], crop field classification and monitoring [55], flood dynamics [56], ice classification and mapping [57], and climate studies [58].

Optical imagery is obstructed by clouds, haze, rain, and fog, which is a fundamental problem as it affects the availability of observing the surface underneath [59], particularly for tropical countries [4]. This leads to spatial and temporal data gaps [60]. Similarly, optical sensors reveal only the top of the canopy. Hence, the lower canopy and soils are obscured and cannot be depicted [28]. Whereas optical photogrammetry does not operate during the night, SAR imagery and monitoring is possible at nighttime since it has its own source of illumination (by emitting electromagnetic waves) [61].

Likewise, radar imagery has its drawbacks. The main drawback is the speckle, which is inherent in all SAR images. It is a signal-dependent granular noise (modeled as

multiplicative noise) produced by constructive and destructive interference of the signal [62] that causes the "salt-and-pepper" effect in the images. Speckle leads to measurement uncertainty [28], edge detection issues [63], difficulty in image interpretation, target identification, and classification [64]. An illustration of the speckle effect is shown in Figure 2.3, where an image has been synthetically altered to be noisy. Furthermore, topography (varying terrain elevation and slope) significantly impacts the geometric and radiometric properties of SAR images [65, 66].

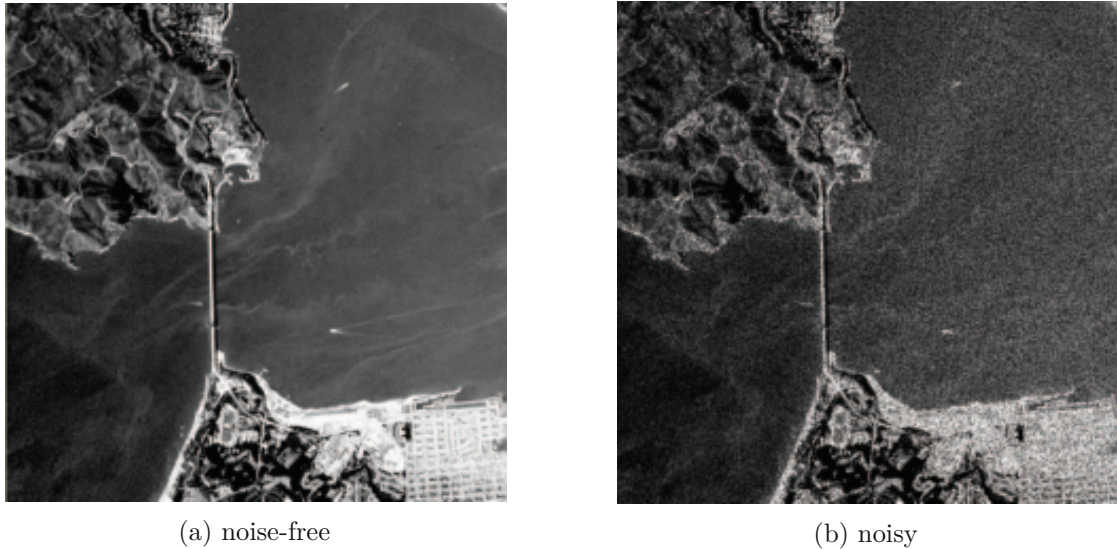


Figure 2.3: Synthetically speckled image from [62]: (a) noise-free reference, (b) noisy.

Spatial resolution, in remote sensing, defines the ability of the sensor to distinguish the reflected signal between two adjacent targets [67]. In higher (or finer) resolution images, smaller objects can be detected than in lower (or coarser) resolution images. In remote sensing, Ground Sampling Distance (GSD) is the distance between neighboring pixels measured on the ground. In the literature, the term *pixel spacing* and GSD are used interchangeably. The higher the GSD value of a remote sensing image, the lower the spatial resolution of the image. Equivalently, having two images covering the same area, the one with more pixels is to be referred as high resolution, the one with less as low resolution.

Remote sensing is an important data source for gathering land-use and land-cover information [68]. By monitoring the earth's surface, better environmental understanding and decision making are possible [69].

## 2.2 Super Resolution

The goal of SR is to accurately estimate an HR image from a given LR image [18]. SR reconstruction is an ill-posed inverse problem [70]. It is an inverse problem, as the

objective is to find a function  $f$  which takes as input a LR image and outputs a SR image. An ill-posed problem is a problem which has an unstable solution, more than one solution, or has no solution. It is also known as an improperly posed problem [71]. The solution is stable, if a slight change to the initial data yields a small change of the solution [72]. SR is ill-posed, since the reconstruction of the HR image is extremely sensitive to the LR data [73]. Likewise, SR is ill-posed, since infinitely many compatible HR images can be downsampled to the same LR image [74, 75].

A method for image upscaling is interpolation [76]. Types of interpolation are nearest neighbor, bilinear, or bicubic interpolation [77]. Not only are those methods used to define a performance baseline, they are also used to create the input for the NN. Instead of using the lower resolution image, the interpolated image is often served to the first layer of the network [78, 79].

With the prevalence of Deep Learning (DL) in computer vision tasks, Dong et al.'s work [78] was one of the first to tackle SR using a convolutional neural network with 3 layers. This neural network is called Super-Resolution Convolutional Neural Network (SRCNN).

Kim et al. enhanced the SRCNN by adding more convolutional layers [79]. Deeper (and wider) network structures increase the model's performance [80]. However, this comes at the cost of overfitting, especially for smaller data sets, and increased training time. It should be noted that stacking too many layers leads to a *degradation problem* - accuracy starts to decrease [81].

To overcome the problems of increased learning time and degradation of accuracy, He et al. proposed a residual learning framework [82]. The framework utilizes shortcut connections [83] to relay the initial input image to the final layer. This technique is used by both Kim et al. [79] and Wagner et al. [84]. While Huang et al. incorporated the very deep convolutional neural network on remote sensing data<sup>1</sup> without success [85], Wagner et al. managed to get promising results.

Recently, GANs have been playing a main role in the field of SR [86]. This is due to the fact that adversarial training works decently even with small datasets<sup>2</sup> (which might be the case when working with spatial data) [87], and that it achieves perceptually realistic results [88]. As a consequence, it is a main component in some of the leading models (SRGAN [89], EEGAN [90], ESRGAN [91], and EESRGAN [92]).

A GAN is a framework for training two models simultaneously: a generator and a discriminator [23]. Following the analogy of Creswell et al. [93] in computer vision, the generator is an art forger and the discriminator is an art expert. The forger tries to create forgeries as close as possible to the original art. Meanwhile, the expert tries to distinguish between a given forgery and the authentic art. The success of one is the failure of the other.

<sup>1</sup>Based on the Sentinel-2 MultiSpectral Instrument, which is a very different product in comparison to the SAR from Sentinel-1, i.e. multi-spectral vs. single band

<sup>2</sup>As low as 400 training images.

The basis for this master thesis are the EEGAN [90], ESRGAN [91], and EESRGAN [92] state-of-the-art models. Their novelty lies in the usage of additional sub-networks for edge-enhancement, so-called *residual dense blocks*, and *residual-in-residual dense blocks*, based on the works of Jiang et al. [90], Zhang et al. [94], and Wang et al. [91], respectively.

Analyzing the ESRGAN makes sense because the authors have won the PRISM2018-SR Challenge in perception [91]. The EEGAN is chosen as it implements an edge-enhancement strategy for obtaining clear edges [90], a feature which is desired in satellite imaging. The EESRGAN model [92] is selected since the authors achieved state-of-the-art results on satellite images for object detection by utilizing SR in comparison to using the original data, EEGAN, or ESRGAN super-resolved images.

Most recently, models based on recurrent neural networks [95] or the novel trainable second-order channel attention [96] have started challenging the dominance of GAN-based SISR models. This master thesis will not focus on those techniques, as they are not yet verified for remote sensing images.

Most of the milestone-defining-papers are based on optical images. In comparison, only scarce literature covers DL-based SISR models evaluated on SAR data [14, 97, 98]. Furthermore, there is a shortage on benchmark datasets, hence it is difficult to derive conclusions for the remote sensing field [21] and especially for SAR imagery.

It is important to note that while this thesis is focusing only on DL approaches, there are non-DL methods such as the works of Kanakaraj et al. [64], and Karimi and Taban [99]. No papers were found covering the novel state-of-the-art SISR DL models on exclusively SAR C-band data. Kanakaraj et al. [100] have reported a SR framework for Sentinel-1 SAR C-band images - however, multiple (12) LR images are required for a single SR image, and it is not a DL approach. Nevertheless, in an adjacent topic, neural networks for fully Polarimetric Synthetic Aperture Radar (PolSAR) images are researched [101, 102, 103].

### 2.3 Semantic Image Segmentation

In computer vision, semantic segmentation is the task of partitioning a given image into multiple segments by assigning a label (class) to each pixel [104]. The output of the semantic segmentation is referred to as *semantic map* [105]. In remote sensing, the terms segmentation and classification are used interchangeably [106, 107, 108, 109]. On the contrary, in computer vision, image classification is referred to as assigning a class to the whole image [110].

Semantic segmentation is crucial to this work, as it will be used to evaluate the selected SR models. An example of semantic segmentation in remote sensing can be seen in Figure 2.4. The image contains five different classes (building, car, tree, low vegetation and impervious surface), each depicted in a different color.





Figure 2.4: Semantic image segmentation with five classes. Source: [111]

Pixel-level semantic segmentation is chosen as the task for the assessment of the SR models, since there are several earth observation use-cases. For instance land use and land cover classification [112], object detection (vehicle [113], ship [114], cloud [115] detection), and change detection [116]. Furthermore, semantic maps can be used as input for object detection models, as used by Audebert et al. [117] for detecting and classifying vehicles. Precise land cover classification indicates the position of borders, which is crucial for infrastructure management and urban planing [118].

Non-DL machine learning approaches focus on extracting low-level hand-crafted features such as color, hue, saturation, gradient, geometric context, shape, and texture [119]. Subsequently, a classifier is applied to predict a class for each pixel [111]. On the other hand, in DL, to extract high-level features, Long et al. [120] employ a Full Convolutional Network (FCN) exceeding SOTA methods for semantic segmentation. The FCN consists of convolutions, pooling [121], and activation functions. The semantic map is upsampled only at the end of the network through strided convolutions [122] to match the shape of the input image.

SegNet is a FCN using an encoder-decoder architecture for semantic pixel-wise segmentation [123]. The encoder part is based on the convolutional layers of the 16 layer VGG network [124]. The decoder part is used for upsampling the feature maps of each encoder layer, so that the final segmentation map matches the shape of the input image.

U-Net [125] is similar to the SegNet. The difference in the encoding part is that it uses the feature maps after the activation function instead of the feature maps after the pooling layer. This change enables the U-Net to compensate for the loss of information

due to the pooling layers [126]

Attention U-Net [127] utilizes attention gates to further improve the U-Net. U-Nets are well-established SOTA methods based on end-to-end deep CNN architectures and more efficient than patch-based models [128]. Therefore, the Attention U-Net is selected for the semantic image segmentation part of this thesis.

### 2.4 Summary

In this chapter formal definitions of remote sensing, SR, and semantic image segmentation have been presented. The varying remote sensors, with focus on optical and radar, together with their strengths and weaknesses have been discussed. Essential remote sensing terms such as speckle, pixel spacing, and ground sampling distance have been specified. Furthermore, SAR imagery, its influencing factors (surface, moisture conditions, soil conditions and texture, topography, frequency, polarization), and interpretation have been illustrated.

This chapter also introduced related work on SISR. Main differences between the referenced models lie in the number of layers, and the usage of residual learning or GAN frameworks. The mentioned models will be further described in Chapter 3.

Interpolation shows its usefulness as means of baseline and input for DL approaches. Additionally, some issues in training SR networks, such as increased training time, overfitting, and the degradation problem, have been depicted. It was shown that there are DL-based SISR models for SAR data. However, no state-of-the-art DL approaches have been validated on the data at hand (SAR C-band).

An overview to semantic image segmentation and its application in EO was given. Earlier semantic image segmentation models were based on feature engineering to extract meaningful visual features. In contrast, more recent methods use deep learning to extract high-level features. The role of pixel-wise segmentation in the thesis has been indicated as a measure to assess the quality of different SR images.



# Foundations of Deep Learning Based Super Resolution and Semantic Segmentation

This chapter gives a general introduction to DL in SR and semantic segmentation. The notation throughout this work is introduced.

Section 3.1 showcases how SR training is conducted, while defining the key loss functions. Section 3.2 introduces the activation functions in the context of neural networks. Section 3.2 covers sub-pixel convolutions, which are used to generate high-resolution representation. Section 3.4 outlines the residual learning framework, which is used to train deeper neural networks. Furthermore, Section 3.4 introduces the concepts of residual blocks, dense blocks, residual dense blocks, and residual-in-residual dense blocks. Those blocks are necessary for the models used throughout this thesis.

The models used to evaluate the research question are described in Section 3.5. Their architectures are depicted and discussed. Section 3.5 additionally introduces the adversarial training framework and displays how it can be utilized in the context of SR.

## 3.1 Training

$I^{LR}$  denotes the low-resolution,  $I^{HR}$  the high-resolution, and  $I^{SR}$  the super-resolved image.  $I^{LR}$ ,  $I^{HR}$ , and  $I^{SR}$  are different versions of the same image. The goal is to find a model  $G$ , which yields  $I^{SR}$  close to  $I^{HR}$ , denoted  $G(I^{LR}) = I^{SR}$ . In this work, in the context of neural networks,  $G$  stands for the generator network.

$I^{LR}$  is represented as a real-valued tensor of size  $H \times W \times C$ , where  $H$  is the height,  $W$  the width, and  $C$  the number of channels of the given image. Whereas  $I^{HR}$  and  $I^{SR}$  are

### 3. FOUNDATIONS OF DEEP LEARNING BASED SUPER RESOLUTION AND SEMANTIC SEGMENTATION

in the shape of  $rH \times rW \times C$ , where  $r$  denotes the upscaling factor, i.e. 2 or 4.  $C = 1$ , as only one channel is used.

The data at hand is sampled only in one resolution, particularly the best and only available resolution is with GSD of 10 meters. Thus, in training, the initial remote sensing image is downsampled and used as the low-resolution image, while having the original image serving as the high-resolution image.

Manhattan ( $L_1$ ) and Euclidean ( $L_2$ ) norms are key elements in training a SR network. The Mean Squared Error (MSE) is equivalent to the squared  $L_2$  norm and is defined as follows:

$$MSE(I^{HR}, I^{LR}) = (G(I^{LR}) - I^{HR})^2 \quad (3.1)$$

In SR one deals with pixel predictions, therefore  $G(I^{LR})$  and  $I^{HR}$  from (3.1) need to be represented as pixels. In particular, the MSE loss becomes:

$$MSE(I^{HR}, I^{LR}) = \frac{1}{r^2 HW} \sum_{w=1}^{rW} \sum_{h=1}^{rH} (G(I^{LR})_{h,w} - I_{h,w}^{HR})^2 \quad (3.2)$$

Where  $G(I^{LR})_{h,w}$  is the pixel at position  $(h, w)$  of the model's prediction.

The  $L_1$  criterion measures the Mean Absolute Error (MAE) and is defined as follows:

$$L_1(I^{HR}, I^{LR}) = |G(I^{LR}) - I^{HR}| \quad (3.3)$$

Another loss used in this work is the texture loss  $L_{texture}$ . It is proposed by Gatys et al. [129] to improve texture representations in CNNs. The loss is defined as follows for a set of layers  $L$ :

$$L_{texture}(I^{HR}, I^{LR}) = \sum_{l \in L} MSE(G^l(I^{LR}), G^l(I^{HR})) \quad (3.4)$$

Where  $G^l \in \mathbb{R}^{n \times n}$  is the Gram matrix. Hence, the texture loss is the Euclidean distance between the Gram matrices of the LR and HR images. The Gram matrix  $G^l$  has matrix entries  $G_{i,j}^l$ , such as:

$$G^l = \begin{pmatrix} G_{1,1}^l & \cdots & G_{1,n}^l \\ \vdots & \ddots & \vdots \\ G_{n,1}^l & \cdots & G_{n,n}^l \end{pmatrix} = \begin{pmatrix} F_1^l \cdot F_1^l & \cdots & F_1^l \cdot F_n^l \\ \vdots & \ddots & \vdots \\ F_n^l \cdot F_1^l & \cdots & F_n^l \cdot F_n^l \end{pmatrix} \quad (3.5)$$

Where the  $(i, j)$  entry of the  $l^{th}$  layer is defined as the dot product  $(\cdot)$  between the feature maps  $i$  and  $j$  of layer  $l$ , in equation:

$$G_{i,j}^l = F_i^l \cdot F_j^l = \sum_k F_{ik}^l F_{jk}^l \quad (3.6)$$

$F^l \in \mathbb{R}^{n \times m}$  is the feature map matrix of layer  $l$  in the network, which have  $n$  feature maps of length  $m$ .  $F_{i,k}^l$  is the activation of the  $i^{th}$  filter at position  $k$  in layer  $l$ . A feature map is the output (after activation) of the convolutional layer [130].

## 3.2 Activation Functions

An activation function defines the output of a neuron (unit of a NN) for a given input [131]. The SR models used in this work make use of different activation functions, namely sigmoid, Rectified Linear Units (ReLU), Leaky ReLU (LReLU), and Parametric ReLU (PReLU). For better understanding of the models used, the activation functions are defined and their differences are emphasized.

The sigmoid function produces an output between 0 and 1, and is defined as  $f(x) = 1/(1 + e^{-x})$  [132].  $f$  is the activation function and  $x$  is the input of the neuron.

ReLU is a function which produces an output between 0 and  $x$ , and is defined as  $f(x) = \max(0, x)$ . There is evidence that ReLU improve the training of deep NNs [133].

LReLU is defined as a piecewise function [134]:

$$f(x_i) = \begin{cases} x_i, & x_i > 0 \\ \alpha_i x_i, & x_i \leq 0 \end{cases} \quad (3.7)$$

Here,  $x_i$  denotes the input of the  $i$ -th channel. Hence, it is possible to have a different activation weight  $\alpha$  for different channels. In the original paper [134],  $\alpha_i$  is fixed at 100, however, this can be seen as a hyperparameter of the NN.

The authors of the LReLU claim that their activation function is more robust during the NN optimization, i.e. avoiding zero gradients. However, they also state that there are no significant performance improvements compared to the ReLU.

He et al. created the PReLU while surpassing human-level performance on the ImageNet 2012 classification dataset [135]. Neither ReLU nor leaky ReLU have any parameters, PReLU is a learned activation unit. In their work, He et al. state that it improves the accuracy, while the new parameter is negligible in comparison to the total number of model parameters. PReLU is defined as Equation (3.7), where  $\alpha_i$  is a learnable parameter [135].

## 3.3 Sub-Pixel Convolution Layer

Shi et al. [136] define the sub-pixel convolution layer as an operation which rearranges the elements of a tensor of size  $H \times W \times Cr^2$  to a tensor of size  $rH \times rW \times C$ . The new

size is exactly the desired shape of  $I^{SR}$ , since it increases the height and width of the original image by the factor  $r$ . Figure 3.1 visualizes the sub-pixel convolution layer. It aggregates the feature maps of the previous layer (low-resolution space) and creates the SR image. Figure 3.1 demonstrates the upscaling by factor 2, i.e.  $r = 2$ .

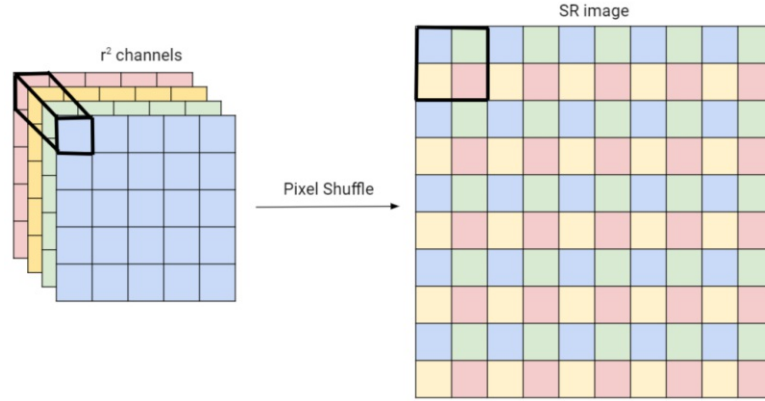


Figure 3.1: Sub-pixel convolution layer. Source: Own image based on [136]

The authors of the sub-pixel convolution layer state that there are two benefits of using a sub-pixel convolution layer, instead of working with an upsampled LR image from the beginning. The first benefit is that the feature extraction happens in the LR space. Hence, it is possible to use smaller filter sizes to cover the same receptive field (in comparison to working with the upsampled LR image). This considerably lowers the computational and memory complexity [136]. The second benefit is that more upscaling filters are learned in contrast to the single upscaling filter when using the bicubically upsampled image. This is due to the fact that there are multiple layers between the input and the sub-pixel convolution layer. Thus, the network is able to learn a more complicated mapping from low-resolution to high-resolution, opposed to the single fixed filter upscaling [136].

## 3.4 Residual-Learning

In the work of He et al. [82], a residual learning framework is proposed to ease up the training of very deep networks. For the time being they have successfully trained the deepest network on the ImageNet [137] visual image recognition challenge.

### 3.4.1 Residual Block

The main concept of the residual learning framework are so-called Residual Blocks (RB). A neural network can have one or more residual blocks. A representation of this concept can be seen in Figure 3.2. The idea is to add the output of a previous layer to a deeper layer.

In the case of SR,  $X$  in Figure 3.2 could represent the initial lower-resolution image, or an output of a previous neural network layer. This input goes through one or more layers

$G(X)$ . In the simplest case, where the first NN layer is  $X$  and the final is  $G(X) + X$ , the residual block is the full architecture of the SR network. In the papers considered,  $G(X)$  typically consists of convolutional layers [138] followed by non-linear functions (i.e. activation functions).

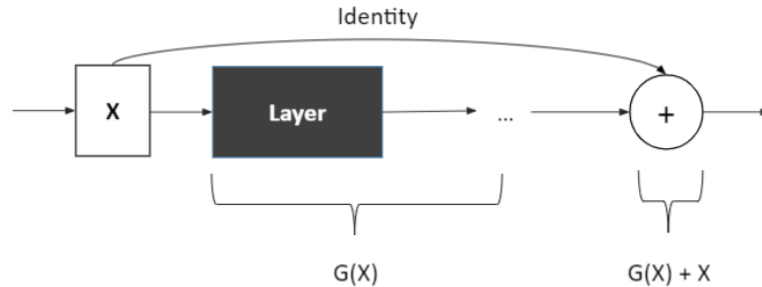


Figure 3.2: A residual block. Source: Own illustration based on [82]

The addition of the identity in this case is called a *shortcut connection*, due to the nature of skipping layers. It takes the input of a previous layer and adds it to the output of a layer ahead. This construct, as seen in Figure 3.2, does not add any extra model parameter nor computational complexity. This makes comparing models with same depth and width more accessible [82].

In their paper [82], He et al. argue - when assuming the identity mappings are optimal - that it would be easier to optimize a residual in comparison to a different mapping. This is due to the fact that most networks use non-linear functions and setting the residual to zero is easier. This statement is especially interesting in the case of SR, as most of the pixels in the input image already contain a lot of correct information.

SR models with residual-learning learn a residual image (instead of the SR image). To generate the SR image, the learned residual is added to the LR image. For this to work, the shapes of both residual and LR need to be equal. Due to the fact that the high-resolution version of a low-resolution image has similar features and structure, residual learning is intuitively a good mechanism.

### 3.4.2 Dense Block

The Dense Block (DB) was proposed by Huang et al. [139] as part of their Dense Convolutional Network (DenseNet), which obtained significant improvements over the SOTA in *all* Canadian Institute For Advanced Research (CIFAR) object recognition datasets [140]. The authors of the DB showed that DenseNets are able to scale to hundreds of layers without training difficulties.

Figure 3.3 shows the structure of the DB. The main idea is that each layer receives the feature maps of all previous layers of the given dense block by using direct connections. This connectivity between the layers is named *dense*. Each direct connection symbolizes a concatenation of the corresponding feature-maps. The concatenation increases the

number of features, which boosts the variation in the input of the consecutive layers and leads to an higher efficiency [139]. The concatenation operation is illustrated in Equation 3.8, where the tensors  $\mathbf{t}_1$  and  $\mathbf{t}_2$  are concatenated to  $\mathbf{t}_3$ .

$$\mathbf{t}_1 = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \mathbf{t}_2 = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}, \mathbf{t}_3 = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \\ 7 & 8 \end{bmatrix} \quad (3.8)$$

The dense block in Figure 3.3 is referred to as 3-layer. It should be noted that the DB does not necessarily need to be a composite function of Conv and ReLU operations, it can contain an arbitrary composition of layers.

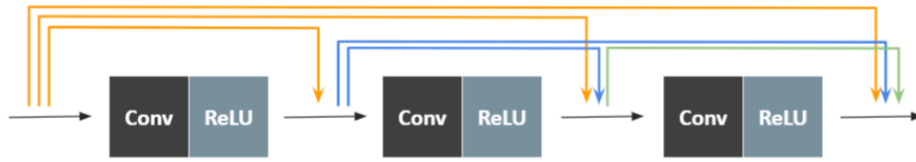


Figure 3.3: Dense Block. Source: Own image based on [139]

Tong et al. [141] first introduced the DB for image SR. The authors state that DBs are useful for learning high-level features.

Zhang et al. [142] argue that the SR models using only residual or dense blocks are missing the hierarchical features of the original low-resolution images. The next incremental improvement to the block structure to overcome this problem was the residual dense block.

#### 3.4.3 Residual Dense Block

An advantage of residual learning is that the network preserves the low-frequency components (color, gradient orientation) [143] of the input image [144]. High-level features (texture, edges, structures) are on the contrary extracted by deep and complex architectures [142]. Nonetheless, for convincing SR results a combination of both low- and high-level features is necessary [145].

To address the shortcomings of the RB and DB, the so-called Residual Dense Block (RDB) was proposed by Zhang et al. [94]. Not only is it able to extract the high-frequency features, but it also uses the hierarchical features of the original LR image.

The residual dense block, inspired by the RB and DB, can be observed in Figure 3.4. It consists of two main parts. The first one is a mechanism to pass each preceding RDB to each layer of the current RDB. This is done by the dense connectivity in the DB. The DB takes as input the previous RDB (RDB  $d - 1$  in 3.4), which enables the extraction of local dense features [94]. The second one is the summation of the RDB input and output,

this is in essence the residual learning as seen Figure 3.2. In their works, Zhang et al. conclude that this stabilizes the training of wider networks, preserves information from the current to the preceding RDBs, and improves the flow of information [94].



Figure 3.4: Residual Dense Block. Source: Own image based on [94]

#### 3.4.4 Residual-in-Residual Dense Block

Wang et al. [91] introduce the Residual-in-Residual Dense Block (RRDB) while creating their ESRGAN model, which earns them the first place in the Prim2018-SR Challenge (region 3) [146]. The RRDB is depicted in Figure 3.5. It is an extension of the residual dense block as it adds the input of the *first* RDB to the output of the *last* RDB. Essentially, a RRDB consists of multiple RDBs in combination with a shortcut connection. The authors claim the substituting DBs, RBs or RDBs for RRDBs increases the capacity of the networks while being easier to train.

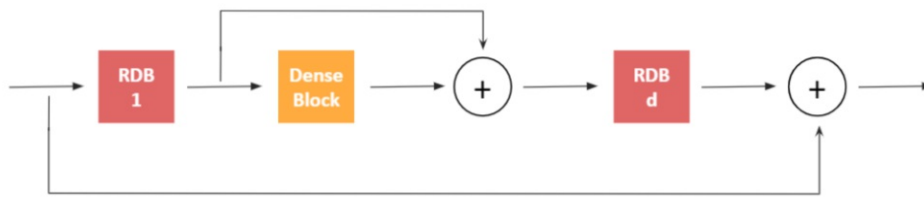


Figure 3.5: Residual-in-Residual Dense Block. Source: Own image based on [91]

### 3.5 Models

This section introduces and compares the models used in the thesis. Hyperparameters are presented in Section 4.6.

#### 3.5.1 Convolutional Neural Networks (CNNs)

The VDSR by Kim et al. [79], the SRCNN by Dong et al. [78], and the implementation of the VDSR for satellite imagery by Wagner et al. [84] use ReLU as their non-linear mapping (activation layers). The referred CNNs take as input the bicubically upsampled LR image (to the size of HR) and return the SR image.

The structure and the differences of the CNN approaches are showcased in the following passage.

#### SRCNN

An overview of the SRCNN network is displayed in Figure 3.6, where  $R$  symbolizes a ReLU layer. Dong et al. [78] use three layers with filters of sizes  $9 \times 9$ ,  $1 \times 1$ , and  $5 \times 5$ , respectively. Only after the first and second convolutional layer a ReLU follows.

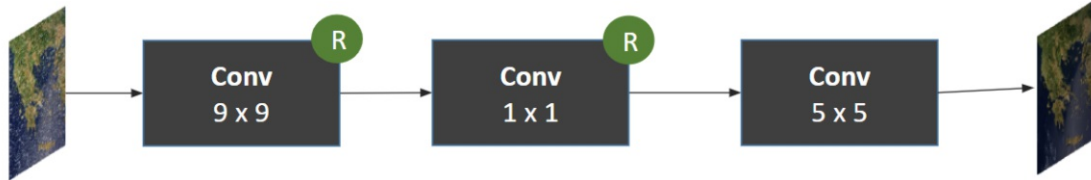


Figure 3.6: SRCNN network structure. Source: Own illustration based on [78]

#### VDSR

While Dong et al. [78] did not achieve any notable success using more layers, Kim et al. [79] managed to outperform the formal SRCNN model by using 20 layers, having filter sizes of  $3 \times 3$ . They named their network Very Deep Super-Resolution (VDSR). The second major change of the network structure is that a residual is learned instead of a high-resolution image. It can be argued that learning the residual is what enabled the training of the VDSR, since residual connection addresses the problem of vanishing and exploding gradients [81]. The structure of the VDSR network is shown in Figure 3.7.

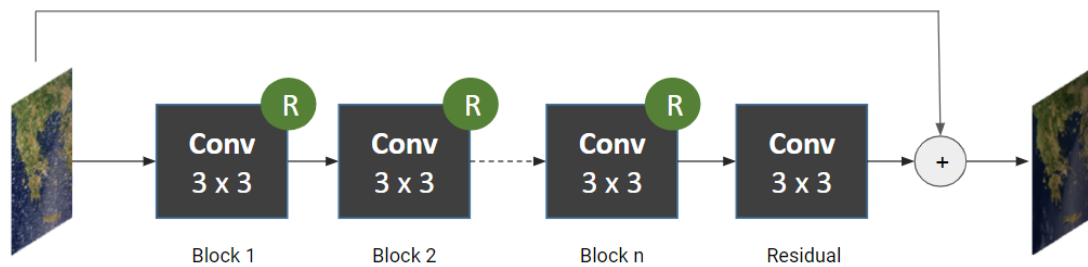


Figure 3.7: VDSR network structure. Source: Own illustration based on [79]

Kim et al. [79] achieve such a deep network by using padding after each feature map. In contrast, Dong et al. [78] do not use any padding. A  $36 \times 36$  px image and a layer with a  $9 \times 9$  filter is assumed as input. Without padding, the output of the layer would be a  $28 \times 28$  px image. With padding the result would have remained the same size as the input image, thus  $36 \times 36$  px, only padded with zeros. On the one hand, the number of layers gets limited by having filter sizes larger than  $1 \times 1$  (as  $1 \times 1$  does not reduce the output image). On the other hand, there is the border effect problem [147], in which pixels at the border are never centered in the filter.

Stacking more layers enables the VDSR network to have a larger receptive field:  $41 \times 41$  in comparison to the  $13 \times 13$  px of the SRCNN. Although having more model parameters,



the authors of the VDSR were able to achieve state-of-the-art results in 4 hours, whereas the SRCNN takes several days to converge. [79]. The VDSR is able to converge faster due to the residual-learning, high learning rate ( $10^4$  times higher than SRCNN), and gradient clipping. To draw a comparison in terms of convergence speed, Figure 3.8 contains the performance curve for residual and non-residual networks. It can be seen that residual models converge faster than non-residual models, and that they are more stable when using higher learning rates.

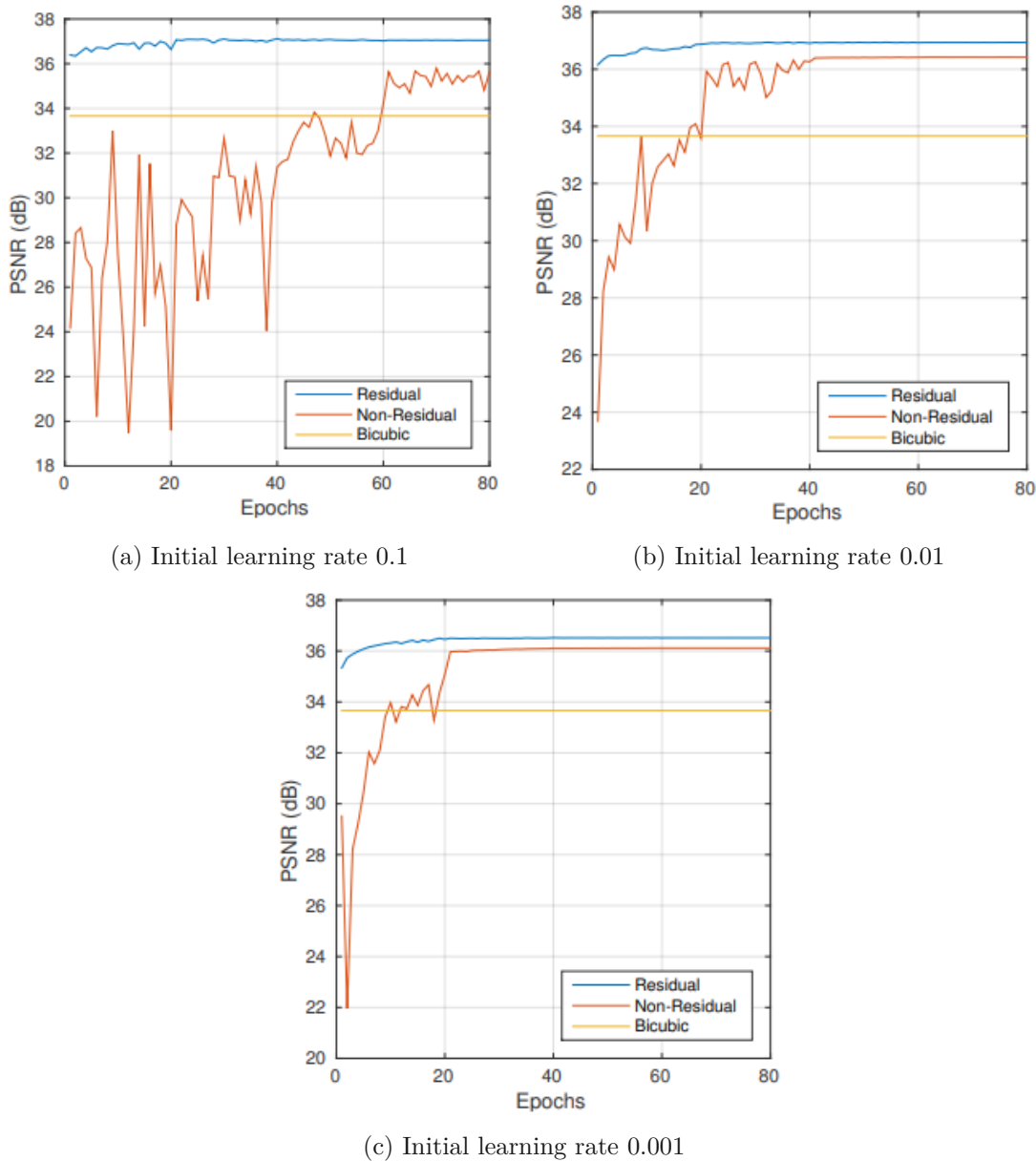


Figure 3.8: Performance curve for residual and non-residual networks. Source: [79]

Unlike typical CNNs, the works presented here do not make use of a pooling layer. This is due to the reason that those layers would reduce the resolution, which is au contraire on what SR is expected to do.

#### 3.5.2 Generative Adversarial Networks

Adversarial training is a framework for deep generative models that make those models competitive in comparison to other deep neural networks. Two main components form this framework: a generator and a discriminator. Given random noise, the generator produces an output which the discriminator needs to classify. Effectively, the discriminator needs to figure out if the output is coming from the distribution of the training data, or not. The generator is rewarded when the discriminator makes a mistake, and vice versa. This simultaneous training mimics a min-max two-player game [23].

The competition between both is what drives each method to improve its model. When the generator is able to fool the discriminator, the discriminator adjusts its model, and vice versa if the discriminator is able to correctly label the data. The models are adjusted by stochastic gradient descent. The training stops once the discriminator is not able to distinguish between both labels, i.e. probability of both labels is  $\frac{1}{2}$ .

A simplified GAN for SR can be observed in Figure 3.9. The generator creates synthesized (super-resolved) images. Both the SR and the HR (real high quality) images are given to the discriminator, which on his part needs to classify which image is real and which is fake.

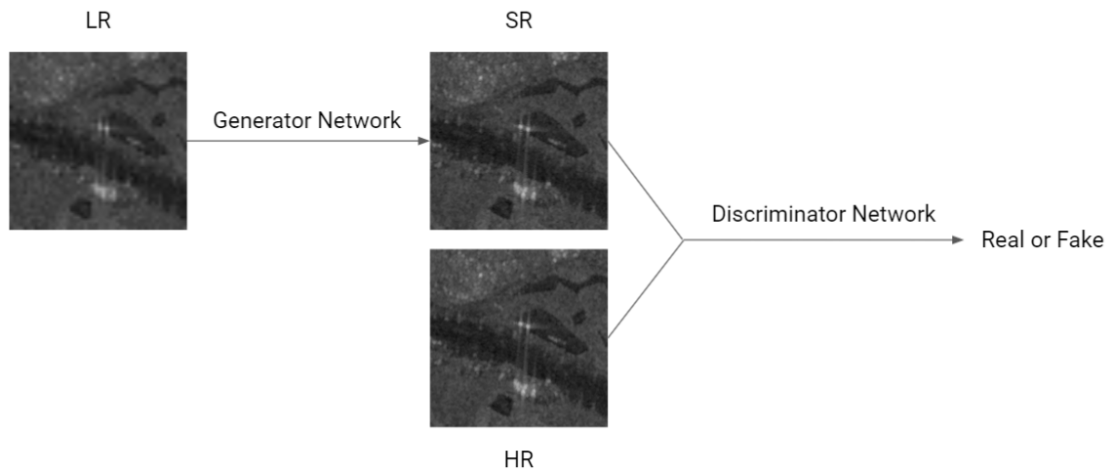


Figure 3.9: A sample GAN used for SR. Source: Own image

Based on GANs, two prominent models (SRResNet and SRGAN) were created by Ledig et al. [89]. They use a deep neural network for both generator and discriminator. In SR, the input for the generator is not random noise. Instead, the low-resolution image is used. Once trained, the generator network is used for creating SR images.

### SRResNet & SRGAN - the Similarities

SRResNet and SRGAN share the same generator and discriminator architecture. The network structures of the corresponding generator and discriminator can be observed in Figure 3.10, where  $k$  denotes the kernel (filter) size,  $n$  the number of feature maps, and  $s$  the stride. For example,  $k3n64s1$  denotes a convolution layer with  $3 \times 3$  filter kernels, 64 feature maps and stride of one.

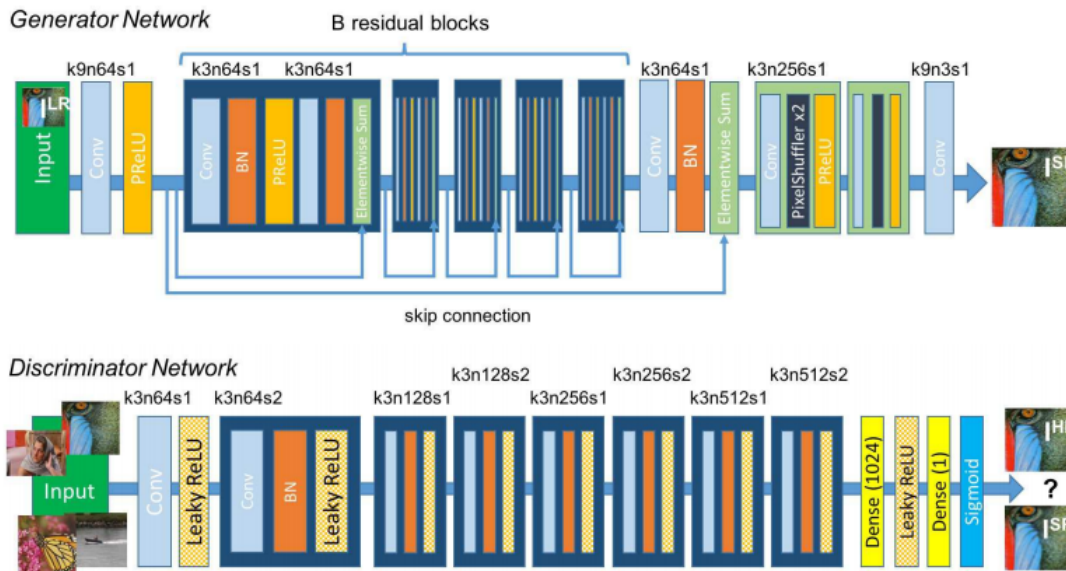


Figure 3.10: Generator and discriminator network architectures of the SRResNet and SRGAN models. Source: [89]

The generator network has 16 identical residual blocks. Each residual block is made of Convolution (Conv) layers, Batch Normalization (BN) layers [148], and PReLU as the activation function. Ledig et al. use in a similar way like the SRCNN and VDSR only filters of sizes  $3 \times 3$  and  $9 \times 9$ .

The discriminator network contains eight Conv layers. The number of feature maps start with 64 and are gradually doubled until 512. LReLU is used with  $\alpha = 0.2$ . Dense (fully-connected) layers are layers in which each neuron receives input from all units of the previous layer. *Dense (1024)* is a dense layer with 1024 neurons.

Contrary to the SRCNN and VDSR networks, which use the bicubically enlarged images, the generators of the SRResNet and SRGAN use the original LR images. Hence, a way is needed to increase the resolution of the input images. To learn the upscaling operation a sub-pixel convolution layer is used (PixelShuffler). Sub-pixel convolution layers speed-up training as opposed to using deconvolution layers [149]. Simultaneously, upscaling at the end of the network instead of in the beginning, leads to more representational power (when comparing two networks with the same number of parameters) [150]. In 3.10 two

sub-pixel convolution layers are used, since the depicted model is created for upscaling images by a factor of 4.

#### SRResNet & SRGAN - the Differences

In their works, Ledig et al. [89] came to the conclusion that MSE (which is used in SRCNN, VDSR and SRResNet), as defined in Equation 3.1, is not the best loss function for creating images pleasant for the human perception. Based on the works of Johnson et al. [151] and Bruna et al. [152], in place of MSE they used the so-called *perceptual loss function*. They argue that minimizing MSE encourages the model to select a solution containing pixel-wise averages of plausible solutions. Those solutions are overly-smooth and have low perceptual quality.

The perceptual loss ( $L_{percep}$ ) is defined as a weighted sum of the content loss ( $L_{content}$ ) and the adversarial loss ( $L_{adv}$ ), in equation:

$$L_{percep} = \lambda_1 L_{content} + \lambda_2 L_{adv} \quad (3.9)$$

Where  $\lambda_1 = 1$  and  $\lambda_2 = 0$  in case of the SRResNet, and  $\lambda_1 = 1$  and  $\lambda_2 = 0.001$  in case of the SRGAN.  $L_{content}$  is in both cases the MSE loss. The adversarial loss  $L_{adv}$  is defined as the sum of the probabilities of the discriminator over all training samples ( $N$ ):

$$L_{adv} = \sum_{n=1}^N -\log D(G(I^{LR(n)})) \quad (3.10)$$

Where  $D$  is the discriminator network, which yields the probability of the image being the original high-resolution image.  $I^{LR(n)}$  is the  $n$ -th low-resolution image.

In the SRResNet, the  $L_{content}$  is the pixel-wise MSE loss. However, in the SRGAN, they utilize the so-called *VGG loss* based on the ReLU activation layers of the pre-trained 19-layer VGG network [124]. In the VGG network, the image dimension is reduced by stacking multiple convolutional and max pooling layers, which extracts higher-level features. The loss is calculated as the Euclidean distance of the VGG-features between the high-resolution image  $I^{HR}$  and generated image  $I^{SR}$ :

$$L_{VGG_{i,j}} = \frac{1}{W_{i,j} H_{i,j}} \sum_{w=1}^{W_{i,j}} \sum_{h=1}^{H_{i,j}} \|\phi_{i,j}(I^{HR})_{w,h} - \phi_{i,j}(I^{SR})_{w,h}\|^2 \quad (3.11)$$

Where  $W_{i,j}$  and  $H_{i,j}$  describe the dimensions of the associated feature maps in the given VGG net.  $\phi_{i,j}$  is the feature map obtained by the  $j^{th}$  convolution (after the non-linear function) before the  $i^{th}$  max pooling layer.

## ESRGAN

Based on the SRGAN, Wang et al. introduced the Enhanced SRGAN (ESRGAN) [91]. Concretely, they improved the SRGAN by altering the network architecture, the adversarial loss, and perceptual loss.

The net design is altered in two ways. First, by introducing the residual-in-residual dense block. Second, by removing the batch normalization layers.

The RRDB replaces the residual blocks in the SRResNet and SRGAN architecture. The authors state that RRDB enables the training of a deeper network and helps improve the perceptual quality of the SR images.

Eliminating BN layers reduces the GPU memory usage<sup>1</sup> [153]. Hence, this freed memory enables the usage of more complicated structures, i.e. RRDS. The removal of BNs is also beneficial in the task of deblurring [154], which is a desired quality in SR. BN might be the cause of undesirable artifacts and hindrance of the generalization ability [91]. Following the authors of the ESRGAN, the artifacts are more likely when the net is deep and a GAN is used.

Batch normalization layers are used to normalize the features after the convolutional layers in the residual blocks of the SRGAN. It can be argued that BNs are applicable in the area of target classification rather than the field of SR.

Altering the adversarial loss means to change the way the discriminator learns. Specifically, they make use of a Relativistic GAN (RaGAN) [155], compared to the Standard GAN (SGAN). RaGAN is designed to measure *which image is more realistic* rather than *which image is real or fake* [155, 156, 91]. More precisely, the discriminator network  $D$  in Equation 3.10 estimates the probability of the input being real (the original HR image). In contrast, the discriminator  $D_{Ra}$  in RaGAN estimates the probability that the HR image is more realistic than the SR image. Following [91], mathematically, the discriminator probability for a given input  $x$  is formulated as follows:

$$D(x) = \sigma(C(x)) \quad (3.12)$$

For RaGAN, the probability changes to:

$$D_{Ra}(I^{HR}, I^{SR}) = \sigma(C(I^{HR}) - \mathbb{E}_{I^{SR}}[C(I^{SR})]) \quad (3.13)$$

$\sigma$  is the sigmoid function,  $C(x)$  the output of the discriminator, and  $\mathbb{E}_{I^{SR}}[.]$  the average of the SR data in the mini-batch. The adversarial loss becomes:

$$L_{adv} = \mathbb{E}_{I^{HR}}[\log(1 - D_{Ra}(I^{HR}, I^{SR}))] - \mathbb{E}_{I^{SR}}[\log(D_{Ra}(I^{SR}, I^{HR}))] \quad (3.14)$$

<sup>1</sup>The SRResNet net has about 40% less memory usage when removing the BNs [153].

### 3. FOUNDATIONS OF DEEP LEARNING BASED SUPER RESOLUTION AND SEMANTIC SEGMENTATION

Jolicoeur-Martineau [155] states that the RaGAN significantly improves data quality and the stability of the GAN, without any cost. Wang et al. [91] state that by using the formulation in Equation 3.14 the NN produces sharper edges and precise textures.

The final difference when comparing ESRGAN with SRGAN is the perceptual loss ( $L_{percep}$ ). The equation (3.11) still holds true, where  $\phi_{i,j}$  is the feature map obtained by the  $j^{th}$  convolution before the  $i^{th}$  max pooling layer. However,  $\phi_{i,j}$  is now the feature map *before* the activation function - whereas in SRGAN it was *after*. Figure 3.11 provides an example of how the perceptual feature looks for a given image. The difference between before and after activation can be seen for two different channels. Simultaneously, it resembles the difference between lower (from  $\phi_{2,2}$ , as used in ESRGAN) and higher-level (from  $\phi_{5,4}$ , as used in SRGAN) features.

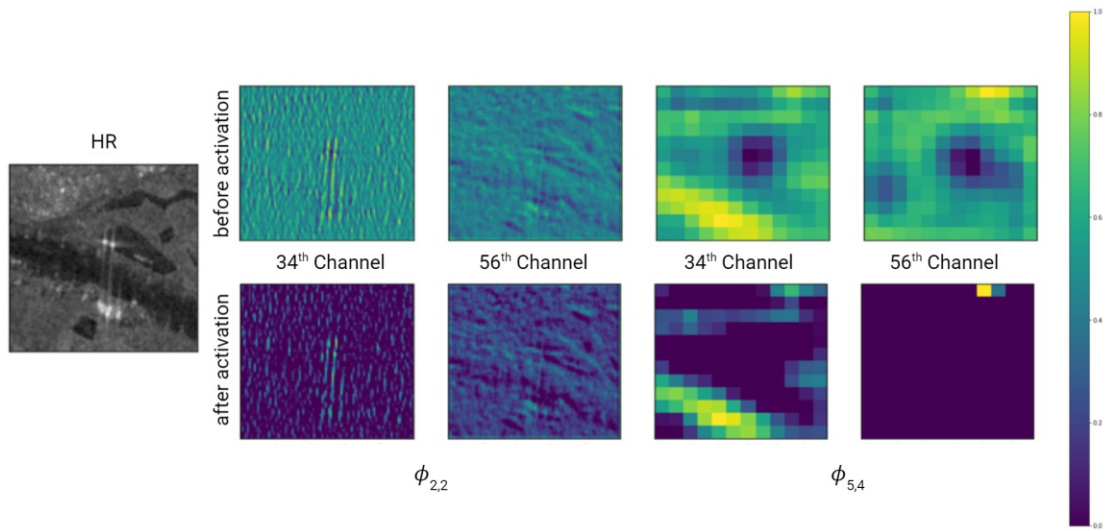


Figure 3.11: Feature maps before and after activation for the 34<sup>th</sup> and 56<sup>th</sup> channels.

The final loss used for the generator is defined as:

$$L_G = \lambda_1 L_{perceptual} + \lambda_2 L_{adv} + \lambda_3 L_1 \quad (3.15)$$

#### EESRGAN

Edge-Enhanced Super-Resolution (EESRGAN) [92] is inspired by the Edge Enhanced GAN (EEGAN) [90] and ESRGAN [91]. Both the EESRGAN and EEGAN were proposed in the field of remote sensing, hence, they are highly relevant for this study.

The authors of both the EESRGAN and EEGAN state that the SR models based on DL miss high-frequency edge information. To solve this problem, they suggest the usage of an Edge-Enhancement Network (EEN). The EEN is integrated as an intermediate network in the generator network. Its goal is to, as the name suggests, improve the borders

of the image by removing noise and artifacts [90]. More precisely, in the EESRGAN, the SR image is now not generated directly by the generator, but within the EEN. In context to the EESRGAN, the image that is produced by the generator is referred to as Intermediate SR (ISR) image, whereas the image produced by the EEN as *SR*.

EEN can be seen in Figure 3.12. *Laplacian* is the Laplacian operator, which is used for edge extraction [157]. The Upsampling Block is equivalent to nearest-neighbor interpolation. The rest of the network is composed of Conv, RRDB, and activation functions. It can be observed that extracted edges (by the Laplacian operator) are subtracted from the ISR image and at the end of the network, the Enhanced Edge is added. Hence, the goal of the network is to learn those *new* improved edges which substitute the original edges.

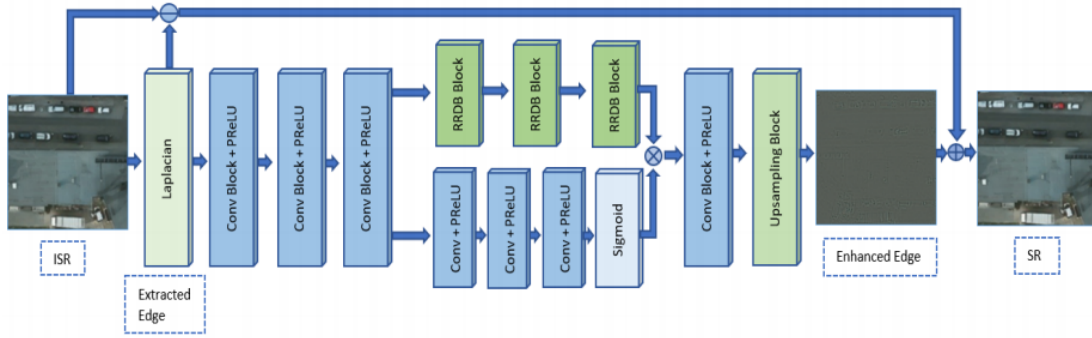


Figure 3.12: Edge-enhancement network used in the EESRGAN. Source: [92]

To increase the quality of the reconstructed images [90] and reduce the artifacts [158], a Charbonnier function [159] is employed. The Charbonnier function is an edge-preserving regularization, which is used to avoid smoothing of the edges in images [159]. The function employed as the Charbonnier penalty function is defined as  $\rho(x) = \sqrt{(x^2 + \epsilon^2)}$ , where  $\epsilon = 0.001$ , as utilized by [158] for SR.

The Charbonnier function is used two-fold. First for image consistency (Eq. 3.16), and second for edge consistency (Eq. 3.17).

$$L_{img\_cst} = \rho(I^{HR} - I^{SR}) \quad (3.16)$$

$$L_{edge\_cst} = \rho(I^{HR\_edge} - I^{SR\_edge}) \quad (3.17)$$

Where  $I^{HR\_edge}$  and  $I^{SR\_edge}$  are the edges of  $I^{HR}$  and  $I^{SR}$  extracted by the Laplacian operator, respectively.

The Charbonnier loss is then defined as:

$$L_{char} = L_{img\_cst} + L_{edge\_cst} \quad (3.18)$$



### 3. FOUNDATIONS OF DEEP LEARNING BASED SUPER RESOLUTION AND SEMANTIC SEGMENTATION

The training of the generator and EEN networks is done simultaneously. The overall loss of ESRGAN (3.15) is extended by the Charbonnier loss. Thus, the loss is defined as:

$$L_G = \lambda_1 L_{perceptual} + \lambda_2 L_{adv} + \lambda_3 L_1 + \lambda_4 L_{char} \quad (3.19)$$

$L_{perceptual}$  is defined as in SRGAN and ESRGAN.  $L_{adv}$  is defined as in ESRGAN.  $L_1$  is the criterion from Equation (3.3).

The discriminator  $D_{Ra}$  is the same as the one from ESRGAN.

#### 3.5.3 U-Nets for Semantic Segmentation

U-Nets in this work are used for pixel-wise image segmentation - assigning a label to each pixel. An example of semantic image segmentation can be observed in Figure 3.13. Each class has a distinguished color.

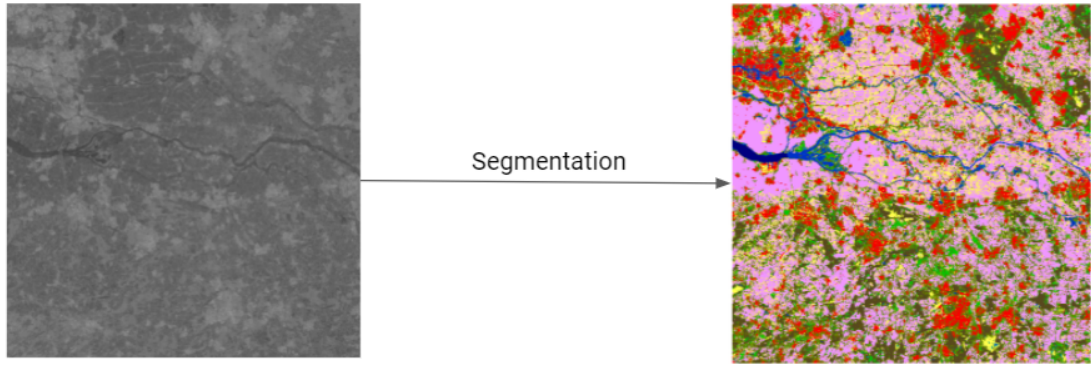


Figure 3.13: Pixel-wise image segmentation. Source: Own illustration based on the TU Wien Sentinel-1 datacube data.

#### U-Net

Ronneberger et al. [125] propose the U-Net as a semantic image segmentation network which outperforms prior models without requiring thousands of annotated training samples. Figure 3.14 demonstrates a general U-Net architecture. The input image has a resolution of  $1 \times H_1 \times W_1$ , where 1 is the number of channels,  $H_1$  and  $W_1$  are the input height and width, respectively. White boxes represent the concatenation with the skip-connected layer.  $F$  represents the number of feature maps.

The architecture consists of two paths: a contracting part (reducing the resolution) to generate high-level features and an expanding path (increasing the resolution) for localization [160]. The localization happens as the contextual information is added to the upsampled layers by skip-connections.



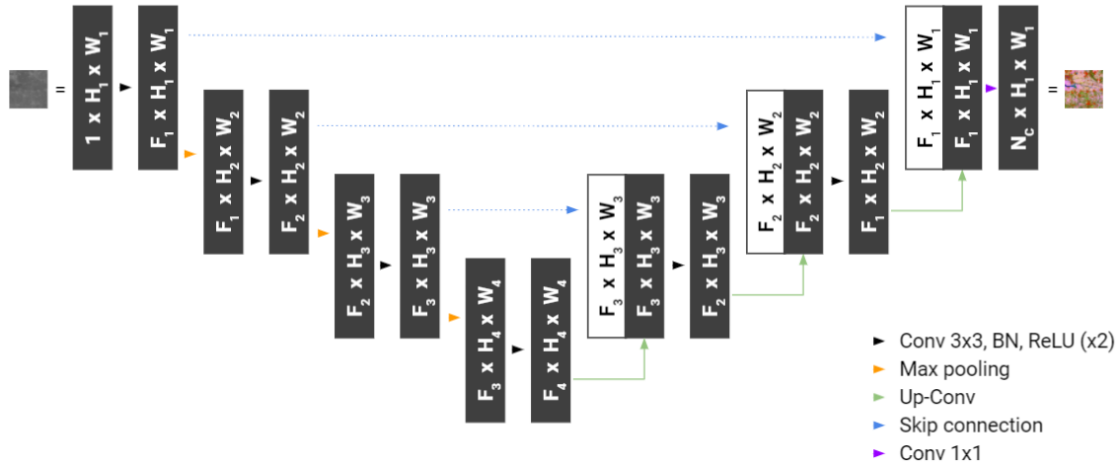


Figure 3.14: U-Net architecture. Source: Own image based on [160]

Features are extracted by the following layer configuration:  $3 \times 3$  Conv, BN, and ReLU. ( $x2$ ) in the legend symbolizes that the layer configuration is done twice. Since the Conv has a padding of one, the layer configuration increases the number of feature maps, while leaving the height and width unchanged. Downscaling is done by applying a  $2 \times 2$  max pooling operation [161]. Max pooling with stride two halves the input resolution for each feature map. Hence,  $H_i = H_{i-1}/2$ , for  $i \in \mathbb{N}_2$ . Whereas in the expanding path, Up-Conv increases the width and height, while halving the number of features. Up-Conv consists of upsampling, which is done by the nearest neighbor algorithm, a  $2 \times 2$  convolution, which halves the number of feature maps, a BN, and a ReLU.

Since the feature maps are calculated based on different scales, the contracting part yields a multi-level, multi-resolution feature representation [162]. The final layer is a  $1 \times 1$  Conv which transforms the number of feature maps to equal the number of classification classes.

### Attention U-Net

Oktay et al. [127] propose the Attention U-net as an improved version of the U-Net model, which already had very good performances on various segmentation applications [125]. The Attention U-Net is suited for multi-class semantic segmentation and is selected as a measure to evaluate the performance of the SR networks.

Figure 3.15 depicts the architecture of the Attention U-Net. The novelty of the Attention U-Net compared to the U-Net is the attention gate. Attention gates suppress irrelevant features while focusing on the important ones [160].

Figure 3.16 schematically shows the structure of an attention gate. An attention gate takes as input the features ( $x$ ) propagated by skip connections and the layers ( $g$ ) in the expanding phase after the Cov layers.  $g$  is the gating signal as depicted in the figure.

### 3. FOUNDATIONS OF DEEP LEARNING BASED SUPER RESOLUTION AND SEMANTIC SEGMENTATION

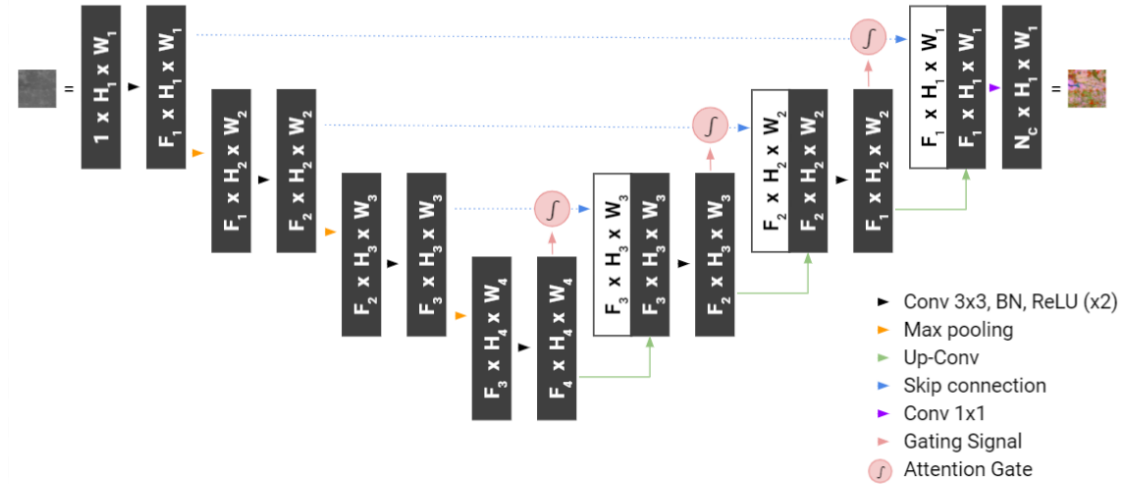


Figure 3.15: Attention U-Net architecture. Source: Own image based on [125]

It has better feature representation compared to  $x$ , however, it is missing the spatial representation. Since  $x$  and  $g$  do not have matching shapes, they need to be transformed. For better explanation, assume  $x$  and  $g$  are the first layers that go through the activation gates, i.e. are the layers symbolized as  $F_3 \times H_3 \times W_3$  and  $F_4 \times H_4 \times W_4$ , respectively.  $x$  passes through an Up-Conv, which scales in the given example  $H_4 \times W_4$  to  $H_3 \times W_3$ . Thereafter, both  $x$  and  $g$  pass through a  $1 \times 1$  Conv, so that both have the same number of feature maps  $F$ . For this reason, they can be summed. While ReLU suppress the irrelevant features, the sigmoid activation function encourages the relevant ones. The final  $1 \times 1$  Conv scales the number of feature maps to equal  $F_3$ , so that a multiplication with  $x$  is possible.

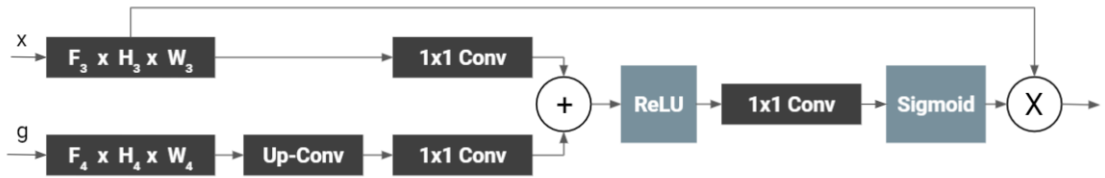


Figure 3.16: Attention gate structure. Example based on the skip connection from  $F_3 \times H_3 \times W_3$  and gating signal from  $F_4 \times H_4 \times W_4$ . Source: Own image based on [125]

## 3.6 Summary

Key loss functions, such as MSE, MAE, perceptual, adversarial, Charbonnier, and VGG were defined. The building blocks of SR were presented, i.e. sub-pixel convolutions, residual blocks, dense blocks, residual dense blocks, and residual-in-residual dense blocks. The difference between low-level (color, gradient orientation) and high-level features

(texture, edges, structures), their importance in reconstructing high-resolution images, and how they are affected by the different building blocks was presented.

It was shown how residual-learning works, and how it naturally fits the goal of SR. It was explained why working with the original LR images is better than working with the upscaled versions thereof, i.e. achieving a larger receptive field when using the same number of model parameters.

The methods, which are going to be used in the experiments of this work, have been depicted. Their differences have been reviewed and discussed, i.e. using the original images or the bicubically upscaled version, the model architectures and how to enable their training. Two types of networks are used for SR - CNNs and GANs. SRCNN, VDSR, and SRResNet are the CNN models. Whereas SRGAN, ESRGAN, and EESRGAN are the GAN networks.

U-Nets for semantic image segmentation have been introduced. The Attention U-net is used for the pixel-wise segmentation in the further experiments, as it is an improved version of the U-Net which performs good on segmentation problems.



# CHAPTER 4

## Methodology

In this chapter, the data at hand is described. Furthermore, the available classes and their distribution are listed. Section 4.2 illustrates the extent of the study together with the sites used for analysis. Section 4.3 showcases how data is pre-processed and split into training, testing, and validation datasets. Section 4.4 covers the topic of data augmentation, which helps tackle the inherent issue of class imbalance. Section 4.5 showcases how the VGG loss is altered for single-channel images.

The training configuration of each models used in this work is described in Section 4.6. This includes learning rate, batch size, optimizer, and loss functions. The metrics used to evaluate the models based on image and segmentation quality are outlined in Section 4.7.

### 4.1 Data

The data used in this work was acquired by the SAR instrument of the two satellites Sentinel-1A and Sentinel-1B of the Copernicus mission. The satellites operate at C-band, with a central frequency of 5.404 GHz [163, 164]. It is an active SAR radar, hence, it emits electromagnetic energy. Each pixel's value depends on the intensity of the reflected radar signal. This type of data is called backscatter. The data used in this thesis was acquired in the interferometric wide swath mode and consists of VV and VH polarized data. The images are displayed in gray scale when only one channel (VH or VV) is provided. An illustration of a VV and VH image stack (two-channel image) is provided in Figure 4.1, based on the tile "E045N021T1" of the Equi7 Grid system [165].

It was decided that only VV data is used for this experiment. It is argued that due to the similarity between VV and VH data, if SR is viable for VV, then it will also be suitable for VH data.

All SAR data, including the reference data for semantic segmentation, is contributed by the Microwave Remote Sensing research group of TU Wien using the TU Wien Sentinel-1

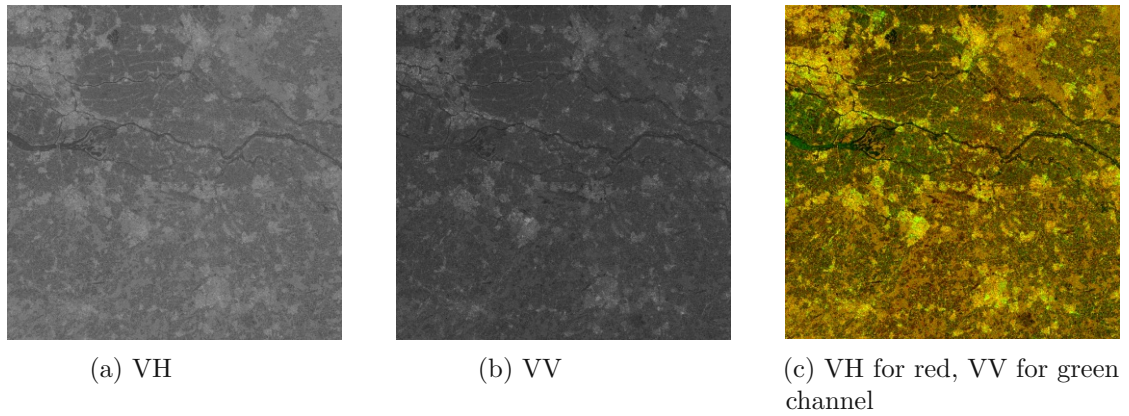


Figure 4.1: Tile "E045N021T1" presented in three different configurations: (a) using VH backscatter information, (b) using VV backscatter information, and (c) using (a) as the red and (b) as the green color channel.

datacube. The SAR data is provided in the GeoTIFF format and is made up of 2-D images. The dimension of each image is  $10000 \times 10000$  pixels with 10 meters GSD. Hence, each image covers an area of  $100 \times 100$  kilometers.

### 4.2 Study Site

The data is provided in the Equi7 Grid. In this study, the two tiles "E045N021T1" and "E051N015T1" are used. The selection is based on the criteria to cover different topography and land covers.

Tile "E045N021T1" is used from two different points in time - 30<sup>th</sup> of March and 13<sup>th</sup> of November 2018. The tile covers a flat area of the Netherlands. For convenience, throughout this work, "E045N021T1" from March is referred to as *March* scene, while the same tile from November is referred as *November* scene. "E051N015T1" was taken on the 2<sup>nd</sup> of June 2018 over a mountainous area in Austria. For convenience, this tile is called *Mountains* scene.

A map with the extent of the study site for March can be observed in Figure 4.2. The Mountains study site is illustrated in Figure 4.3. Due to the strong similarity between the March and November sceneries, the November study map is depicted in the appendix (Figure A). Each figure illustrates the scene and the corresponding areas which are investigated more thoroughly throughout this study.

The reference data for the semantic segmentation is available as part of the released Collection 2 of the Copernicus Global Land Cover layers [166]. It is resampled from 100m to 10m resolution for it to match the HR data at hand. It should be noted that the land covers are from 2015. Furthermore, following the corresponding product user manual, errors in the ground truth are present [167]. This is due to the reason that algorithms

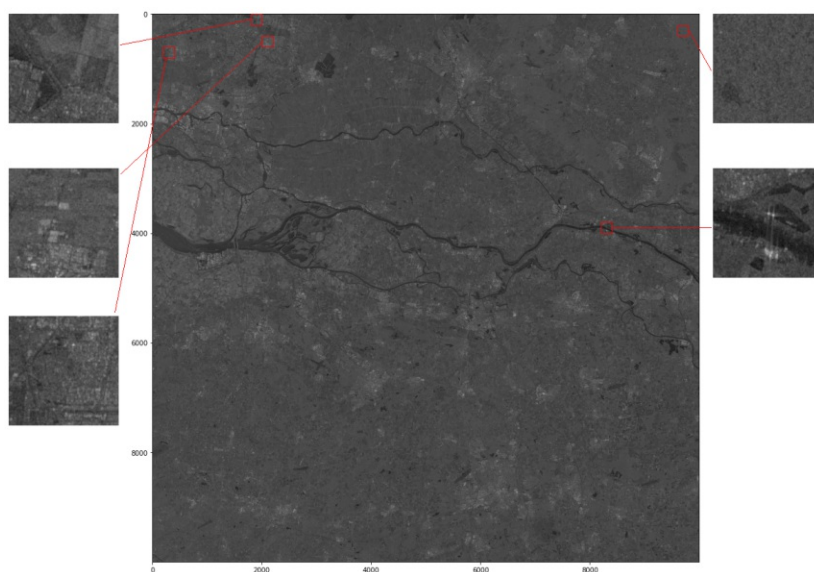


Figure 4.2: Extent of the study site of the March scene.

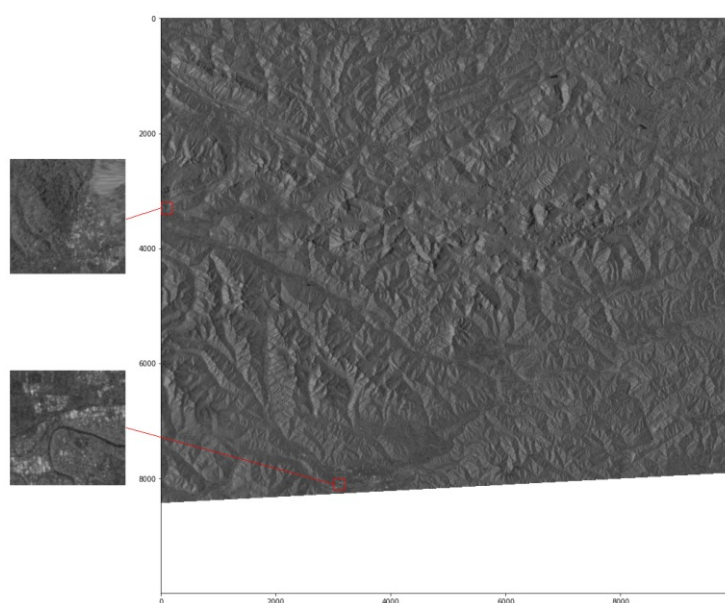


Figure 4.3: Extent of the study site of the Mountains scene.

were used to create the semantic map. Figure 4.4 illustrates the segmentation maps of the two tiles in this study.



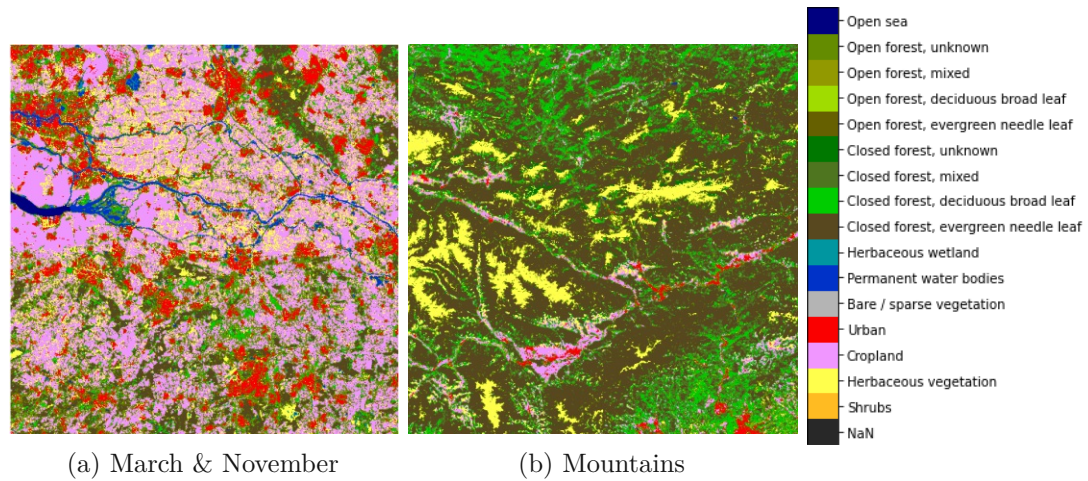


Figure 4.4: Semantic maps used in the experiments for the tile: (a) March & November and (b) Mountains

### 4.3 Data Pre-processing and Splitting

The authors of the models that are evaluated in this work have used different patch sizes for the HR images. For instance, Dong et al. [78] (SRCNN) use images of  $36 \times 36$ px, Kim et al. [79] (VDSR) use  $41 \times 41$ px, Rabbi et al. [92] (EESRGAN) use  $64 \times 64$ px, Ledig et al. [89] (SRGAN) use  $96 \times 96$ px, and Wang et al. [91] (ESRGAN) use up to  $192 \times 192$ px patches. For better comparison of the SR images, all experiments are conducted with the same patch size, namely  $200 \times 200$ px. The size has been selected since it divides the remote-sensing images of  $10000 \times 10000$ px without remainder and the need of overlapping to 2500 patches. The increase in patch size is favorable, since using larger patches leads to better performance, especially deeper networks benefit more than shallower networks [91].

Following the SR competitions Nitre2017 [168] and Prim2018 [146], the data used for the experiments is split semi-randomly into train (80%), validation (10%) and test (10%) sets. Train is used for training, validation for measuring the models performance while training, and test for evaluating data not seen before.

The data is split semi-randomly due to the fact that while splitting the March tile into patches every  $10^{th}$  patch is used as validation, starting from the  $5^{th}$  patch. Starting from the  $10^{th}$  patch, every  $10^{th}$  patch is used as test. The rest is used as the train dataset, besides the first and second occurrences of a patch containing the *bare* class, which are set as validation and test data, respectively. Similarly, from the first three occurrences of the class *shrubs* one is set aside as validation, the rest as test. This is done due to the fact that there are in total only 15 and 9 patches containing the classes *shrubs* and *bare*, respectively. Without doing so, the split yields zero samples of those minority classes for the validation or test sets. Patches containing Not a Number (NaN) values are skipped.



The splitting yields in total 5 datasets. A train dataset containing 2001 March patches (before augmentation). A validation set containing 248 March patches. A test dataset containing 250 March patches. A test dataset containing 2499 November patches. A test dataset containing 2018 Mountains patches.

The values of the tiles are transformed to floats in the range between 0 and 1 by using the formula:

$$X_{scaled} = \frac{(X - \min(X))}{\max(X) - \min(X)} \quad (4.1)$$

Where  $X$  and  $X_{scaled}$  are the input and output of the scaler, respectively. For the scenes November and Mountains, the min and max are taken based on the March scene, to avoid data leakage.

After pre-processing and splitting, the test datasets for the March, November, and Mountains scenes can be observed in Figure 4.5. The number of samples in the March test dataset is substantially lower in comparison to the other two tiles, as 90% of it are used for training and validation. The November and Mountains acquisitions are used solely for testing.

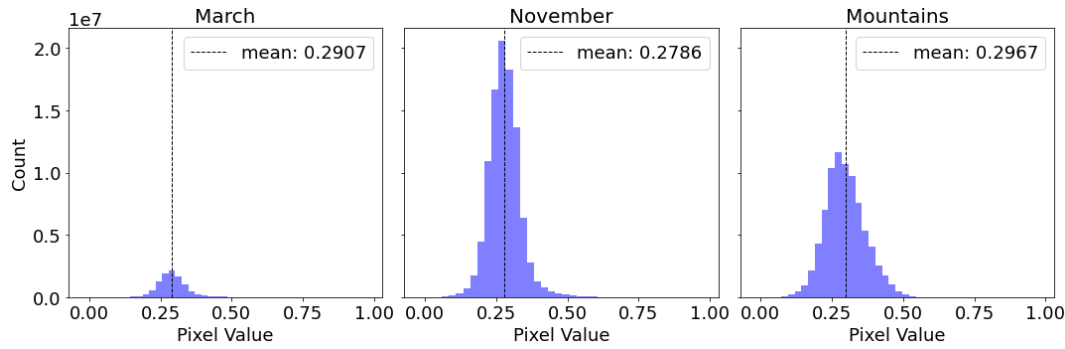


Figure 4.5: Gray value distribution of the test data of the scenes March, November, and Mountains.

It can be observed that the November scene has more values near 0 and above 0.50 than the Mountains scene, due to the fact that since the November tile is more urbanized, which leads to higher backscatter values. The second reason is that it contains more water bodies, which reflect less in the direction of the radar in contrast to urbanized areas.

Both the super resolution and the semantic segmentation task share the same training, validation, and test datasets. Table 4.1 contains the classes available in the datasets. The pixel count is related to the training dataset. Colors of each class stay unchanged throughout this work. The class with the most examples is cropland, which has 30538 times more samples than the class with the least examples - bare / sparse vegetation.

This indicates the inherent problem of class imbalance. In case of skewed data, Machine Learning (ML) systems may have troubles learning the concepts related to the minority classes, which may lead to a performance gap between the majority and minority classes [169].

















#	Name	Color	Count	Share
1	Cropland		27942665	36.083%
2	Open forest, unknown		18323635	23.662%
3	Herbaceous vegetation		8778472	11.336%
4	Urban		8545059	11.034%
5	Closed forest, evergreen needle leaf		6223785	8.037%
6	Closed forest, deciduous broad leaf		2117285	2.734%
7	Open forest, evergreen needle leaf		1528481	1.974%
8	Permanent water bodies		1484678	1.917%
9	Herbaceous wetland		808403	1.044%
10	Closed forest, mixed		678319	0.876%
11	Open forest, deciduous broad leaf		465237	0.601%
12	Closed forest, unknown		296076	0.382%
13	Open sea		217311	0.281%
14	Open forest, mixed		26321	0.034%
15	Shrubs		3358	0.004%
16	Bare / sparse vegetation		915	0.001%

Table 4.1: List of classes with the corresponding colors as well as count and share of pixels based on the training dataset.

## 4.4 Data Augmentation

Increasing the size of the training set by using data augmentation can yield better results [170]. Data augmentation also helps handling the problem of class imbalance [171], which is an issue for the semantic segmentation task at hand, as the distribution of the classes is skewed.

However, it is unclear how much augmentation is too much [172]. In a pixel segmentation task, Liu et. al [173] found out that the data additionally generated should be between 30 and 70%. The benefit of augmentation is higher for the minority classes [173].

The authors of both the SRGAN and ESRGAN make use of 90 degree rotation and

horizontal flip. Additionally to those two transformations, vertical flip, 180 degree rotation, and 270 degree rotation are used. The data augmentation is done for all patches containing less than 25% of the majority classes (cropland and open forest unknown). Hence, in total 1485 additional training images are generated, which equals to about 60% of the total training samples. SR and semantic segmentation is learned based on the same augmented training dataset.

Data distribution before and after augmentation can be seen in the Figure 4.6a and 4.6b. It can be observed that the share of the majority classes (Cropland and Open forest unknown) is reduced by 10% and 5% after the augmentation, respectively. Despite the effort of data augmentation, the dataset is still imbalanced. A perfect class balance would be  $16/100 = 6.25\%$ .

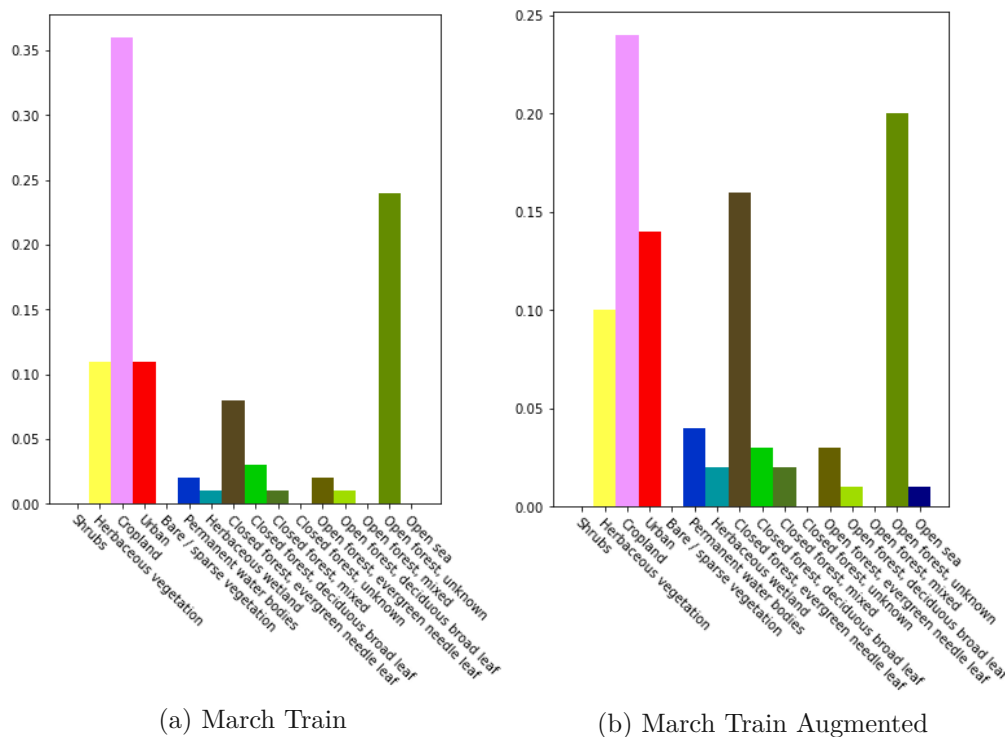


Figure 4.6: Distribution of the classes in datasets: (a) March Train, (b) March Train Augmented.

The distribution in the test datasets can be observed in Figure 4.7. It can be seen that the class distribution of March Train 4.6b, March Test 4.7a, and November Test 4.7b is very similar. This is as expected, since both the March and November scenes share the same ground truth mask. Mountains Test 4.7c is not only imbalanced, but the classes are in contrast to the March dataset, on which the segmentation network is trained. Hence, some classes are poorly characterized by the train dataset.

An alternative to data augmentation for handling the problem of class imbalance is

## 4. METHODOLOGY

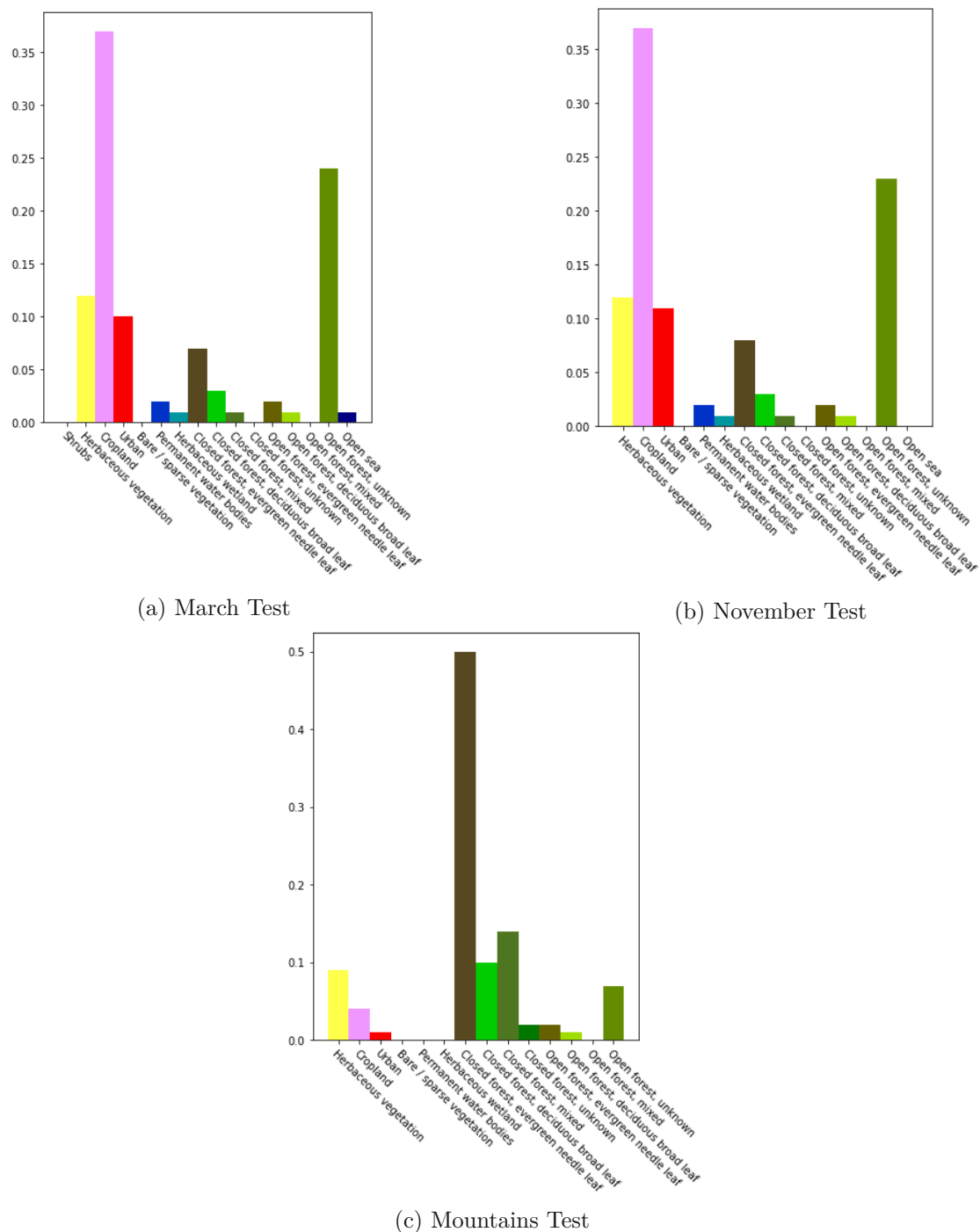


Figure 4.7: Distribution of the classes in datasets: (a) March Test, (b) November Test, (c) Mountains Test.

undersampling. Undersampling eliminates examples of the majority class [169]. However, the technique of undersampling is not utilized in this thesis, since on the one hand the reviewed literature uses solely data augmentation. On the other hand, undersampling adds the risk of losing useful information from the removed samples [174]. Nevertheless, undersampling can improve the prediction accuracy on the minority classes [175].

## 4.5 VGG Loss

The VGG loss, as defined in Equation 3.11, plays a crucial role in SRGAN, ESRGAN, and EESRGAN. The VGG loss is based on the pre-trained VGG-19 network [124]. However, this network is trained and made for three-channel images, i.e. RGB. Nonetheless, the data at hand is a single-channel SAR image. Hence, network or input need to be altered.

It is impossible to adapt the pre-trained VGG network to work with single-channel images without retraining. Therefore, the input to the VGG network is adjusted.

Since the VGG loss is applied to both the HR and SR images, both images are modified before being input to the VGG network, such that the number of channels increases to three. In particular, the shape of HR and SR is transformed from  $H \times W \times C$  to  $H \times W \times 3C$  by copying the existing channel values to the other two.

## 4.6 Model & Training Configurations

In this section, the designs and the training configurations of each evaluated model are presented. The experiments are carried out as similar as possible to papers presented in Section 3.5. The main differences are discussed.

The models covered in this chapter are adapted to work with grayscale images by modifying the first and final layers of the NNs. In other words, the filter of the first convolutional layer has an input of one channel instead of three channels. Simultaneously, the filter of the last convolutional layer has one instead of three output channels.

When not additionally noted, training is conducted by using a batch size of eight and the Adam [176] optimizer with betas 0.9 and 0.99.

*Early stopping* describes a technique to halt the NN training before finishing all training steps or epochs. Early stopping is used to avoid overfitting [177].

### 4.6.1 SRCNN

MSE is used as loss function. Learning rate starts with 0.0001, following [178]. The learning rate is reduced by factor 10 every 40 steps.

Early stopping (20 epochs without PSNR improvement) is utilized to avoid performance degradation.

### 4.6.2 VDSR

Training is conducted with MSE as the loss function and utilizing gradient clipping. Learning rate starts with 0.001. The learning rate is reduced by factor 10 every 40 steps.

The early stopping from the SRCNN is used to lower the generalization error.

The final VDSR model and training configurations differ from the works of Kim et al. [79] in terms of learning rate and weight decay. The training was not successful when using the specified weight decay of 0.0001. At the same time it was also not possible to train the network with higher learning rates (0.1, 0.01, 0.001) as recommended by the authors of the VDSR paper due to unstable gradients. The creators of the VDSR reduce the learning rate by factor 10 every 20 steps. However, the learning rate reduction happens every 40 steps in the experiment. This is to counter the fact that the learning rate was strongly reduced in the beginning.

### 4.6.3 SRResNet and SRGAN

The training of the SRGAN is done in two steps. First, the generator network (SRResNet) is trained. Second, the SRResNet is used to initialize the weights of the SRGAN generator network. This initialization is possible, as both the SRResNet and SRGAN share the same generator architecture.

SRResNet is trained for 200 epochs with a perceptual loss following Equation (3.9) with  $\lambda_1 = 1$  and  $\lambda_2 = 0$ . This is equivalent to using the MSE loss. The first 100 epochs use a learning rate of 0.0001, the rest use a learning rate of 0.00001.

The SRGAN has the same learning rates and number of epochs as the SRResNet however, the loss function of the SRGAN is with  $\lambda_1 = 0.36$ , and  $\lambda_2 = 0.001$ . In the SRGAN,  $\lambda_1$  is chosen slightly higher than in the original paper so that both losses are in the same scale.

Following the original paper,  $\phi_{5,4}$  is set after the activation (ReLU) of the layer. This layer is seen as a later layer, focusing on higher-level features, as previously seen in Figure 3.11.

Both SRResNet and SRGAN are made up of 16 residual blocks. Each residual block is compromised of  $3 \times 3$  Conv, BN, PReLU,  $3 \times 3$  Conv, and BN.

### 4.6.4 ESRGAN, ESRGAN<sub>PSNR</sub>, and ESRGAN<sub>Texture</sub>

The ESRGAN, ESRGAN<sub>PSNR</sub>, and ESRGAN<sub>Texture</sub> are three models with the same generator structure. Similar to SRGAN, the generator network of ESRGAN needs to be initialized. The generator network which is used for the initialization of the weights is trained without the discriminator (adversarial loss). In this work it is referred to as ESRGAN<sub>PSNR</sub>. The name is chosen, as the network's goal is to achieve high PSNR values, potentially for the cost of perceptual quality.

Following the original paper, ESRGAN<sub>PSNR</sub> starts with a learning rate of 0.0002 and is decayed by a factor of 2 every 100 steps.  $\lambda_1 = 0$ ,  $\lambda_2 = 0$ , and  $\lambda_3 = 1$ .

Inspired by [179] and [180], the  $\text{ESRGAN}_{\text{PSNR}}$  is extended by adding a texture loss following Equation (3.4). Following [180], the activation layers  $L$  selected for the Gram matrices are 7 ( $\phi_{2,2}$ ), 16 ( $\phi_{3,4}$ ), 25 ( $\phi_{4,4}$ ), and 30 ( $\phi_{5,2}$ ) of VGG-19. This extended model is called  $\text{ESRGAN}_{\text{Texture}}$ . The weight of this additional loss is 10000 for it to be in the same scale as the MAE loss. Besides the loss, the training setup of  $\text{ESRGAN}_{\text{PSNR}}$  and  $\text{ESRGAN}_{\text{Texture}}$  is the same.

ESRGAN uses the VGG loss as the content loss ( $L_{\text{content}}$ ). In comparison to the SRGAN, ESRGAN uses  $\phi_{2,2}$  in the loss function and it takes the feature maps *before* the activation layer. Following the ESRGAN paper,  $\lambda_1 = 1$ ,  $\lambda_2 = 0.005$ , and  $\lambda_3 = 0.01$ .

In total, 16 RRDBs are used. Each RRDB contains 3 RDBs. Each DB is a 5-layer dense block consisting of Conv and LReLU (0.2) layers, besides the last layer, which is only a Conv. Batch size is selected as four, since a higher number leads to *out of memory* errors.

It is important to note that the output of the DB is multiplied with 0.2 when adding it in the RDB (3.4 as a reminder). Similarly, the output of the last RDB is multiplied with 0.2, when adding it all together at the end of 3.5. This scaling by factor 0.2 is called *residual scaling*, as introduced in the Inception-v4 net [181], which prevents training instability [153].

#### 4.6.5 EESRGAN

In the case of EESRGAN, both generator and discriminator are trained simultaneously. The generator loss uses Equation (3.19) with  $\lambda_1 = 1$ ,  $\lambda_2 = 0.001$ ,  $\lambda_3 = 0.01$ , and  $\lambda_4 = 5$ .

Following the original paper, learning rate is set to 0.0001 and is halved every 100 steps. Batch size is two, as a batch size of eight leads to memory errors.

16 RRDBs are used in the generator with the same structure as in ESRGAN. The edge-enhancement network contains six  $3 \times 3$  Conv layers, each followed by a LReLU (0.2). The dense sub branch contains 5 RRDB blocks. The mask sub branch has three  $3 \times 3$  Conv layers, each followed by a LReLU (0.2). At the end of the mask branch, a sigmoid is used as an activation function.

It is important to note that the EESRGAN paper [92] and the corresponding implementation<sup>1</sup> (task of SR) are contradicting each other. Precisely, the EESRGAN loss (Equation (3.19)) is missing the edge-consistency component (3.17). Only their second implementation<sup>2</sup> (task of SR and object detection) included the edge consistency. Since the second implementation includes the corresponding object detection network, and the first implementation is more similar to the task of this work, it was decided that in the experiments the edge-consistency component will be also removed.

<sup>1</sup>[https://github.com/Jakaria08/EESRGAN/blob/master/model/ESRGAN\\_EESN\\_Model.py](https://github.com/Jakaria08/EESRGAN/blob/master/model/ESRGAN_EESN_Model.py) as of 7th of July, 2021.

<sup>2</sup>[https://github.com/Jakaria08/EESRGAN/blob/master/model/ESRGAN\\_EESN\\_FRCNN\\_Model.py](https://github.com/Jakaria08/EESRGAN/blob/master/model/ESRGAN_EESN_FRCNN_Model.py) as of 7th of July, 2021.

#### 4.6.6 Attention U-Net

Following the Attention U-Net representation as of Figure 3.15, the first double convolution yields  $F_1 = 64$ . Each subsequent double convolution doubles the feature maps, i.e.  $F_2 = 2F_1$ . Each max pooling doubles halves the width ( $W$ ) and height ( $H$ ). It is important to note that since the input image is of shape  $H_1 \times W_1 = 200 \times 200$ ,  $W_4$  and  $H_4$  equal to  $25/2$ , which is not divisible without remainder. Hence, the result of the Up-Conv and the block connected by the skip connection do not match. To overcome this issue, the result of the first Up-Conv is zero-padded to have matching shapes.

The number of classes ( $N_C$ ) is 16. The number of epochs is set to 100 with batch size eight. Adam optimizer with learning rate of 0.001 is used. Following [182] (a NN for urban scene segmentation model for HR SAR data), a reduce-on-plateau learning rate schedule is used (factor: 0.5, patience 15, relative threshold: 0.0001). The model, which has achieved the highest precision value on the validation dataset, is chosen.

### 4.7 Metrics

Two types of metrics are used in this work - metrics for segmentation and image quality.

#### 4.7.1 Segmentation Metrics

The metrics used are accuracy, precision, recall, and Intersection-over-Union (IoU). IoU, also referred to as Jaccard index, is a standard metric for segmentation problems, which takes into account the class imbalance issue (that is apparent in the data at hand) [183]. IoU measures the overlap between the predicted segmentation and the given ground truth mask. The referred metrics for class  $k$  are defined as count based measurements by the following equations:

$$Accuracy_k = \frac{TP_k + TN_k}{TP_k + TN_k + FP_k + FN_k} \quad (4.2)$$

$$Precision_k = \frac{TP_k}{TP_k + FP_k} \quad (4.3)$$

$$Recall_k = \frac{TP_k}{TP_k + FN_k} \quad (4.4)$$

$$IoU_k = \frac{TP_k}{FP_k + TP_k + FN_k} \quad (4.5)$$

Where  $TP_k$ ,  $TN_k$ ,  $FP_k$ , and  $FN_k$  denote the true positive, true negative, false positive, and false negative counts of class  $k$ , respectively. In fact, the metrics are calculated for



each class separately and then weighted based on the class weight. The class weight for a given class  $k$  is defined as:

$$w_k = \frac{\text{Pixels of class } k \text{ for given scene}}{\text{Total pixels}} \quad (4.6)$$

Hence, the weighted metric is:

$$\text{Weighted metric} = \sum_{k=1}^K w_k \text{metric}_k \quad (4.7)$$

$\text{metric}_k$  is one of the defined segmentation metrics for class  $k$ ,  $K$  the number of classes,  $w_k$  the class weight. The sum over the weights is 1.

The confusion matrix and the calculation of the TP, TN, FP, and FN for a given class  $b$  is depicted in Figure 4.8. It can be observed that for class  $b$ , all rows and columns not containing class  $b$  are marked as TN. In comparison to TN - TP, FN and FP become insignificant. As noted by Zhang et al. [184], it is easy to get high accuracy in a multi-class segmentation problem. This is the reason why only one metric based on TN is selected for evaluating the segmentation experiments.

		PREDICTED classification				
		Classes	a	b	c	d
ACTUAL classification	a	TN	FP	TN	TN	
	b	FN	TP	FN	FN	
	c	TN	FP	TN	TN	
	d	TN	FP	TN	TN	

Figure 4.8: TP, TN, FP, and FN in case of multi-class segmentation with respect to class  $b$ . Source: [185]

Micro-average and weighted-average are recommended by Singh et al. [186] in case of multi-class classification and class imbalance. In the case of micro-average, the total number of TP, TN, FP, and FN are aggregated for each class, subsequently the desired

metrics are calculated. On the contrary, macro-average first calculates the metric for each class and thereafter yields the final metric by averaging. Thus, macro-average weights all classes equally, without taking into account that some classes might have more or less samples. This is accounted by weighted-averaging, as done by employing the weights  $w_k$ . Hence, weighted-average as described in Equation 4.7 is used in this work.

#### 4.7.2 Image Quality Metrics

To additionally evaluate the performance of the SR models, the metrics Peak Signal-to-Noise Ratio (PSNR) and Structural SIMilarity (SSIM) are used. These are utilized due to the fact that most state-of-the-art papers (Section 2.2) make use of them as quantitative metrics for image similarity. PSNR is an approximation to human perception of reconstruction quality. Essentially, it is a measurement of differences for each pixel. In comparison to PSNR, SSIM does not estimate the absolute errors. It measures the perceived change in structural information and not the pixel values. It makes use of loss of correlation, luminance and contrast distortion [187]. PSNR and SSIM are defined as follows:

$$PSNR(I^{HR}, I^{SR}) = 20 \log_{10} \left( \frac{max_I}{\sqrt{MSE(I^{HR}, I^{SR})}} \right) \quad (4.8)$$

$$SSIM(I^{HR}, I^{SR}) = \frac{(2\mu_{I^{HR}}\mu_{I^{SR}} + C_1)(2\sigma_{I^{HR}I^{SR}} + C_2)}{(\mu_{I^{HR}}^2 + \mu_{I^{SR}}^2 + C_1)(\sigma_{I^{HR}}^2 + \sigma_{I^{SR}}^2 + C_2)} \quad (4.9)$$

Where  $max_I$  is the maximum possible pixel value of the image, in our case it is 1.  $C_1$  and  $C_2$  are two small constants, in this work following [188], they are chosen as 0.0001 and 0.0009, respectively.  $\mu_{I^{HR}}$ ,  $\mu_{I^{SR}}$ ,  $\sigma_{I^{HR}}$ ,  $\sigma_{I^{SR}}$ , and  $\sigma_{I^{HR}I^{SR}}$  are the means, standard deviations, and cross-covariances for the images  $I^{HR}$  and  $I^{SR}$ .

Having a higher PSNR or SSIM is better. However, the image with the higher quantitative measure may not be perceived as the better one by human perception (qualitative measurement) [189, 190, 191].

### 4.8 Summary

The data at hand, in terms of tiles of the Equi7 Grid system, was introduced, i.e. the scenes March, November, and Mountains. Necessary data pre-processing steps were presented. It was shown how the data is split for the experiments.

Data augmentation was introduced as an important vehicle for increasing the training set and reducing its imbalance. The data augmentation techniques used were pointed out, i.e. flips and rotations. It was illustrated how the classes are distributed before and after data augmentation. It was demonstrated that the classes of the Mountains tile are poorly characterized by the train dataset, since it has a very different distribution of classes.

Training and implementation details were described. Parameters and their deviation from the original papers were shown. The metrics used for measuring image quality (PSNR and SSIM) and classification correctness (accuracy, precision, recall, IoU) were formally defined and discussed.



# CHAPTER 5

## Experiments

This chapter presents the experiments. Section 5.1 covers the experimental design used to evaluate the research questions. Section 5.2 depicts the implementation environment. Section 5.3 presents and discusses the results of the different SR methods for each test dataset, i.e. March, November, and Mountains.

### 5.1 Experimental Design

The original SAR images are downsampled by a factor of 2 or 4. Thereafter, the SR models outlined in Section 4.6 are used to generate the corresponding SR images.

For the purpose of comparison, two additional models, beyond the different CNN architectures, are considered: the Adaptive Importance Sampling Unscented Kalman Filter (AISUKF) method [192] and bicubic interpolation. The AISUKF is a state-of-the-art non-DL model for SAR SR, whereas bicubic interpolation will define a basis of comparison for a method without the need of training.

To answer the research question a quantitative analysis is carried out. The upscaled images are compared with the original high-resolution images from the same scene, i.e. March, November, or Mountains, based on an earth observation model for land cover segmentation. The EO model (trained Attention U-Net from Section 4.6.6) is created as part of the master thesis. It detects 16 different land covers (e.g. permanent water bodies, urban or forest) based on the CCI Land Cover dataset<sup>1</sup> [193]. The labeled data (ground truth) are available.

The EO model is trained on the HR data of the March scene. To assess the quality of the SR networks, the EO model will be evaluated once on the HR and once on the SR SAR images. Accuracy, precision, recall, and IoU are used to measure the difference in

<sup>1</sup>Dataset is provided by ESA containing land cover maps from 1992 to 2015.

the model's results. The EO model applied on the HR March, November, and Mountains scenes will serve as a baseline for the corresponding experiment.

The suitability of SR for SAR C-band images is given if comparable results to the March baseline are achieved when using SR images as input to the EO model. It can be argued, that the EO model should be trained on the SR and not the HR data, given that the SR images can have different properties, e.g. be smoother and contain less noise. Therefore, the model trained on the HR image might not work well and the metrics would be thus lower for the SR image. Nevertheless, training an additional EO model for SR data is seen as increased effort if a HR EO model already exists. Therefore, in this thesis SR for SAR C-band images is also identified as viable if an EO model trained on SR data attains comparable results to the March baseline.

Sub-goals of this thesis are to evaluate how well SR models can handle unseen temporal and spatial conditions. A paradigm for unseen spatial conditions is the mountainous region in the Mountains tile. Whereas an example for unseen temporal condition is the change of seasons, i.e. autumn in the November scene. The effect of autumn is visible in Figure 5.1, in which the backscatter of the *Herbaceous vegetation* is strongly reduced in comparison to the *Cropland* and *Urban* land covers of the corresponding area of the March scene. Figure 5.2 exemplifies how closed forest is not affected by the change of the seasons. In contrast, open forest has an increased backscatter, as depicted in Figure 5.3.

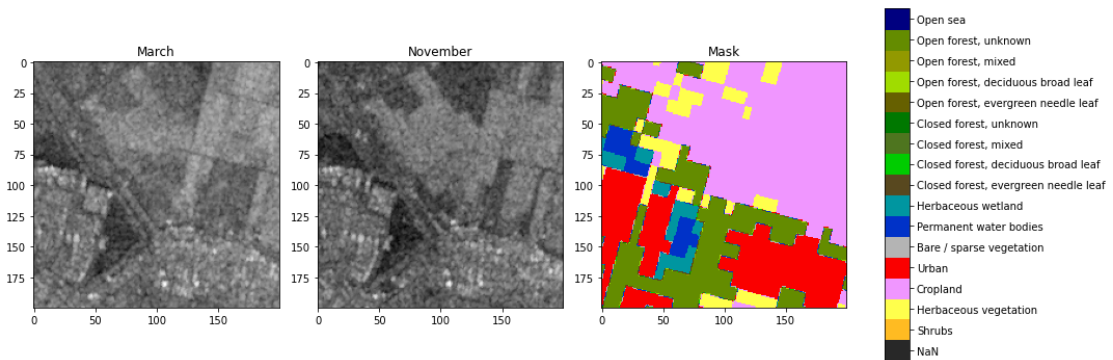


Figure 5.1: Image patch 10 of the scenes March and November, along with the associated segmentation mask

To accept or refute the two hypotheses, the EO model will be applied on the SR images of the November and Mountains scenes. Subsequently, the resulting segmentation maps will be compared with the November and Mountains baselines, respectively. Hence, the SR models are evaluated on conditions without being trained on them.

The experiments consider upscaling by factors of 2 and 4. Super-resolving images from 20m to 10m and 40m to 10m are symbolized as SR x2 and SR x4, respectively.

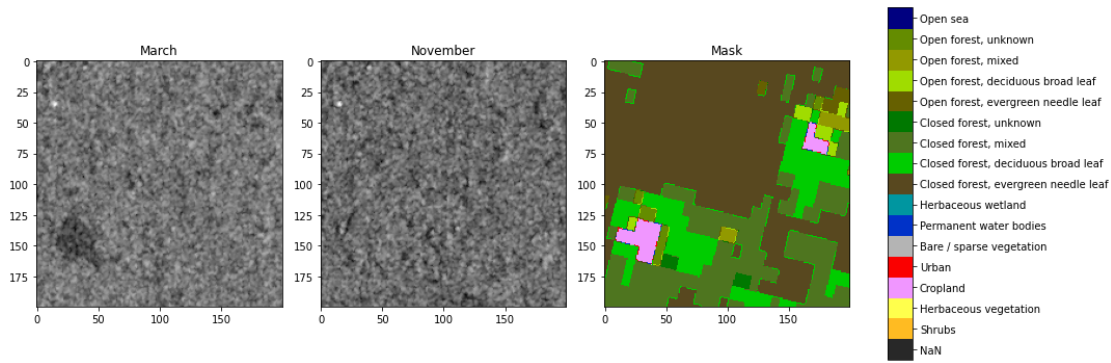


Figure 5.2: Image patch 99 of the scenes March and November, along with the associated segmentation mask

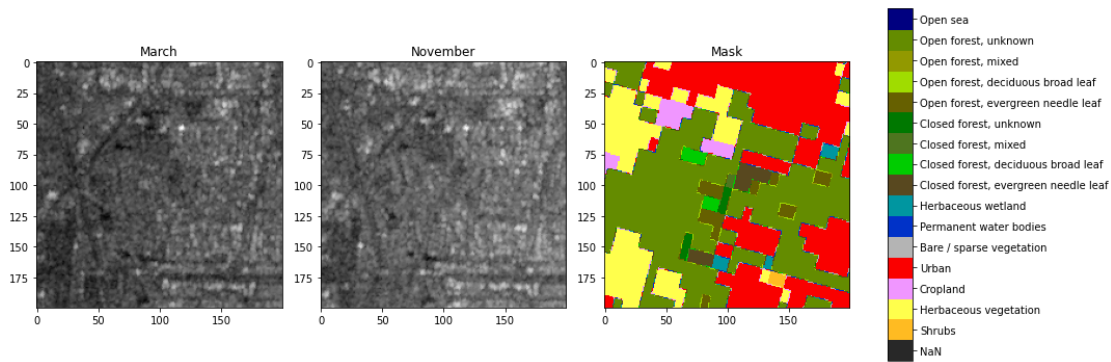


Figure 5.3: Image patch 152 of the scenes March and November, along with the associated segmentation mask

Figure 5.4 visualizes the experimental design of the master thesis. First, the LR data will be created by downsampling the 10m HR images, i.e. 10m to 20m and 10m to 40m. Second, the different SR models will be used to create the SR images from the LR data. Third, the SR and HR images will be compared based on PSNR and SSIM, and at the same time they will be evaluated by the segmentation task with focus on accuracy, precision, recall, and IoU.

## 5.2 Implementation Environment

All implementations are done in the programming language Python utilizing the PyTorch [194] machine learning framework. For reproducibility, random initialization seeds are set to 1 in numpy and PyTorch.

The experiments have been achieved using the Vienna Scientific Cluster 3 (VSC3). The computations are run on a node compromised of NVIDIA Pascal GeForce GTX 1080 GPU, 2× Intel Xeon E5-2650v2 CPUs (2,6 GHz, 8-Core, Codename Ivy-Bridge), 64 GB

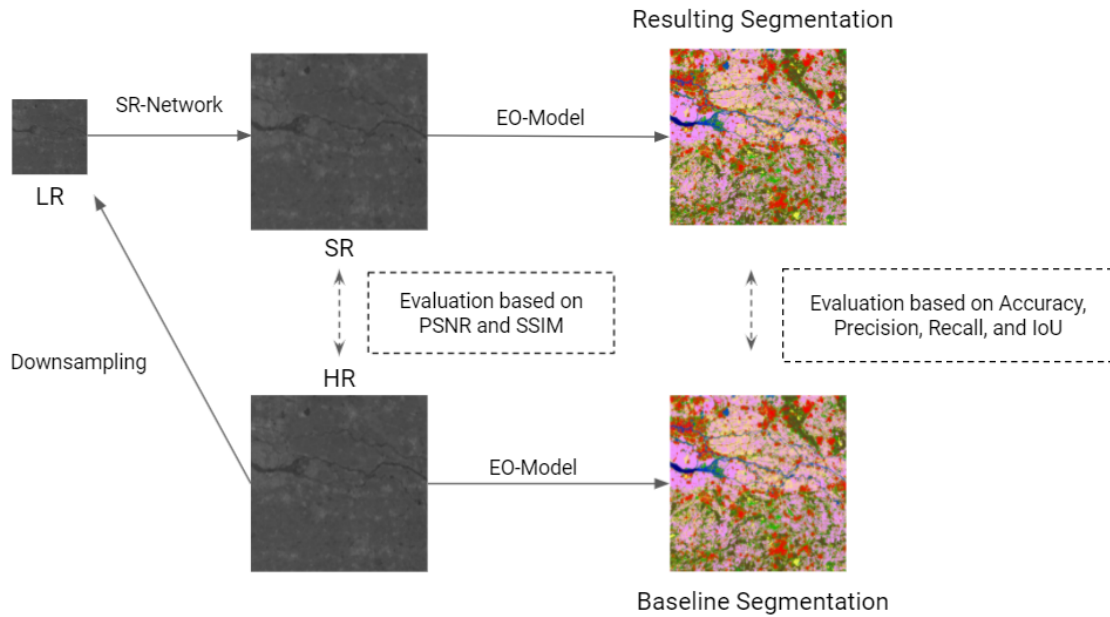


Figure 5.4: Experimental design. SR-Network and EO-Model are trained on the March scene using the HR data. The corresponding tiles, i.e. March, November, or Mountains are used when evaluating the HR and SR images

RAM (DDR3, ECC, 1.866 MHz).

## 5.3 Results

This chapter covers the evaluation of the research questions. Each hypothesis is evaluated once for SR x2 and once for SR x4.

### 5.3.1 EO Model for Semantic Segmentation

Figure 5.5 illustrates the confusion matrix for the predictions of the EO model on the HR images of the March scene. The corresponding segmentation metrics are depicted in Table 5.1, in which division by zero results in NaN. It can be observed that the model does not make any prediction on the sparsely available classes, e.g. *Shrubs*, *Bare / sparse vegetation*, *Open forest*, *mixed*. This is in accordance to Table 4.1, which listed the count of the pixels of each class for the training set. Hence, the EO model itself is not able to perfectly distinguish all the different classes. This explains the lower precision, recall, and IoU compared to the accuracy as seen in the corresponding Sections 5.3.3 (SR x2) and 5.3.4 (SR x4).

Moreover, both the confusion matrix and the corresponding segmentation metrics suggest that the imbalance in the class distribution is a problem even after the data augmentation



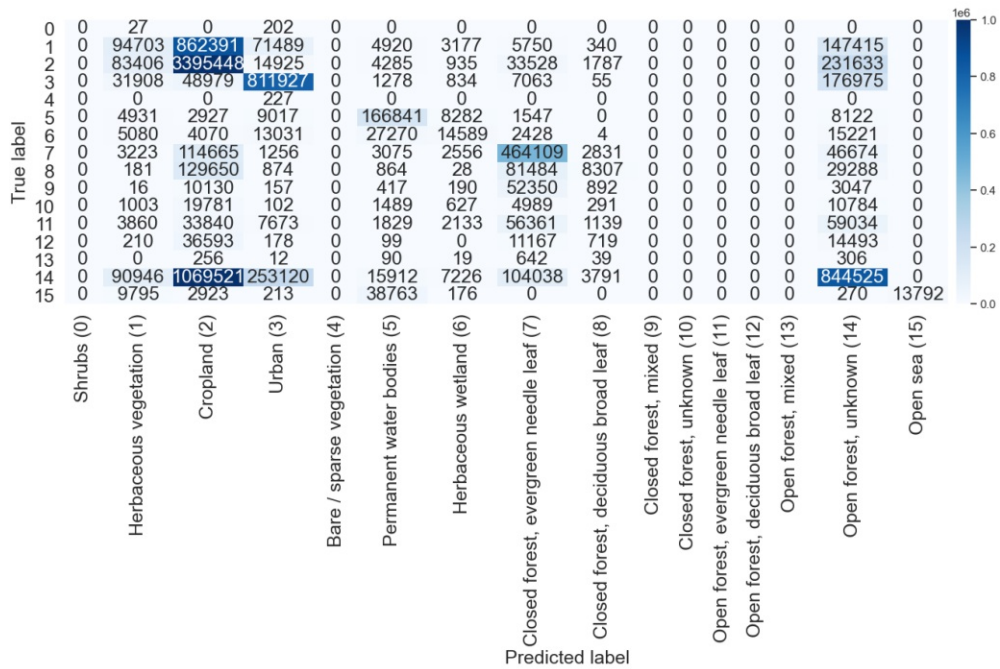


Figure 5.5: Confusion matrix for the evaluation on the HR March scene.

Class	Accuracy	Precision	Recall	IoU
Shrubs	1	NaN	0	0
Herbaceous vegetation	0.867	0.2876	0.0796	0.0665
Cropland	0.7294	0.5925	0.9016	0.5565
Urban	0.936	0.6855	0.7525	0.5594
Bare / sparse vegetation	1	NaN	0	0
Permanent water bodies	0.9865	0.6246	0.8273	0.5525
Herbaceous wetland	0.9907	0.3578	0.1786	0.1352
Closed forest, evergreen needle leaf	0.9464	0.5622	0.727	0.4642
Closed forest, deciduous broad leaf	0.9746	0.4113	0.0331	0.0316
Closed forest, mixed	0.9933	NaN	0	0
Closed forest, unknown	0.9961	NaN	0	0
Open forest, evergreen needle leaf	0.9834	NaN	0	0
Open forest, deciduous broad leaf	0.9937	NaN	0	0
Open forest, mixed	0.9999	NaN	0	0
Open forest, unknown	0.7712	0.5319	0.3535	0.2696
Open sea	0.9948	1	0.2092	0.2092

Table 5.1: List of classes with the corresponding segmentation metrics for the March scene based on the HR data.

as described in Section 4.4. To counteract this, associated classes can be merged. For instance, different forest classes can be grouped together into one *forest* class. Another possibility is to create a two-stage classifier [195]. In particular, the NN of the first stage would predict the grouped classes (e.g. forest). The NN of the second stage would then more precisely segment the grouped class (e.g. Closed forest, evergreen needle leaf). This staged approach is advantageous if the grouped classes are not practical for the EO task at hand.

The confusion matrices and segmentation metrics of the November and Mountains scene are similar. These can be viewed in the appendix (Figures B and C and Tables A and B).

### 5.3.2 Dismissed Model

The AISUKF method is dismissed for the further experiments as it showed poor results. An example upscaled image by the AISUKF method can be seen in Figure 5.6, where interpolation and the original HR image are displayed as reference. The upscaling factor is 2. It can be observed that the classical SAR approach denoises the image, however, at the cost of high-level feature loss and over-smoothing.

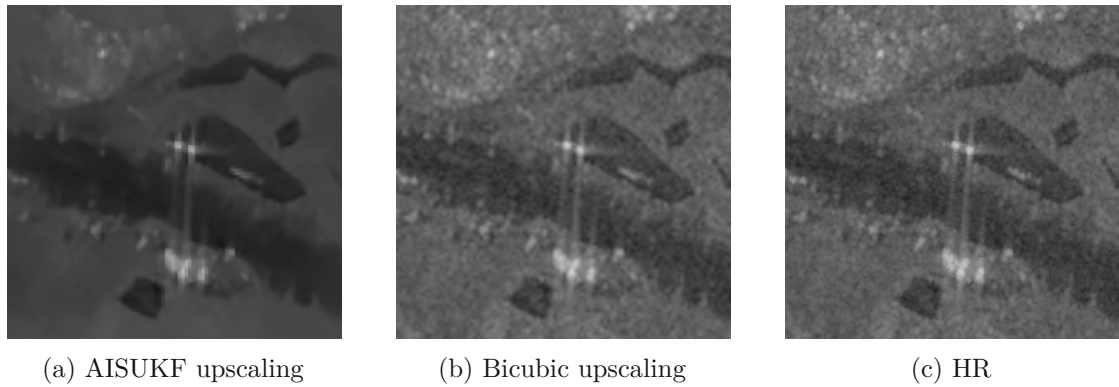


Figure 5.6: Image from March tile presented in three different configurations: (a) Adaptive ISUKF upscaling, (b) Bicubic upscaling, and (c) HR

The AISUKF method is run on MATLAB R2020b with the original code provided by the authors on Github<sup>2</sup>. The model takes a LR image and 15 transformations of it as input. The 15 transformations are generated by random horizontal and vertical shifting by -15 to 15 pixel.

The super-resolved SAR images presented in the AISUKF work [192] showed promising results, hence this method was evaluated. It can be argued that the reason why the method fails is that the SAR used in their dataset has a higher GSD, as cars, boats, or even building antennas can be observed. In contrast, the SAR images used in this work have a GSD of 10, hence, small objects are not visible. It is not stated which SAR dataset the authors used.

<sup>2</sup><https://github.com/sitharavpk/Adaptive-ISUKF>, last accessed on July 15<sup>th</sup>, 2021.

In summary, it can be said that the AISUKF method might be useful when trying to remove the inherent speckle noise. However, one of the goals of this work is to provide a compression and decompression model without large loss of details, which is the case in the AISUKF method. Hence, this method is not evaluated in the further experiments.

### 5.3.3 SR x2

This section includes the quantitative and qualitative results of the SR models when upscaling SAR images by a factor of 2. The section is divided into four subsections, one for each hypothesis of this thesis.

#### Main hypothesis: SR with an upscaling factor of 2 is suitable for SAR C-band images

The quantitative measurements of SR by a factor of 2 can be seen in Table 5.2. The Evaluation tile is depicted above the results. Numbers are rounded off in the fourth decimal place. Best results for each evaluation metric are marked in **bold**. Second best are underlined.

Observing the image quality metrics (PSNR and SSIM) in Table 5.2, it can be seen that SRResNet has the best results for the March and November scenes. In contrast, SRCNN and VDSR achieve the best PSNR and SSIM results on the Mountains tile, respectively.

The aggregated results over all scenes are presented in Table 5.3. The result of each scene is weighted by the count of its pixels compared to the total pixels in all three scenes. The number of pixels are 10000000, 99960000, and 80720000 for the March, November, and Mountain acquisitions, respectively. Equivalently in percent, 5.24%, 52.42%, and 42.33% of the total pixels. Overall, the SRResNet shows best performance compared to other models based on the image metrics. Based on the semantic segmentation metrics, the two best models are ESRGAN<sub>PSNR</sub> and ESRGAN<sub>Texture</sub>.

Corresponding visual results of the SR models for patch 111 of the March scene can be seen in Figure 5.7<sup>3</sup>. Each depicted image contains the PSNR and SSIM metrics compared to the HR image. It becomes clear that it is not easy to spot differences nor to perceptually decide which method works best (when ignoring SRGAN).

<sup>3</sup>Despite having four times less pixels, the LR is depicted as the same size as the HR and SR images. This is the reason why the individual pixels are observed in the LR, which makes the image appear (as it is) as lower-resolution.

## 5. EXPERIMENTS

March scene	PSNR (in dB)	SSIM	Accuracy	Precision	Recall	IoU
HR	100	1	0.8158	0.5267	0.5814	0.3863
Bicubic	37.7048	0.9286	0.7729	0.4534	0.4926	0.2847
SRCNN	37.5998	0.9308	0.7696	0.4512	0.4907	0.2818
VDSR	38.2782	0.9357	0.7752	0.4571	0.5034	0.2955
SRResNet	<b>38.5967</b>	<b>0.9402</b>	0.7844	0.4771	0.5231	0.3170
SRGAN	13.1765	0.1006	0.7497	0.1570	0.0211	0.0015
ESRGAN <sub>PSNR</sub>	37.8991	0.9306	<b>0.7976</b>	<b>0.5067</b>	<b>0.5502</b>	<b>0.3473</b>
ESRGAN <sub>Texture</sub>	37.8428	0.9302	<u>0.7925</u>	<u>0.4930</u>	<u>0.5357</u>	<u>0.3308</u>
ESRGAN	35.3867	0.8974	0.7664	0.4590	0.4111	0.2535
EESRGAN	<u>38.5406</u>	<u>0.9392</u>	0.7824	0.4752	0.5215	0.3144
November scene						
HR	100	1	0.8181	0.5191	0.5550	0.3810
Bicubic	37.5624	0.9282	0.7708	0.4438	0.4848	0.2802
SRCNN	37.4586	0.9304	0.7672	0.4411	0.4811	0.2760
VDSR	38.1400	0.9353	0.7740	0.4506	0.4937	0.2897
SRResNet	<b>38.4551</b>	<b>0.9397</b>	0.7846	0.4677	0.5108	0.3106
SRGAN	13.5922	0.1039	0.7569	0.1520	0.0209	0.0014
ESRGAN <sub>PSNR</sub>	37.7371	0.9295	<b>0.7981</b>	<b>0.4867</b>	<b>0.5342</b>	<b>0.3405</b>
ESRGAN <sub>Texture</sub>	37.6980	0.9294	<u>0.7973</u>	<u>0.4843</u>	<u>0.5298</u>	<u>0.3327</u>
ESRGAN	34.9865	0.8968	0.7499	0.4537	0.3099	0.1904
EESRGAN	<u>38.4131</u>	<u>0.9389</u>	0.783	0.4662	0.5086	0.3084
Mountains scene						
HR	100	1	0.6558	0.4209	0.0752	0.0317
Bicubic	<u>37.1874</u>	0.9225	0.6658	0.4186	<u>0.0776</u>	<u>0.0393</u>
SRCNN	36.9932	<b>0.9234</b>	0.6642	0.4218	0.0718	0.0338
VDSR	<b>37.2511</b>	<u>0.9228</u>	0.6636	0.4211	0.0722	0.0336
SRResNet	37.0602	0.9199	0.6657	<u>0.4241</u>	0.0769	0.0390
SRGAN	12.3143	0.0913	<b>0.6894</b>	0.0133	0.0013	0.0008
ESRGAN <sub>PSNR</sub>	36.2105	0.9044	0.6579	<b>0.4288</b>	0.0651	0.0280
ESRGAN <sub>Texture</sub>	35.9431	0.8990	<u>0.6663</u>	0.4028	<b>0.0839</b>	<b>0.0464</b>
ESRGAN	34.2602	0.8732	0.6390	0.3764	0.0540	0.0104
EESRGAN	36.9094	0.9175	0.6634	0.4249	0.0741	0.0358

Table 5.2: SR results for upscaling factor of 2. All training is done based on the March scene. Evaluation is done separately for each scene.

To be able to visually distinguish the differences of the predictions, parts of the images are zoomed in. More precisely, the patch from Figure 5.7 is displayed in Figure 5.8. The yellow square in the first HR shows a  $20 \times 20$ px surface, which is then enlarged for better

Overall	PSNR (in dB)	SSIM	Accuracy	Precision	Recall	IoU
HR	100	1	0.7492	0.4779	0.3532	0.2334
Bicubic	37.4074	0.9257	0.7264	0.4336	0.3128	0.1784
SRCNN	37.2652	0.9274	0.7236	0.4334	0.3083	0.1738
VDSR	37.7672	<u>0.9299</u>	0.7273	0.4384	0.3157	0.1816
SRResNet	<b>37.8682</b>	<b>0.9313</b>	0.7342	0.4497	0.3277	0.1959
SRGAN	13.0281	0.0984	0.7279	0.0935	0.0126	0.0012
ESRGAN <sub>PSNR</sub>	37.0956	0.9188	<u>0.7386</u>	<b>0.4632</b>	<u>0.3364</u>	<u>0.2085</u>
ESRGAN <sub>Texture</sub>	36.9590	0.9165	<b>0.7415</b>	<u>0.4502</u>	<b>0.3413</b>	<b>0.2114</b>
ESRGAN	34.6965	0.8868	0.7037	0.4212	0.2068	0.1175
EESRGAN	<u>37.7794</u>	0.9298	0.7323	0.4491	0.3253	0.1933

Table 5.3: Overall results for upscaling factor of 2 by weighted averaging over all scenes.

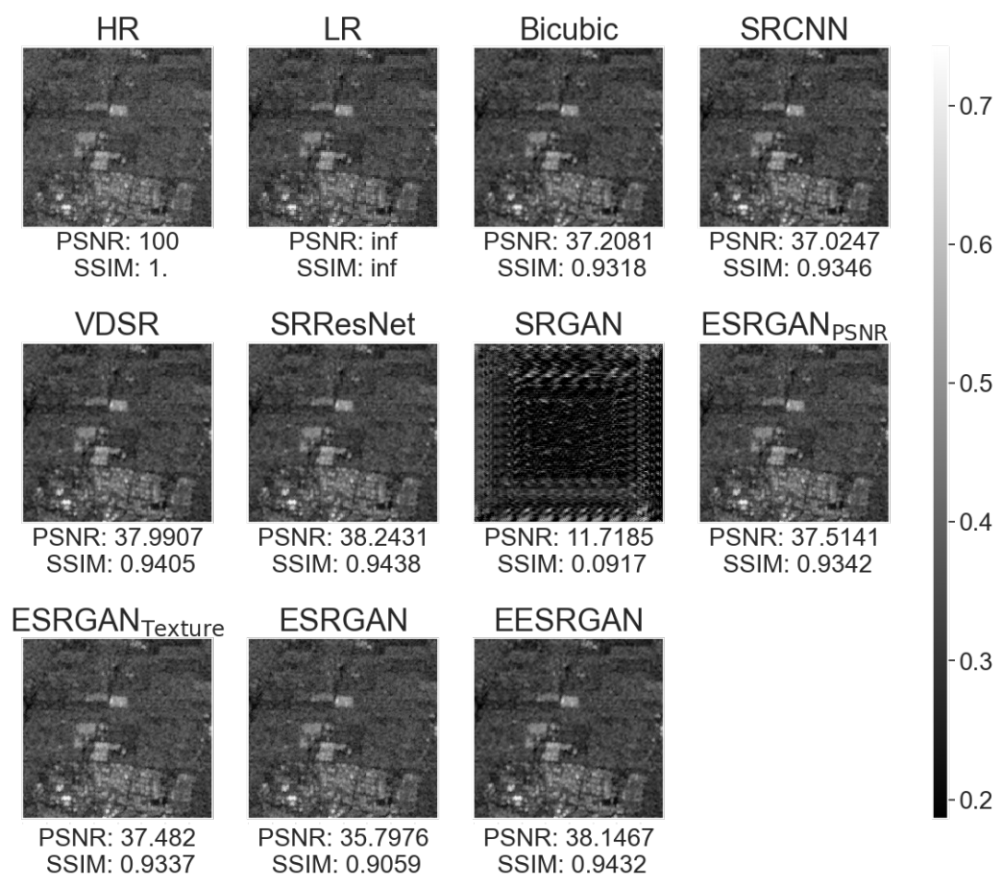


Figure 5.7: SR by a factor of 2 for the patch 111 of the March scene

visual observation. This zoomed area in the March is of importance, as it is where very high backscatter values appear. The reconstruction of the high backscatter area as well



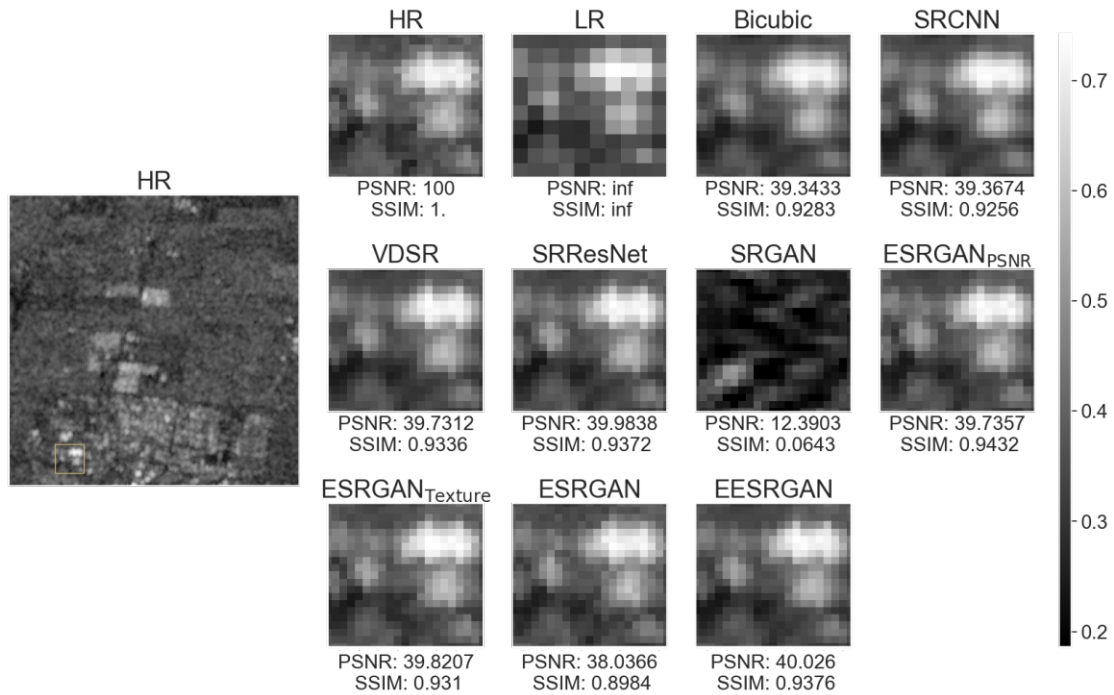


Figure 5.8: SR by a factor of 2 for the patch 111 of the March scene, zoomed in

as its surroundings resemble the original image. From this it follows that high backscatter is not an issue for the SR models.

Lower and medium backscatter surfaces can be found in Figure 5.9. The figure depicts a vessel (medium backscatter) on permanent water bodies (low backscatter) next to an unknown open forest (medium backscatter). Since the DL reconstructions are very similar to the ground truth for all three types of surfaces found on the image, it can be concluded that medium and low backscatter are also not an issue for SR x2.

At a more granular level, both close-up Figures 5.8 and 5.9 show that the SR networks slightly reduce speckle from the HR images. Furthermore, the more complex models (SRResNet and the ESRGAN variations) appear perceptually less blurry than the models using the bicubically upscaled images as input.

Evaluating the models based on the March segmentation metrics, no model outperforms the others significantly. However, ESRGAN<sub>PSNR</sub> (best) and ESRGAN<sub>Texture</sub> (second best) stand out, as they are slightly better than the rest in the given categories. It is important to note, that the models with the highest image quality metrics do not also have the highest segmentation metrics. This result is related to the conclusion of Ledig et al. [89], where PSNR and SSIM fail to appraise image quality with respect to the human visual system.

Comparing the results of this experiment with the work of Wang et al. [91] (as it compares

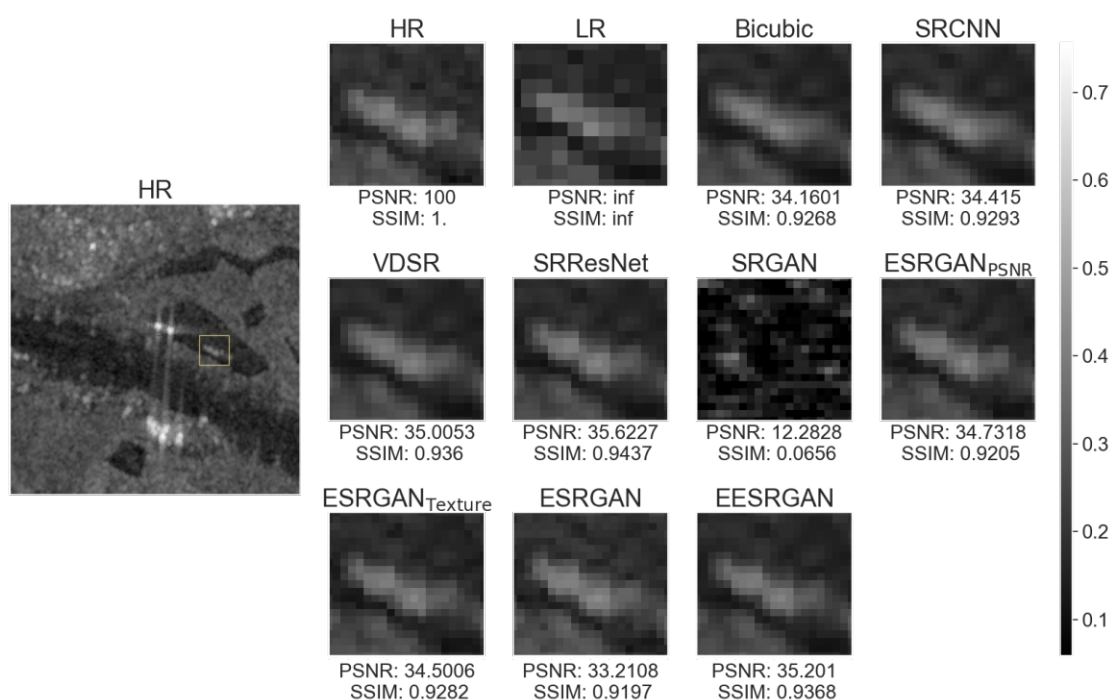


Figure 5.9: SR by a factor of 2 for the patch 991 of the March scene, zoomed in

Bicubic, SRCNN, SRGAN, and ESRGAN for RGB images), one can see that it is not unusual that the bicubic interpolation and the simpler CNNs (i.e. SRCNN, VDSR) score higher PSNR than the more complex CNN models (i.e. SRGAN and ESRGAN). However, both the experiments of this work and the work of Wang et al. showcase perceptually less appealing images (i.e. more blurry or noisy) generated by the Bicubic and the simpler CNNs.

Figure 5.10 visualizes, how the segmentation model works for different SR data. It depicts a river crossing an area containing primary the land covers *Urban*, *Cropland*, and *Open forest unknown*. The metrics below each segmentation map are calculated for the individual patch. The figure underlines the results depicted in Table 5.2, i.e. the segmentation maps are marginally worse when using SR as opposed to HR images.

Due to the fact that the EO model achieves segmentation metrics very close to the March baseline, it can be concluded that SR is viable for upscaling SAR images by a factor of 2. This was also deduced by visually inspecting the generated patches.

While the main hypothesis for SR x2 is already verified, an additional experiment is conducted to study the effects of training the EO model with the SR data. The models evaluated for this part of the experiment are narrowed down to the four best models based on the classification metrics, i.e. being best on any of the metrics. The selected models are Bicubic, ESRGAN<sub>PSNR</sub>, ESRGAN<sub>Texture</sub>, and ESRGAN. Bicubic and ESRGAN are not part of the best models for SR by a factor of 2, however, they are part of the best

## 5. EXPERIMENTS

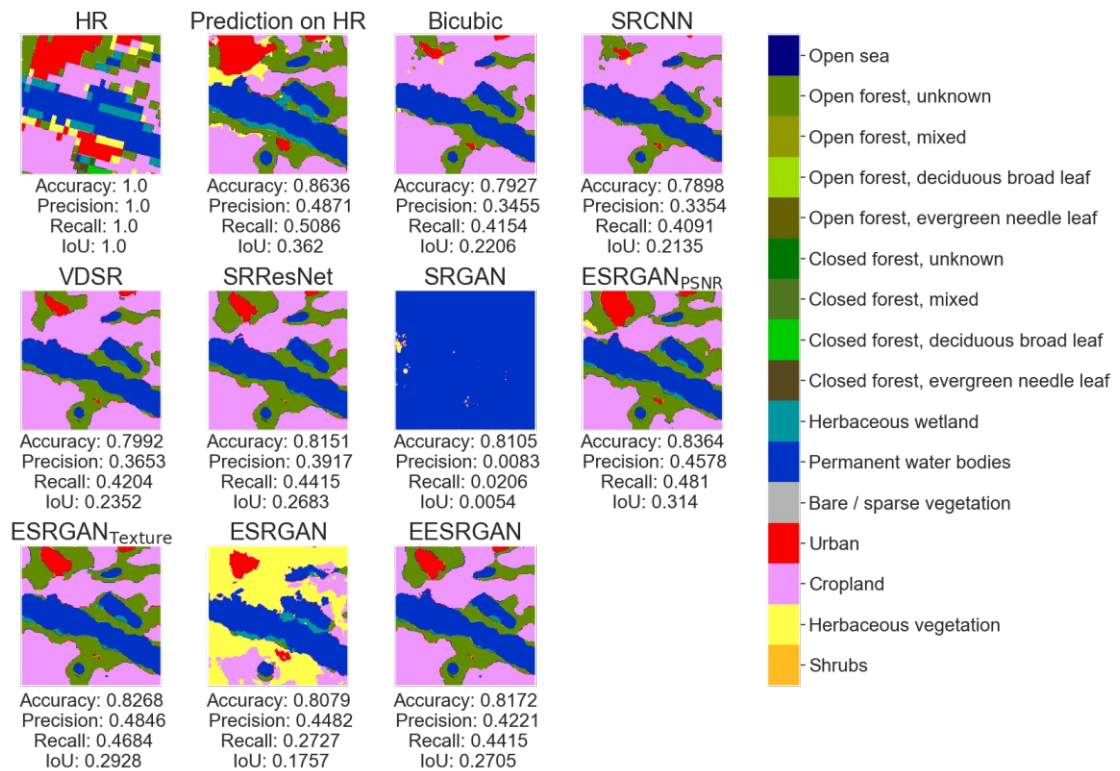


Figure 5.10: Segmentation on the SR images scaled by a factor of 2 for the patch 991 of the March scene

models for SR by a factor 4, as will be shown later.

The results of training the pixel-wise image segmentation model using the SR data can be seen in Table 5.4. In all scenes and all metrics (besides accuracy in November and Mountains), the segmentation models trained on SR are better than the EO model trained on HR. Despite outperforming the model based on HR data, the improvement is not significant. It can be argued that the marginal superiority is due to the fact that the SR images are slightly less noisy (less speckle). Moreover, the better performance compared to the March baseline, verifies that SR x2 is suitable for SAR C-band images when training the EO model on the SR images.



March scene	Accuracy	Precision	Recall	IoU
HR	0.8158	0.5267	0.5814	0.3863
Bicubic	0.8181	<b>0.5383</b>	0.5851	0.3943
ESRGAN <sub>PSNR</sub>	<b>0.8231</b>	<u>0.5339</u>	<b>0.5905</b>	<b>0.4041</b>
ESRGAN <sub>Texture</sub>	0.8216	0.5308	0.5874	0.3960
ESRGAN	<u>0.8230</u>	0.5205	<u>0.5893</u>	<u>0.4023</u>
November scene				
HR	0.8181	0.5191	0.5550	0.3810
Bicubic	0.8085	0.5160	0.5393	0.3690
ESRGAN <sub>PSNR</sub>	<u>0.8144</u>	<u>0.5305</u>	<u>0.5516</u>	<u>0.3836</u>
ESRGAN <sub>Texture</sub>	0.8132	<b>0.5364</b>	0.5428	0.3737
ESRGAN	<b>0.8165</b>	0.5268	<b>0.5585</b>	<b>0.3884</b>
Mountains scene				
HR	0.6558	0.4209	0.0752	0.0317
Bicubic	0.6501	<b>0.4759</b>	<b>0.0754</b>	<u>0.0268</u>
ESRGAN <sub>PSNR</sub>	<b>0.6545</b>	0.3977	<u>0.0741</u>	<b>0.0358</b>
ESRGAN <sub>Texture</sub>	<u>0.6502</u>	0.4087	0.0674	0.0256
ESRGAN	0.6489	<u>0.4284</u>	0.0636	0.0248
Overall				
HR	0.7492	0.4779	0.3532	0.2334
Bicubic	0.7419	<b>0.5001</b>	0.3453	0.2254
ESRGAN <sub>PSNR</sub>	<b>0.7471</b>	0.4744	<b>0.3515</b>	<b>0.2374</b>
ESRGAN <sub>Texture</sub>	0.7446	0.4820	0.3438	0.2275
ESRGAN	<u>0.7458</u>	<u>0.4848</u>	<u>0.3506</u>	<u>0.2352</u>

Table 5.4: Segmentation metrics when the segmentation model is trained on the SR data with an upscaling factor of 2.

The performance improvement between the overall results in Table 5.3 and Table 5.4 is depicted in Figure 5.11. ESRGAN<sub>PSNR</sub> and ESRGAN<sub>Texture</sub> improve less when compared to Bicubic and ESRGAN. This phenomenon is due to the fact that ESRGAN<sub>PSNR</sub> and ESRGAN<sub>Texture</sub> have overall best and second best segmentation results when using the EO model based on the HR data.

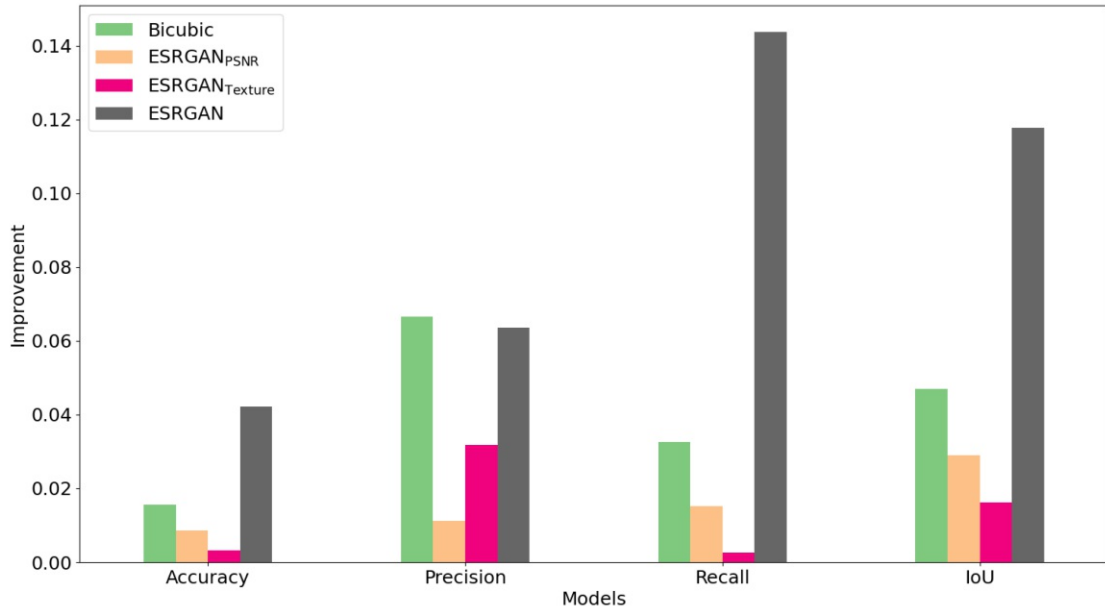


Figure 5.11: Performance improvement when the segmentation model is trained on the SR data with an upscaling factor of 2 based on the overall performance.

To further differentiate between the models used for SR, the respective parameters and run times are depicted in Table 5.5. The results in comparison to the overall IoU metrics are illustrated in Figure 5.12. It can be observed that there is a strong correlation between the prediction time and the number of model parameters. Simultaneously, the more complex models do not necessarily lead to better IoU.

Model	#Parameters	Training Time (in h)	Prediction Time (in s)
Bicubic	0	0	0.0005
SRCNN	39,001	2.37	0.0017
VDSR	664,704	2.8	0.0036
SRResNet	1,381,011	8.47	0.0129
SRGAN	16,903,260	15.01	0.01
ESRGAN <sub>PSNR</sub>	11,807,425	35.54	0.0566
ESRGAN <sub>Texture</sub>	11,807,425	38.73	0.0498
ESRGAN	27,329,674	42.12	0.0537
EESRGAN	32,883,787	61.01	0.4287

Table 5.5: Parameters and run times for the x2 SR models.

The prediction time for a single image patch is a multiple more when using DL models compared to bicubic interpolation. The time for a whole tile is for the comparatively expensive models (on the basis of ESRGAN) yet only 30 seconds. Nevertheless, depending

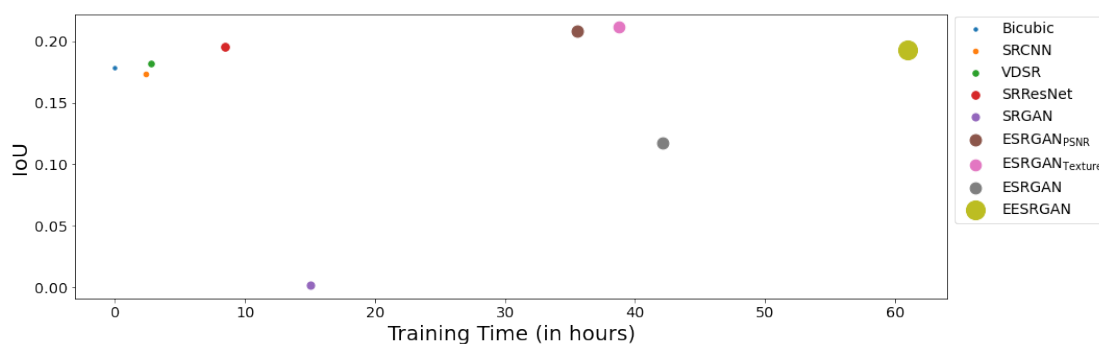


Figure 5.12: Precision and training time is depicted for each SR x2 model. Size of the markers depends on the corresponding prediction time.

on the application, this can be significant or of no importance.

Training is mandatory for the DL models compared to the bicubic interpolation. However, this adds the opportunity for the DL networks to further enhance image quality. Furthermore, training time can be reduced by adjusting the hyperparameters, e.g. learning rate and epochs.

### Sub-hypothesis 1: SR networks with an upscaling factor of 2 are able to handle unseen temporal conditions

Observing Table 5.2, it can be noted that the metrics are slightly worse on the November scene, in comparison to the March scene. However, this statement is valid for the results on both the HR and SR data. The worse performance on HR data indicates that the Mountains tile, as expected, is more challenging for the EO model than the others.

Figure 5.13 contains the same area from two different points in time (March and November). It compares the March results of Figure 5.7 with the corresponding November image reconstructions. The four best models (Bicubic, ESRGAN<sub>PSNR</sub>, ESRGAN<sub>Texture</sub>, and ESRGAN) are selected for comparison. It can be observed that the DL approaches restore the high-backscatter urban area more precisely, due to the fact that the quantitative measures are better and the image appears less blurry. Similar conclusion can be made for lower backscatter crop fields and herbaceous vegetation areas of the same patch, as seen in Figure 5.14.

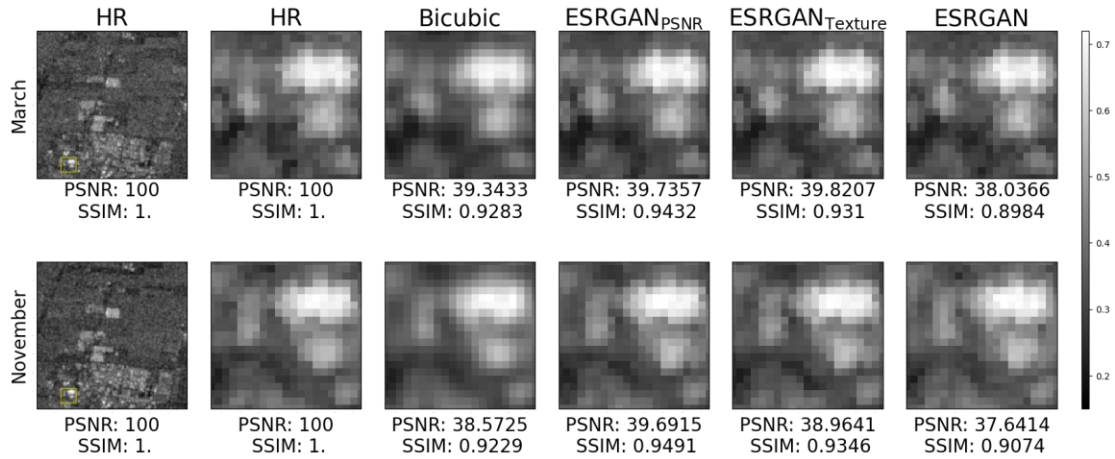


Figure 5.13: SR by a factor of 2 for the patch 111 of the March and November scenes. Urban area is magnified

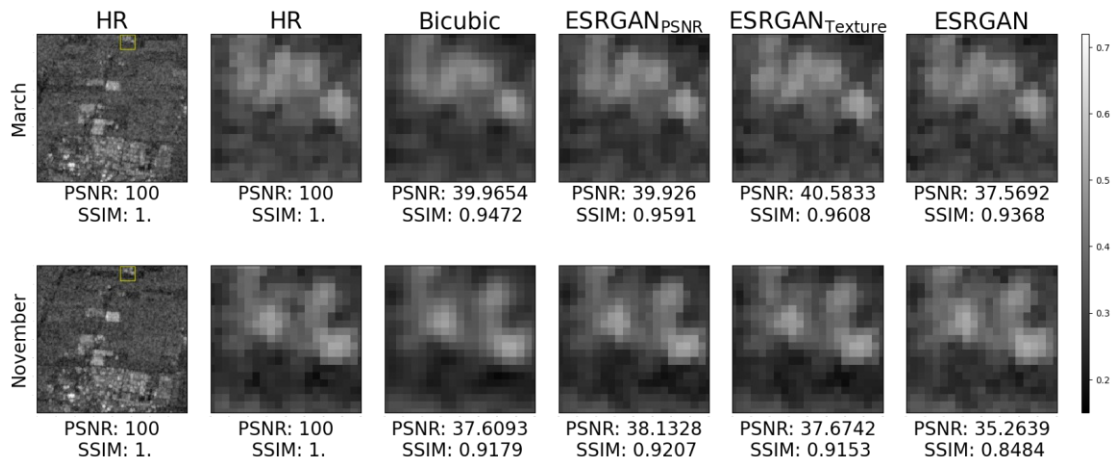


Figure 5.14: SR by a factor of 2 for the patch 111 of the March and November scenes. Crop field and herbaceous vegetation area is magnified

A direct comparison between the March and November segmentation maps of a given patch is observable in Figure 5.15. From this figure, it can be deduced that the EO model segments the March and November SR images similarly.

It is concluded that SR with an upscaling factor of 2 can handle unseen temporal conditions as a direct consequence of the proximity between the November segmentation metrics and the November baseline. This deduction is also made through the observation of the generated SR images and the corresponding segmentation maps.

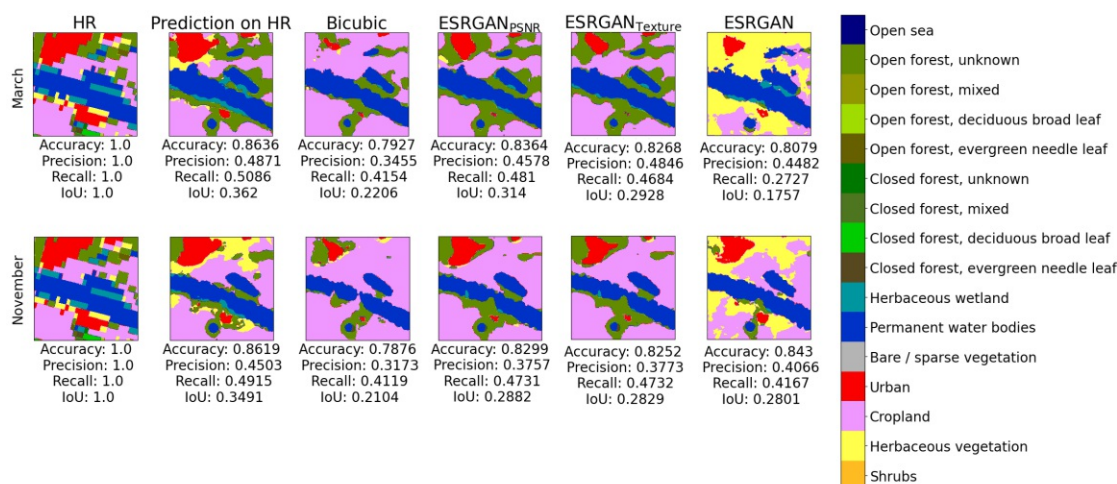


Figure 5.15: Comparison between the segmentation masks for SR x2 on patch 991 of the March and November scenes

### Sub-hypothesis 2: SR networks with an upscaling factor of 2 are able to handle unseen spatial conditions

Table 5.2 shows that there is a significant drop in the segmentation metrics when comparing the Mountains and March scenes. Specifically, there is a considerable difference in the recall and IoU metrics. This is true for both the EO model evaluated on HR and SR data. This indicates that the EO model has difficulties handling unseen spatial conditions.

Figure 5.16 illustrates the SR images and the corresponding segmentation maps of the same area. It can be seen, that both image and segmentation metrics are closer to the results of the March scene than to the Mountains scene.

Figure 5.17 shows the surfaces where the EO model has difficulties in segmenting. The figure represents a similar composition of land covers to Figure 5.16, i.e. primary a mix of urban, crop field, and forests. The SR images showcase no sudden drop in quality. However, the segmentation quality is significantly worse. The difference between the selected areas is that patch 800 (worse performance) is located in a mountainous area - whereas patch 2007 (better performance) is located in the valley. It follows that not specifically the type of land cover is the issue for the EO model, but that the topological relief influences the quality of the segmentation maps. This corresponds to Song et al. [196], who conclude that the terrain relief (e.g. elevation, slope) must be considered *before* segmentation.

Considering that all four segmentation metrics on the Mountains scenes are better compared to the Mountains baseline (Table 5.2), it follows that SR is suitable for unseen spatial conditions.

## 5. EXPERIMENTS

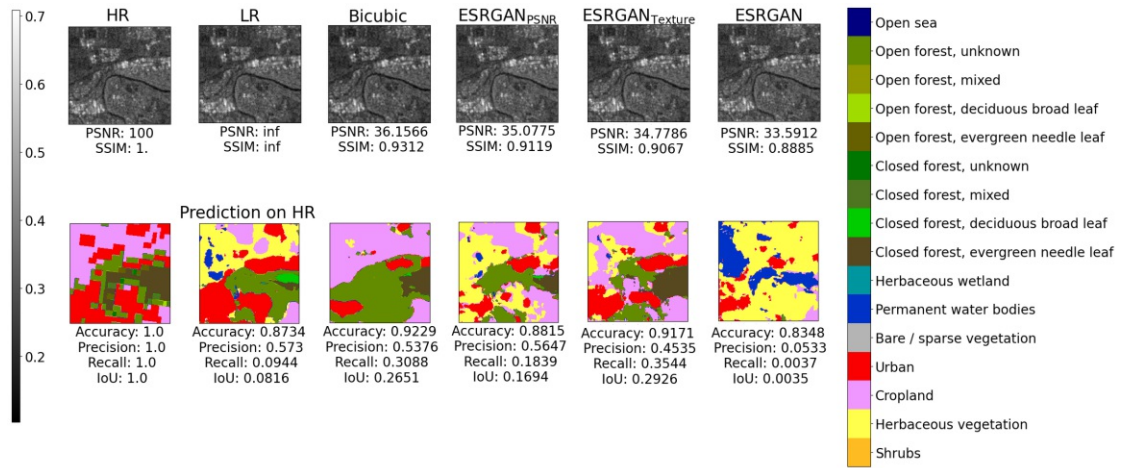


Figure 5.16: SR and corresponding segmentation map with an upscaling factor of 2 for the patch 2007 of the Mountains scene

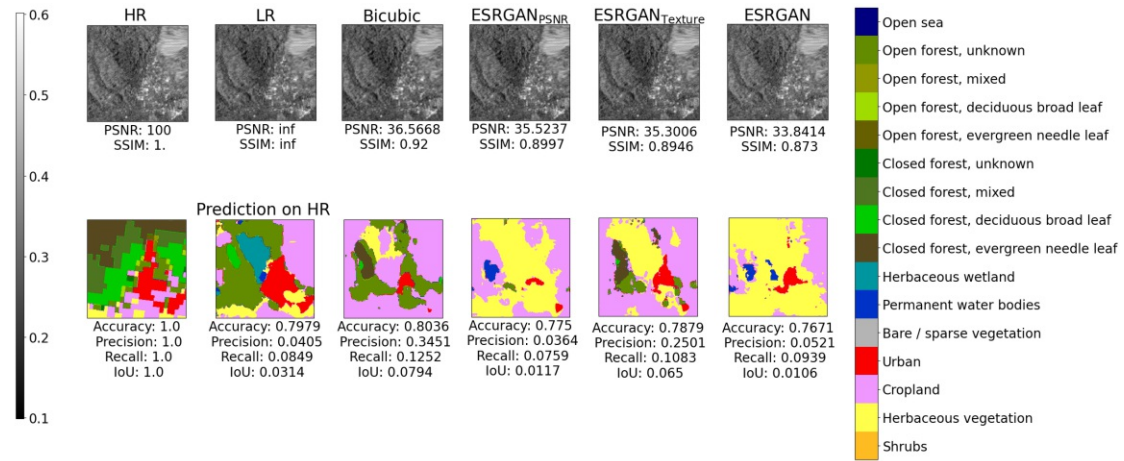


Figure 5.17: SR and corresponding segmentation map with an upscaling factor of 2 for the patch 800 of the Mountains scene

### Sub-hypothesis 3: GAN improves the SR networks with an upscaling factor of 2

The only model which performs worse (in all metrics besides accuracy) compared to its peers is SRGAN. As noted in Section 4.7.1, it is easier to get high accuracy compared to the other metrics. To explain the worse performance of the SRGAN, it can be argued that the model converged to a bad local minimum. Since the networks weights are initialized with the SRResNet structure, which is in terms of PSNR and SSIM the best performing model for two out of the three scenes, the fault is not in the generator architecture. However, the potential issues are twofold. First, the content loss (VGG loss) substitutes



the pixel loss. SRGAN is the only CNN that has no pixel loss. Second, it was observed in Figure 3.11 that the feature maps after activation of the 56<sup>th</sup> channel contain limited information.

EESRGAN is a GAN-based model and achieves second best PSNR and SSIM for the March and November scenes. ESRGAN is also a GAN-based model and scores best and second best for multiple metrics in the additional experiment (EO training on SR data). This indicates that GAN training can occasionally improve the results.

#### 5.3.4 SR x4

This section includes the quantitative and qualitative results of the SR models when upscaling SAR images by a factor of 4. The section is divided into four subsections, one for each hypothesis of this thesis.

##### **Main hypothesis: SR with an upscaling factor of 4 is suitable for SAR C-band images**

The evaluations of SR by a factor of 4 are depicted in Table 5.6. In terms of PSNR and SSIM, VDSR achieves the best results for all scenes with a slight lead over EESRGAN and SRCNN. PSNR and SSIM are significantly lower compared to the experiments with upscaling by a factor of 2. This confirms that SR by higher upscaling factors is more complex than by lower upscaling factors.

Overall, the results in Table 5.2 (x2) are significantly better than the results of Table 5.6 (x4). This demonstrates not only that SR x4 is more difficult than SR x2, but also that classification works better on the more precisely reconstructed SR data.

In contrast to x2 SR, PSNR and SSIM results for x4 SR are comparable across the different scenes. It differs from model to model if the best results are achieved on the March, November or Mountains acquisitions. This demonstrates that the SR x4 networks produce images in the same quality independently of seen or unseen temporal and spatial conditions, when not considering the segmentation task. Nevertheless, it can be argued that the distinction between the scenes is lower due to the generally lower metrics.

## 5. EXPERIMENTS

March scene	PSNR (in dB)	SSIM	Accuracy	Precision	Recall	IoU
HR	100	1	0.8158	0.5267	0.5814	0.3863
Bicubic	30.7499	0.7116	0.6963	0.3129	<u>0.4016</u>	<u>0.1729</u>
SRCNN	31.2726	0.7179	0.6847	0.2314	0.3882	0.1545
VDSR	<b>31.3594</b>	<b>0.7257</b>	0.6854	0.2536	0.3890	0.1553
SRResNet	31.0854	0.7191	0.6872	0.3144	0.3909	0.1578
SRGAN	22.7373	0.3527	0.6762	0.2294	0.1752	0.0919
ESRGAN <sub>PSNR</sub>	30.5179	0.6965	<u>0.6963</u>	<u>0.3445</u>	0.3998	0.1710
ESRGAN <sub>Texture</sub>	30.6791	0.6964	0.6901	0.3416	0.3951	0.1623
ESRGAN	28.2876	0.6146	<b>0.7142</b>	<b>0.3789</b>	<b>0.4042</b>	<b>0.186</b>
EESRGAN	<u>31.3329</u>	<u>0.7249</u>	0.6862	0.3015	0.3900	0.1559
November scene						
HR	100	1	0.8181	0.5267	0.5814	0.3863
Bicubic	30.6016	0.7102	0.6976	<u>0.3598</u>	<b>0.3913</b>	<b>0.1681</b>
SRCNN	31.1090	0.7163	0.6842	0.3147	0.3693	0.1452
VDSR	<b>31.2011</b>	<b>0.7245</b>	0.6845	0.3208	0.3692	0.1457
SRResNet	30.9850	0.7192	0.6861	0.3350	0.3700	0.1478
SRGAN	22.2488	0.3470	0.6847	0.2203	0.1749	0.0920
ESRGAN <sub>PSNR</sub>	30.3636	0.6946	<u>0.6963</u>	0.3596	0.3820	<u>0.1637</u>
ESRGAN <sub>Texture</sub>	30.5739	0.6970	0.6950	<b>0.3705</b>	<u>0.3855</u>	0.1626
ESRGAN	28.4939	0.6189	<b>0.7105</b>	0.3400	0.3558	0.1640
EESRGAN	<u>31.1791</u>	<u>0.7236</u>	0.6832	0.3441	0.362	0.1423
Mountains scene						
HR	100	1	0.6558	0.4209	0.0752	0.0317
Bicubic	30.9076	0.7276	<b>0.6572</b>	0.4487	0.0409	0.0051
SRCNN	<u>31.3089</u>	0.7282	0.6565	0.4377	0.0383	0.0028
VDSR	<b>31.3587</b>	<b>0.734</b>	0.6565	0.4496	0.0385	0.0030
SRResNet	31.0765	0.7242	0.6566	0.4397	0.0387	0.0033
SRGAN	21.273	0.3398	0.6483	0.2311	<b>0.0522</b>	<b>0.0129</b>
ESRGAN <sub>PSNR</sub>	30.0617	0.6818	<u>0.6571</u>	<b>0.4542</b>	0.0410	0.0055
ESRGAN <sub>Texture</sub>	30.6045	0.6998	0.6567	<u>0.4497</u>	0.0391	0.0036
ESRGAN	26.3183	0.6149	0.6504	0.4400	<u>0.0483</u>	<u>0.0102</u>
EESRGAN	31.3058	<u>0.7315</u>	0.6566	0.4742	0.0382	0.0032

Table 5.6: SR results for upscaling factor of 4 when training on the March scene.

Based on the overall results (Table 5.7), the VDSR network achieves highest PSNR and SSIM, narrowly followed by the EESRGAN. Depending on the segmentation metric, Bicubic, ESRGAN<sub>Texture</sub>, or ESRGAN achieve the best results.



Overall	PSNR (in dB)	SSIM	Accuracy	Precision	Recall	IoU
HR	100	1	0.7492	0.4779	0.3532	0.2334
Bicubic	30.7358	0.7176	<u>0.6804</u>	0.3949	<b>0.2435</b>	<u>0.0993</u>
SRCNN	31.1991	0.7213	0.6724	0.3624	0.2301	0.0854
VDSR	<b>31.2730</b>	<b>0.7285</b>	0.6726	0.3718	0.2302	0.0858
SRResNet	31.0259	0.7212	0.6736	0.3782	0.2308	0.0871
SRGAN	21.8591	0.3442	0.6688	0.2253	0.1230	0.0585
ESRGAN <sub>PSNR</sub>	30.2409	0.6892	0.6796	<u>0.3988</u>	0.2385	0.0971
ESRGAN <sub>Texture</sub>	30.5893	0.6981	0.6785	<b>0.4025</b>	<u>0.2393</u>	0.0953
ESRGAN	27.5593	0.6169	<b>0.6852</b>	0.3843	0.2281	<b>0.100</b>
EESRGAN	<u>31.2377</u>	<u>0.7269</u>	0.6720	0.3969	0.2264	0.0841

Table 5.7: Overall results for upscaling factor of 4 by weighted averaging over all scenes.

To visualize and distinguish the images on a pixel level, an area from patch 111 of the March scene is zoomed in in Figure 5.18. Following the SR metrics, VDSR offers the best image, however, ESRGAN<sub>Texture</sub> is perceived as most similar, as it offers the same structure of the highly-reflected surface.

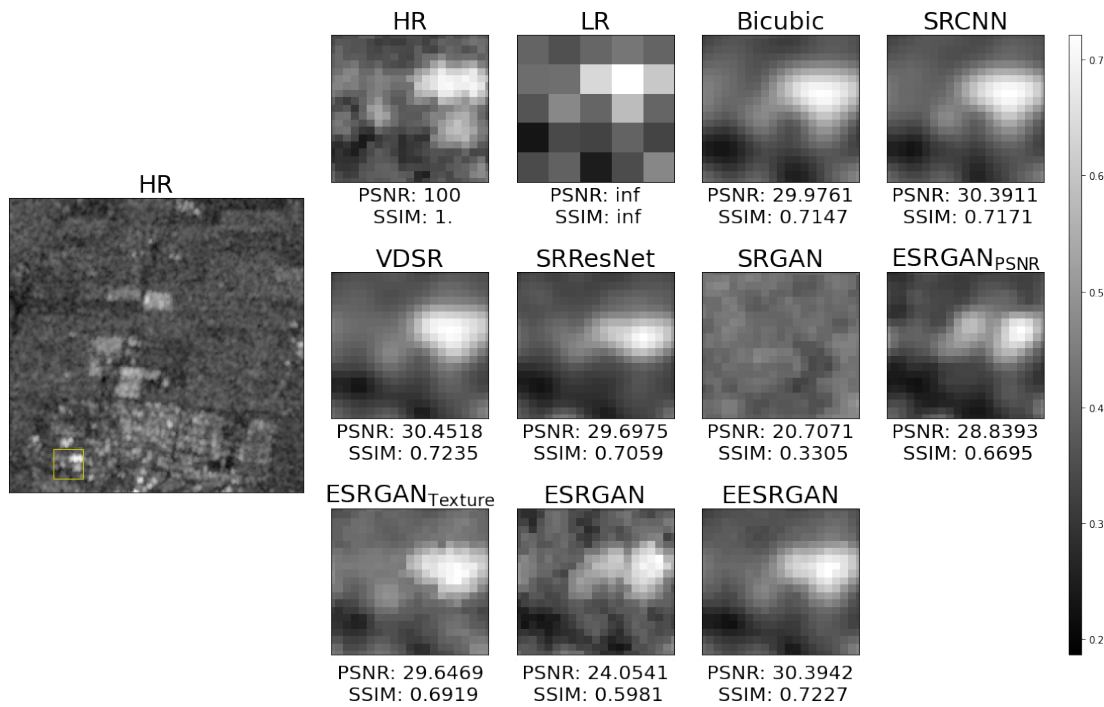


Figure 5.18: SR by a factor of 4 for the patch 111 of the March scene, zoomed in

The patch of Figure 5.9 (SR x2) showcasing lower and medium backscatter surfaces is visualized for SR x4 in Figure 5.19. Even with the information of what should be seen

on the zoomed area, the LR image on its own offers no potential of recognition. Not only is the difference between LR and HR more apparent, but the models utilizing bicubically upscaled images (Bicubic, SRCNN, VDSR) as well as SRResNet and EESRGAN are perceived as blurry compared to the HR and ESRGAN variations. On the other hand, the DL methods are able to reconstruct the three different types of surfaces. Therefore, it can be concluded that for low, medium, and high backscatter is not an issue for SR x4. However, details are lost when comparing the images to the SR x2 results.

In contrast to the SR x2 networks, which learned to reconstruct the speckle, the SR x4 networks (besides ESRGAN) reduce the speckle. This finding is particularly interesting, since on the one hand, speckle is seen as a downside (see Section 2.1). On the other hand, it is argued that this is the reason why the EO model works well on the SR x4 ESRGAN data compared to its peers. Furthermore, the EO model achieves multiple best and second best segmentation results on the SR x4 ESRGAN, and none on the SR x2 ESRGAN data.

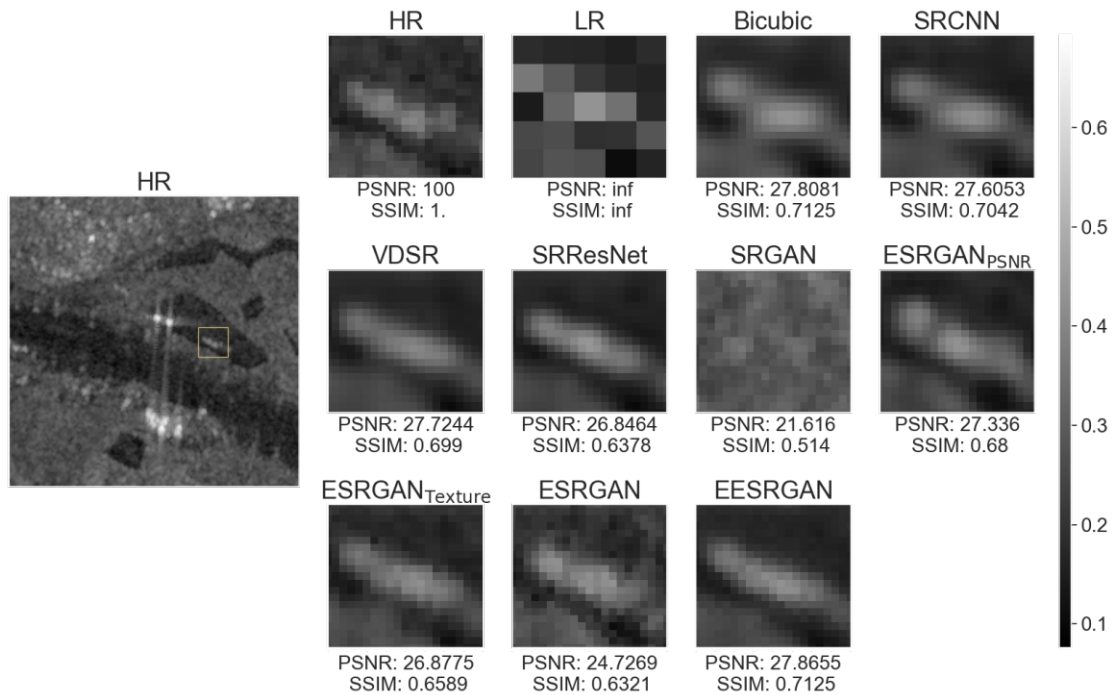


Figure 5.19: SR by a factor of 4 for the patch 991 of the March scene, zoomed in

Assessing the models by the segmentation metrics, the best models are Bicubic, ESRGAN<sub>PSNR</sub>, ESRGAN<sub>Texture</sub>, and ESRGAN being either best or second best on one of the selected metrics. The models overlap with the best models of the x2 experiments, as discussed with the exception of ESRGAN. Again, better SR metrics (PSNR and SSIM) do not lead to better classification metrics (accuracy, precision, recall, IoU).

Figure 5.20 helps assessing the quality of the segmentation masks for the different SR

inputs. The results depicted in Table 5.6 are confirmed, i.e. the EO model is worse on the SR x4 compared to the SR x2 (Figure 5.10) or the HR data. More precisely, there is a significant difference between the March baseline and the rest. Therefore, it can be concluded that SR by a factor of 4 is not suitable for upscaling SAR images on the grounds of the task-evaluation, when working with the EO model trained on HR data.

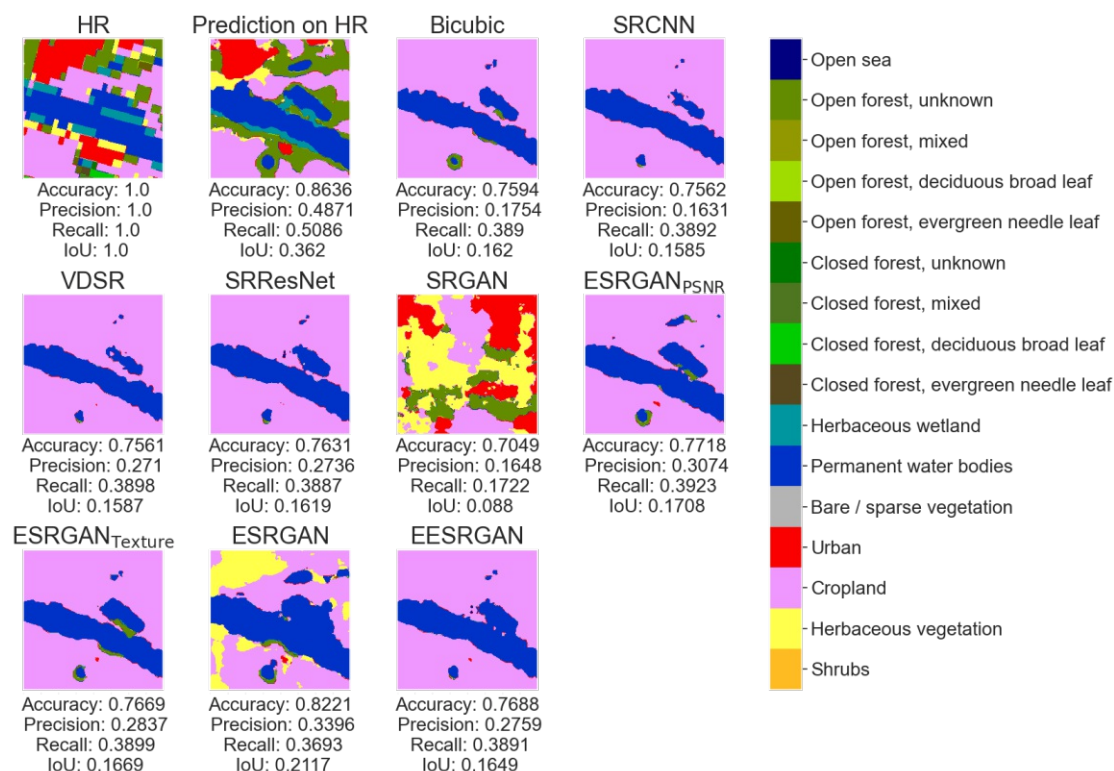


Figure 5.20: Segmentation on the SR images scaled by a factor of 4 for the patch 991 of the March scene

To evaluate the effect of the training data on the EO model, an additional experiment is conducted by training the EO model with the SR data. The four best methods (Bicubic, ESRGAN<sub>PSNR</sub>, ESRGAN<sub>Texture</sub>, and ESRGAN) are used for this additional experiment. Best is determined based on the classification metrics in both the x2 and x4 experiments. The result of this additional experiment can be observed in Table 5.8.

March scene	Accuracy	Precision	Recall	IoU
HR	0.8158	0.5267	0.5814	0.3863
Bicubic	<b>0.8122</b>	<b>0.5024</b>	<b>0.5652</b>	<b>0.374</b>
ESRGAN <sub>PSNR</sub>	0.8025	<u>0.5017</u>	<u>0.5574</u>	0.3605
ESRGAN <sub>Texture</sub>	0.8035	0.5005	0.5569	0.3600
ESRGAN	<u>0.8043</u>	0.4971	0.5572	<u>0.3617</u>
November scene				
HR	0.8181	0.5191	0.5550	0.3810
Bicubic	<b>0.8014</b>	<b>0.4920</b>	0.5072	0.3437
ESRGAN <sub>PSNR</sub>	0.7958	0.4759	0.5160	0.3419
ESRGAN <sub>Texture</sub>	0.7988	0.4822	<u>0.5167</u>	<u>0.3438</u>
ESRGAN	<u>0.8001</u>	<u>0.4897</u>	<b>0.5257</b>	<b>0.3497</b>
Mountains scene				
HR	0.6558	0.4209	0.0752	0.0317
Bicubic	0.6524	0.3902	0.0488	0.0141
ESRGAN <sub>PSNR</sub>	0.6528	<b>0.4333</b>	<u>0.0526</u>	0.0138
ESRGAN <sub>Texture</sub>	<b>0.6552</b>	<u>0.4191</u>	<b>0.0528</b>	<b>0.0180</b>
ESRGAN	<u>0.6535</u>	0.3898	0.0523	<u>0.0158</u>
Overall				
HR	0.7492	0.4779	0.3532	0.2334
Bicubic	<b>0.7388</b>	0.4494	0.3161	0.2057
ESRGAN <sub>PSNR</sub>	0.7355	<b>0.4592</b>	0.3220	0.2040
ESRGAN <sub>Texture</sub>	<u>0.7382</u>	<u>0.4564</u>	<u>0.3224</u>	<u>0.2067</u>
ESRGAN	<u>0.7382</u>	0.4478	<b>0.3269</b>	<b>0.2090</b>

Table 5.8: Segmentation metrics when the segmentation model is trained on the SR data with an upscaling factor of 4.

Figure 5.21 shows the performance improvement of the segmentation metrics when training the EO model on the SR data. It can be observed that all four models have similar overall improvements, which is due to the reason that there was no model dominating the others for all metrics. All four selected SR datasets show substantially improved results, when training with the SR data compared to the HR data. Similar as

the EO models results for SR x2, no SR x4 model is best for all scenes.

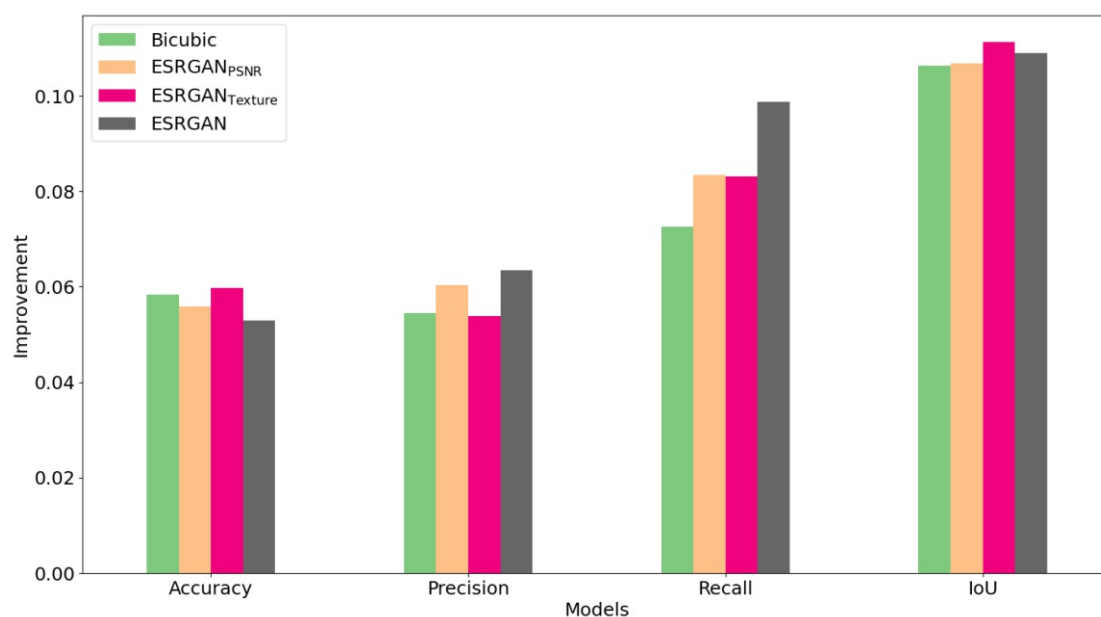


Figure 5.21: Performance improvement when the segmentation model is trained on the SR data with an upscaling factor of 4 based on the overall performance.

A comparison between the EO model trained on HR and SR can be observed in Figure 5.22. The positive effect of re-training is visible in both the segmentation maps and the associated metrics, as they approximate the results of the second column (*Prediction on HR*).

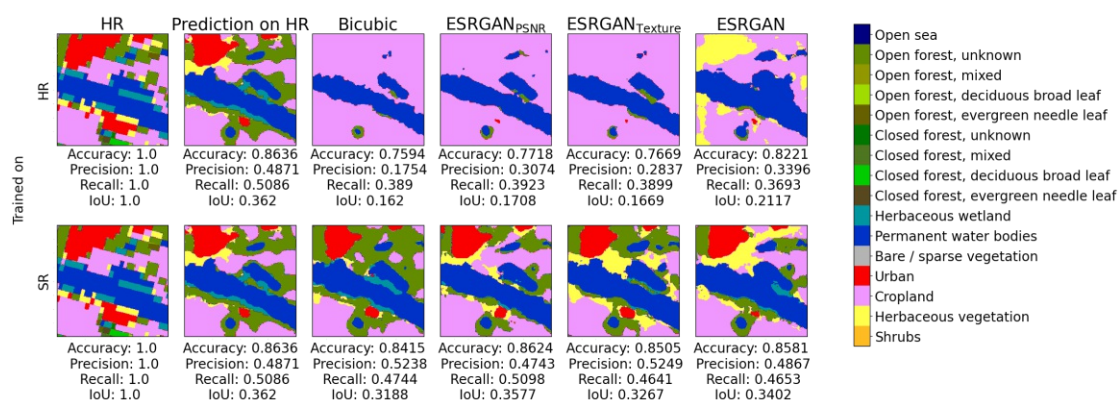


Figure 5.22: Comparison between the segmentation maps of the EO model trained on HR and SR. The two rows of the first and second column contain the same image, as the ground truth label and prediction does not change. Segmentation maps based on patch 991 the March scene using upscaling factor of 4.

The metrics of the EO model trained on SR x4 are on a par or slightly worse than the metrics of the EO model trained on HR. From this it follows that SR x4 is viable, then and only then, when the EO model is based on the SR x4 data.

Due to the fact that SR x4 is not viable without re-training the EO model, further sub-hypothesis evaluation is based on the results of the models trained with the SR images.

The number of parameters and run times are presented in Table 5.9 and illustrated in comparison to the overall IoU in Figure 5.23. Again, there is a strong correlation between the prediction time and the number of model parameters. Simultaneously, the best IoU is achieved by the models with the second most parameters (ESRGAN<sub>Texture</sub>).

Comparing the x2 results to the x4 results (Table 5.5), it is apparent that the training takes less time. This is due to the fact that the images are smaller which leads to less Input/Output wait time (reading data). Moreover, the models that do not use the bicubically interpolated images have slightly increased parameters.

Model	#Parameters	Training Time (in h)	Prediction Time (in s)
Bicubic	0	0	0.0005
SRCNN	39,001	2.13	0.0017
VDSR	664,704	2.33	0.0044
SRResNet	1,528,724	6.05	0.0125
SRGAN	17,050,973	12.83	0.01
ESRGAN <sub>PSNR</sub>	11,955,137	12.59	0.0585
ESRGAN <sub>Texture</sub>	11,955,137	15.98	0.0616
ESRGAN	27,477,386	20.34	0.0495
EESRGAN	33,031,499	26.79	0.4155

Table 5.9: Parameters and run times for the x4 SR models.

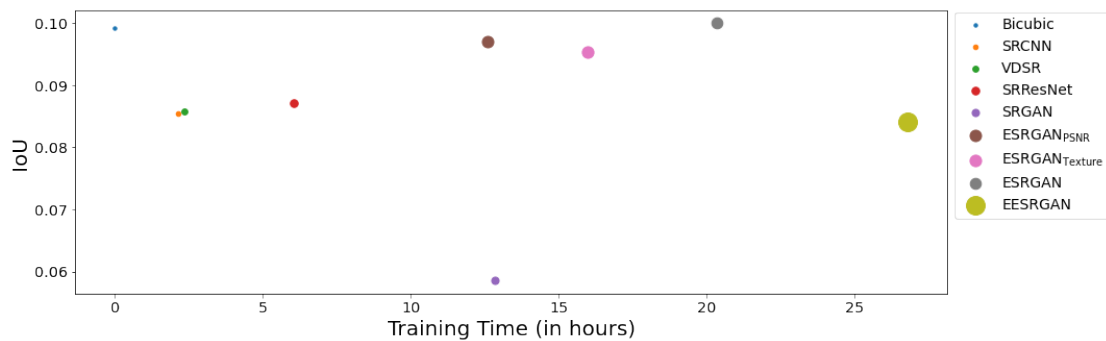


Figure 5.23: Precision and training time is depicted for each SR x4 model. Size of the markers depends on the corresponding prediction time.

The conclusions comparing the bicubic interpolation with the DL models x4 are analog



to the comparison with DL models x2. In particular, it depends on the application if the additional training and prediction time is of importance for the selection of the SR technique.

**Sub-hypothesis 1: SR networks with an upscaling factor of 4 are able to handle unseen temporal conditions**

Figures 5.24 and 5.25 compare a high backscatter and a low to medium backscatter area between March and November, respectively. The generated SR images for a given scene are approximations of each other and the ground truth. No artifacts are visible. The speckle reduction of March is also applicable in November.

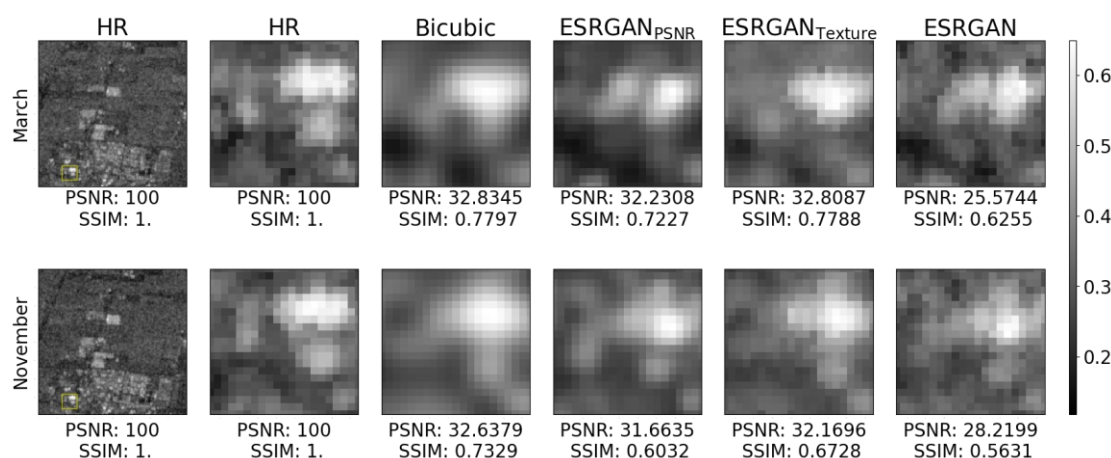


Figure 5.24: SR by a factor of 4 for the patch 111 of the March and November scenes. Urban area is magnified

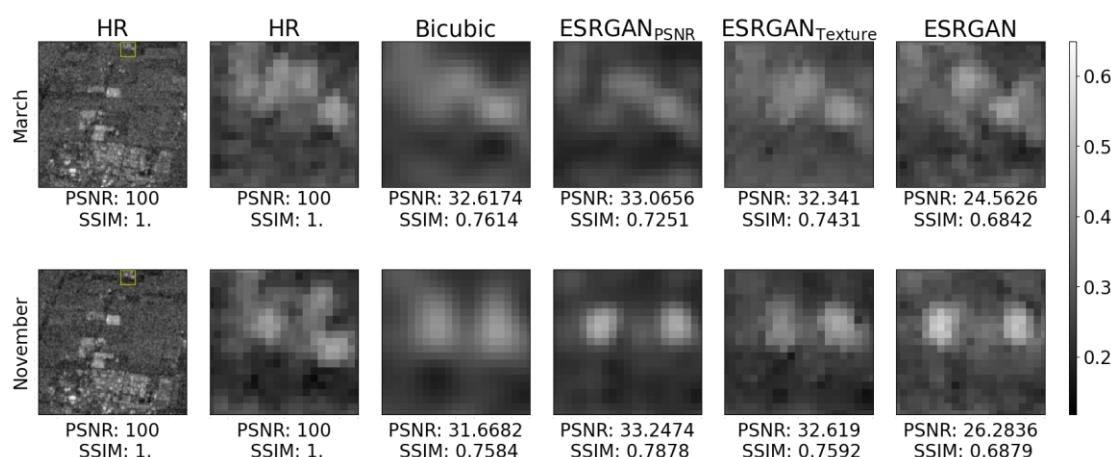


Figure 5.25: SR by a factor of 4 for the patch 111 of the March and November scenes. Crop field and herbaceous vegetation area is magnified

Figure 5.26 showcases how the segmentation models work using x4 SR data for the two different scenes of March and November. Depending on the model and the metric, the corresponding results between the scenes are slightly better or worse. Nevertheless, this example verifies the sub-hypothesis of temporal independence. Furthermore, the results of the re-trained EO model (Table 5.8) approximate the November benchmark. This indicates that x4 SR is suitable for unseen temporal conditions, then and only then, when the EO model is based on the SR x4 data.

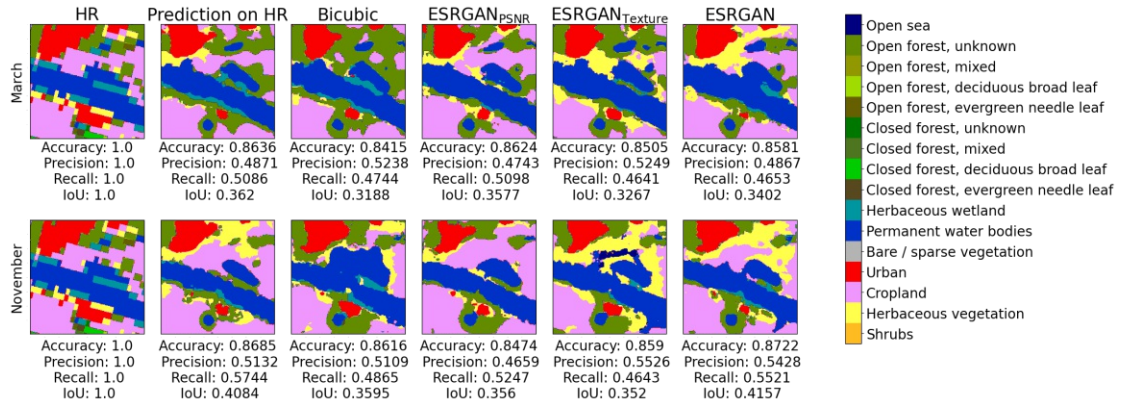


Figure 5.26: Comparison between the March and November segmentation maps of the EO model trained on SR. Segmentation maps based on patch 991 the March scene using upscaling factor of 4.

### Sub-hypothesis 2: SR networks with an upscaling factor of 4 are able to handle unseen spatial conditions

Comparing a valley area (Figure 5.27) and a mountainous area (Figure 5.28) of the Mountains scene, the conclusions from the experiments of SR x2 are replicated. In particular, the predictions on the valley are more accurate than on the mountainous area. Simultaneously, the SR images show a similar loss of quality in terms of details and blurriness. Therefore, the topological relief is more influential for the EO task than the SR task.

Considering that accuracy and precision of the Mountains scenes are on par with the Mountains baseline (Table 5.8). Simultaneously, recall and IoU are slightly worse than the baseline, it follows that SR is suitable for unseen spatial conditions, then and only then when the EO model is re-trained on the SR data.

### Sub-hypothesis 3: GAN improves the SR networks with an upscaling factor of 4

Once more, the SRGAN performs poorly compared to its peers. Nevertheless, other GAN models are successfully trained. Therefore, the explanation of this phenomenon stays the same - lacking pixel loss and wrong feature maps.



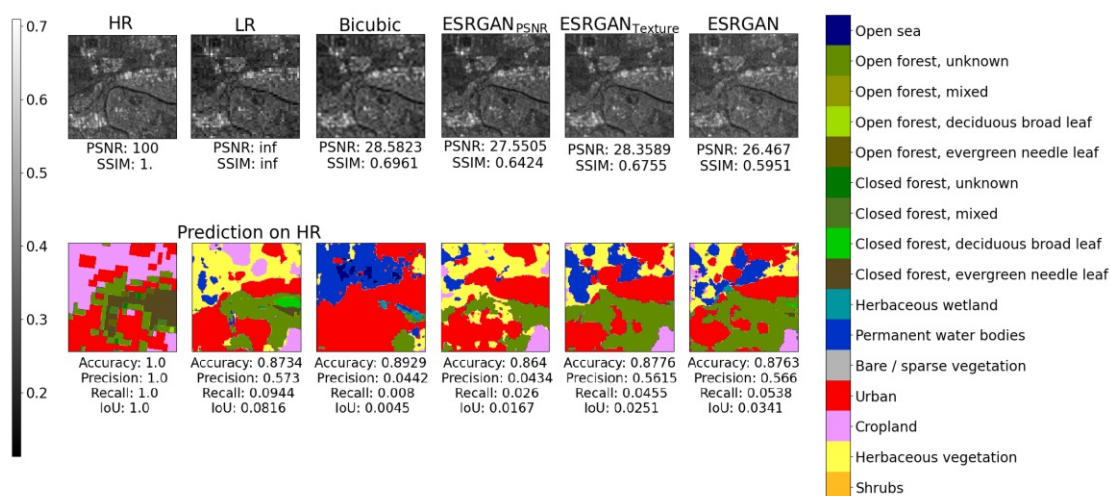


Figure 5.27: SR and corresponding segmentation map with an upscaling factor of 4 for the patch 2007 of the Mountains scene

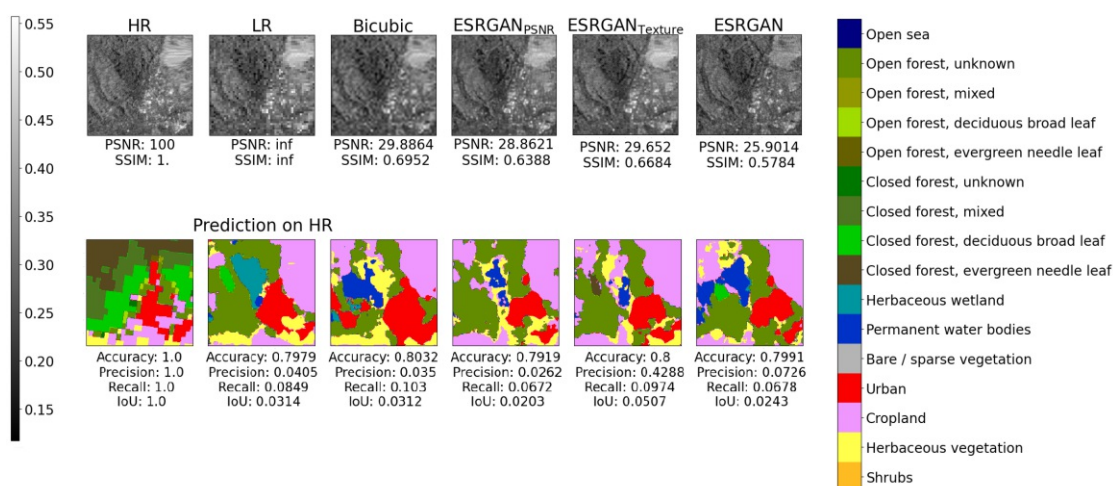


Figure 5.28: SR and corresponding segmentation map with an upscaling factor of 4 for the patch 800 of the Mountains scene

EESRGAN achieves second best PSNR and SSIM altogether. Overall, ESRGAN scores best for two out of the four segmentation metrics, respectively. When the EO model is trained on the ESRGAN data, it achieves the two best metrics and one second best. It can be argued that the ESRGAN yields the visually most pleasing and least blurry images. The deblurring is achieved by the utilization of a loss based on the feature space (VGG loss as stated in Equation 3.11) instead of the pixel space (L1 or L2 criterion). This all hints that a GAN based approach can improve the results of the given CNN.

## 5.4 Summary

It was decided that the suitability of SR will be measured based on the task of land cover classification with 16 different classes. On the one hand, PSNR and SSIM are used for quantifying the image quality of the generated images. On the other hand, accuracy, precision, recall, and IoU are used for answering the scientific questions of the thesis.

This chapter showed the experimental design and the corresponding results of x2 and x4 SR. The AISUKF model was dismissed from the experiments, as it showed unpromising results on random samples.

No model achieved simultaneously best SR and classification metrics. Overall, VDSR and SRResNet scored best in PSNR and SSIM. In contrast, Bicubic, ESRGAN<sub>PSNR</sub>, ESRGAN<sub>Texture</sub>, and ESRGAN scored best on accuracy, precision, recall, and IoU. This indicated that the given EO model performs better on the more visually pleasing data, besides Bicubic which also yielded comparable results. Furthermore, due to the fact that PSNR and SSIM fail to represent the human perception of image visual quality, it was concluded that image and segmentation quality metrics do not correlate.

The SRGAN showed no promising perceptual results nor good SR metrics. It was argued that the reason is the wrongly selected feature maps or the missing pixel loss.

GAN did not always enhance the results with focus on PSNR, SSIM, accuracy, precision, recall, or IoU. However, it was stated that the GAN in the ESRGAN yielded the perceptually most pleasing results.

SR reduced the speckle inherent in the HR, LR, and bicubically interpolated images. The noise reduction was stronger for the SR x4 in comparison to the SR x2 methods.

SR with a scaling factor of 2 has no issues handling seen or unseen temporal and spatial conditions. With focus on upscaling factor of 4, the SR quality was equally good for unseen temporal and spatial conditions, when considering both quantitative (PSNR and SSIM) and qualitative (perceptual impression) measurements. The segmentation did not achieve the desired results (similar to the HR EO model) for any temporal and spatial conditions. Hence, x4 SR is not suitable for the given EO task. However, x4 SR is viable for both unseen temporal and spatial conditions when training the segmentation model on the SR x4 data.

# CHAPTER 6

## Conclusion

This chapter summarizes the thesis and outlines the main findings. In addition, a number of recommendations for future research are given.

### 6.1 Summary

This thesis provided grounds for evaluating and comparing SR methods for SAR C-band images. A methodology was presented to evaluate the suitability of SR for earth observation tasks.

An overview of SR models was given and their differences were presented. Additionally, useful components for SR were showcased and discussed. It was observed that deeper network architectures can be difficult to train but have the potential to substantially increase the networks performance as they allow modeling mappings of high complexity.

With regards to the research question, it is concluded that SR is suitable for increasing the resolution of SAR C-band images by a factor of 2. On the other hand, SR by a factor of 4 is not suitable without additional effort. More precisely, for SR x4 to be suitable, the EO model needs to be trained with the SR data.

Furthermore, the thesis evaluated if the SR methods can handle unseen temporal and spatial conditions. It is determined that SR by a factor of 2 can handle both unseen temporal and spatial conditions, as the classification metrics for the November and Mountains scenes were on a par with the results for March scene. With focus on SR by a factor of 4, the method can handle unseen temporal and spatial conditions only with the additional effort of training the EO model on the SR data.

Another sub-hypothesis of this thesis was to evaluate the GAN framework as a vehicle to improve the results. GAN situationally improved both the classification and image quality metrics. However, it can not be stated with certainty in which cases the GAN

enhances the results. Based on perceptual sentiment, a GAN-based method (ESRGAN) yielded the most pleasing images.

Depending on the scaling factor and the metric of interest (i.e. precision or recall), applying computationally expensive deep learning methods is adequate in comparison to the trivial bicubic upscaling method. Furthermore, the adequateness of using computationally more complex methods is dependent on if unseen temporal or spatial conditions are significant for the EO task. Additionally, it needs to be taken into consideration if the EO model will be re-trained based on the SR data or not.

The tasks of SR and pixel segmentation are harder when the initial images are with lower GSD. This fact follows as the results of both image quality and classification metrics declined when comparing the x2 and x4 experimental results.

Even though PSNR and SSIM are used as an approximation to human perception of reconstruction quality, the results have shown that perceptual loss is a promising alternative to pixel-based loss functions (MSE, MAE). Additionally, it was showcased how to adapt the VGG network to function with SAR images. It was also shown that the pixel-based loss is a necessary part for SAR SR, as indicated by the SRGAN results.

Moreover, this study examined the impact of SR to the inherent speckle of the SAR C-band images. The findings clearly indicate that SR is able to reduce the noise present in the HR, LR, and bicubically upscaled images. Together with the fact that the EO models trained on the SR images occasionally outperformed the March, November, and Mountains baselines, it follows that SR helps improve earth science models.

Overall, SR cuts down processing and storage capacity through its ability to decompress low-resolution images. For example, splitting the November scene into HR ( $200 \times 200\text{px}$ ), LR for 2x SR ( $100 \times 100\text{px}$ ), and LR for 4x SR ( $50 \times 50\text{px}$ ) patches requires 381, 96, and 24 megabyte of storage space, respectively. This is approximately a reduction by a factor of 4 and 16, respectively for the x2 and x4 LR patches. This compression strongly correlates to the number of pixels which is reduced. Hence, using SR greatly reduce the disk space needed, which is especially relevant to the TU Wien as part of the EODC, since they store petabytes of satellite data.

SR is a well researched discipline in DL. It was shown that both CNN and GAN architectures are able to work for the spatial data at hand. The suitability of SAR SR lays a strong basis for new or improved EO models.

### 6.2 Future Work

Training on more images could improve the results. Hence, the impact of the dataset size should be inspected. For training both the SR and segmentation networks, 3486 images were used. This equals to  $3486 \times 200 \times 200 \approx 139\text{M}$  pixels or about 0.5 GB of data. Simultaneously, there are terabytes of SAR C-band data recorded each year. In contrast, the authors of the ESRGAN use  $\sim 22.5$  billion pixels. Simultaneously, it should

not be neglected that the SAR images used in this work have only one channel, whereas the ESRGAN images have three (RGB).

Concerning the number of channels, this work adapted the three-channel VGG loss for single-channel SAR images. Using a feature loss trained on different images might be a limiting factor in terms of SR quality. Moreover, Wang et al. [91] fine-tune the VGG network for material recognition (to focus on textures rather than objects) and use this as the feature loss to improve the visual quality of the SR images. Therefore, evaluating different feature loss functions may prove to be an important area for future research.

The pixel segmentation task chosen for the assessment of the research question could be restated. A potential transformation, which would make the task more general could be to group the classes, as there were similar classes (e.g. open forest unknown, open forest mixes, open forest evergreen needle leaf, open forest deciduous broad leaf). If the reduction of the total classes is not an option for the EO task, then another approach is to use a two-step segmentation approach. For example, a first segmentation model is used to determine the group, e.g. forest or water. A second segmentation model is used to determine the specific sub-class in the group, e.g. Closed forest, evergreen needle leaf or open sea.

Further research could include an end-to-end approach. In this approach, the segmentation model (and loss) would be included in the training of the SR network. Hence, the SR network would be able to learn to generate images, which are optimal for a given EO model.

Moreover, this thesis used only VV backscatter. However, the Sentinel-1 C-band SAR instrument satellite, in its dual polarization mode, additionally provides the VH backscatter. VV and VH data matches both temporally and spatially. Thus, VH can be used as an additional input for the SR CNNs.



# List of Figures

1.1	SAR image over Carinthia, Austria. Source: TU Wien Sentinel-1 datacube	2
1.2	Same geographic area in (a) LR (40m GSD) and (b) HR (10m GSD). . .	3
2.1	Electromagnetic wave with horizontal and vertical polarization. Source: [35]	6
2.2	Image showcasing the backscattering mechanisms in urban areas: (a) Single Bounce Scattering, (b) Double Bounce Scattering, and (c) Triple Bounce Scattering. Source: [47] . . . . .	7
2.3	Synthetically speckled image from [62]: (a) noise-free reference, (b) noisy.	8
2.4	Semantic image segmentation with five classes. Source: [111] . . . . .	11
3.1	Sub-pixel convolution layer. Source: Own image based on [136] . . . . .	16
3.2	A residual block. Source: Own illustration based on [82] . . . . .	17
3.3	Dense Block. Source: Own image based on [139] . . . . .	18
3.4	Residual Dense Block. Source: Own image based on [94] . . . . .	19
3.5	Residual-in-Residual Dense Block. Source: Own image based on [91] . . .	19
3.6	SRCNN network structure. Source: Own illustration based on [78] . . . .	20
3.7	VDSR network structure. Source: Own illustration based on [79] . . . . .	20
3.8	Performance curve for residual and non-residual networks. Source: [79] . . .	21
3.9	A sample GAN used for SR. Source: Own image . . . . .	22
3.10	Generator and discriminator network architectures of the SRResNet and SRGAN models. Source: [89] . . . . .	23
3.11	Feature maps before and after activation for the 34 <sup>th</sup> and 56 <sup>th</sup> channels. .	26
3.12	Edge-enhancement network used in the EESRGAN. Source: [92] . . . . .	27
3.13	Pixel-wise image segmentation. Source: Own illustration based on the TU Wien Sentinel-1 datacube data. . . . .	28
3.14	U-Net architecture. Source: Own image based on [160] . . . . .	29
3.15	Attention U-Net architecture. Source: Own image based on [125] . . . . .	30
3.16	Attention gate structure. Example based on the skip connection from $F_3 \times H_3 \times W_3$ and gating signal from $F_4 \times H_4 \times W_4$ . Source: Own image based on [125] . . . . .	30
4.1	Tile "E045N021T1" presented in three different configurations: (a) using VH backscatter information, (b) using VV backscatter information, and (c) using (a) as the red and (b) as the green color channel. . . . .	34
		83



4.2	Extent of the study site of the March scene. . . . .	35
4.3	Extent of the study site of the Mountains scene. . . . .	35
4.4	Semantic maps used in the experiments for the tile: (a) March & November and (b) Mountains . . . . .	36
4.5	Gray value distribution of the test data of the scenes March, November, and Mountains. . . . .	37
4.6	Distribution of the classes in datasets: (a) March Train, (b) March Train Augmented. . . . .	39
4.7	Distribution of the classes in datasets: (a) March Test, (b) November Test, (c) Mountains Test. . . . .	40
4.8	TP, TN, FP, and FN in case of multi-class segmentation with respect to class b. Source: [185] . . . . .	45
5.1	Image patch 10 of the scenes March and November, along with the associated segmentation mask . . . . .	50
5.2	Image patch 99 of the scenes March and November, along with the associated segmentation mask . . . . .	51
5.3	Image patch 152 of the scenes March and November, along with the associated segmentation mask . . . . .	51
5.4	Experimental design. SR-Network and EO-Model are trained on the March scene using the HR data. The corresponding tiles, i.e. March, November, or Mountains are used when evaluating the HR and SR images . . . . .	52
5.5	Confusion matrix for the evaluation on the HR March scene. . . . .	53
5.6	Image from March tile presented in three different configurations: (a) Adaptive ISUKF upscaling, (b) Bicubic upscaling, and (c) HR . . . . .	54
5.7	SR by a factor of 2 for the patch 111 of the March scene . . . . .	57
5.8	SR by a factor of 2 for the patch 111 of the March scene, zoomed in . . .	58
5.9	SR by a factor of 2 for the patch 991 of the March scene, zoomed in . . .	59
5.10	Segmentation on the SR images scaled by a factor of 2 for the patch 991 of the March scene . . . . .	60
5.11	Performance improvement when the segmentation model is trained on the SR data with an upscaling factor of 2 based on the overall performance. . . .	62
5.12	Precision and training time is depicted for each SR x2 model. Size of the markers depends on the corresponding prediction time. . . . .	63
5.13	SR by a factor of 2 for the patch 111 of the March and November scenes. Urban area is magnified . . . . .	64
5.14	SR by a factor of 2 for the patch 111 of the March and November scenes. Crop field and herbaceous vegetation area is magnified . . . . .	64
5.15	Comparison between the segmentation masks for SR x2 on patch 991 of the March and November scenes . . . . .	65
5.16	SR and corresponding segmentation map with an upscaling factor of 2 for the patch 2007 of the Mountains scene . . . . .	66



5.17	SR and corresponding segmentation map with an upscaling factor of 2 for the patch 800 of the Mountains scene . . . . .	66
5.18	SR by a factor of 4 for the patch 111 of the March scene, zoomed in . . . .	69
5.19	SR by a factor of 4 for the patch 991 of the March scene, zoomed in . . . .	70
5.20	Segmentation on the SR images scaled by a factor of 4 for the patch 991 of the March scene . . . . .	71
5.21	Performance improvement when the segmentation model is trained on the SR data with an upscaling factor of 4 based on the overall performance. . . .	73
5.22	Comparison between the segmentation maps of the EO model trained on HR and SR. The two rows of the first and second column contain the same image, as the ground truth label and prediction does not change. Segmentation maps based on patch 991 the March scene using upscaling factor of 4. . . . .	73
5.23	Precision and training time is depicted for each SR x4 model. Size of the markers depends on the corresponding prediction time. . . . .	74
5.24	SR by a factor of 4 for the patch 111 of the March and November scenes. Urban area is magnified . . . . .	75
5.25	SR by a factor of 4 for the patch 111 of the March and November scenes. Crop field and herbaceous vegetation area is magnified . . . . .	75
5.26	Comparison between the March and November segmentation maps of the EO model trained on SR. Segmentation maps based on patch 991 the March scene using upscaling factor of 4. . . . .	76
5.27	SR and corresponding segmentation map with an upscaling factor of 4 for the patch 2007 of the Mountains scene . . . . .	77
5.28	SR and corresponding segmentation map with an upscaling factor of 4 for the patch 800 of the Mountains scene . . . . .	77
A	Extent of the study site of the November scene. . . . .	93
B	Confusion matrix for the evaluation on the HR November scene. . . . .	94
C	Confusion matrix for the evaluation on the HR Mountains scene. . . . .	94



# List of Tables

4.1	List of classes with the corresponding colors as well as count and share of pixels based on the training dataset. . . . .	38
5.1	List of classes with the corresponding segmentation metrics for the March scene based on the HR data. . . . .	53
5.2	SR results for upscaling factor of 2. All training is done based on the March scene. Evaluation is done separately for each scene. . . . .	56
5.3	Overall results for upscaling factor of 2 by weighted averaging over all scenes.	57
5.4	Segmentation metrics when the segmentation model is trained on the SR data with an upscaling factor of 2. . . . .	61
5.5	Parameters and run times for the x2 SR models. . . . .	62
5.6	SR results for upscaling factor of 4 when training on the March scene. . .	68
5.7	Overall results for upscaling factor of 4 by weighted averaging over all scenes.	69
5.8	Segmentation metrics when the segmentation model is trained on the SR data with an upscaling factor of 4. . . . .	72
5.9	Parameters and run times for the x4 SR models. . . . .	74
A	List of classes with the corresponding segmentation metrics for the November scene based on the HR data. . . . .	95
B	List of classes with the corresponding segmentation metrics for the Mountains scene based on the HR data. . . . .	95



# List of Acronyms

**AISUKF** Adaptive Importance Sampling Unscented Kalman Filter.

**BN** Batch Normalization.

**CIFAR** Canadian Institute For Advanced Research.

**CNN** Convolutional Neural Network.

**Conv** Convolution.

**DB** Dense Block.

**dB** Decibel.

**DenseNet** Dense Convolutional Network.

**DL** Deep Learning.

**EEGAN** Edge Enhanced GAN.

**EEN** Edge-Enhancement Network.

**EESRGAN** Edge-Enhanced Super-Resolution.

**EO** Earth Observation.

**EODC** Earth Observation Data Center.

**ESRGAN** Enhanced SRGAN.

**FCN** Full Convolutional Network.

**GAN** Generative Adversarial Network.

**GSD** Ground Sampling Distance.

**HH** Horizontal on transmit, Horizontal on receive.

**HR** High-Resolution.

**HV** Horizontal on transmit, Vertical on receive.

**IoU** Intersection-over-Union.

**ISR** Intermediate SR.

**LR** Low-Resolution.

**LReLU** Leaky ReLU.

**MAE** Mean Absolute Error.

**ML** Machine Learning.

**MSE** Mean Squared Error.

**NaN** Not a Number.

**NN** Neural Network.

**PolSAR** Polarimetric Synthetic Aperture Radar.

**PReLU** Parametric ReLU.

**PSNR** Peak Signal-to-Noise Ratio.

**RaGAN** Relativistic GAN.

**RB** Residual Blocks.

**RDB** Residual Dense Block.

**ReLU** Rectified Linear Units.

**RRDB** Residual-in-Residual Dense Block.

**SA** Superauflösung.

**SAR** Synthetic Aperture Radar.

**SGAN** Standard GAN.

**SISR** Single-Image Super-Resolution.

**SOTA** State-Of-The-Art.

**SR** Super-Resolution.

**SRCNN** Super-Resolution Convolutional Neural Network.

**SSIM** Structural SIMilarity.

**VDSR** Very Deep Super-Resolution.

**VH** Vertical on transmit, Horizontal on receive.

**VSC** Vienna Scientific Cluster.

**VSC3** Vienna Scientific Cluster 3.

**VV** Vertical on transmit, Vertical on receive.





# Appendix

## Figures & Tables

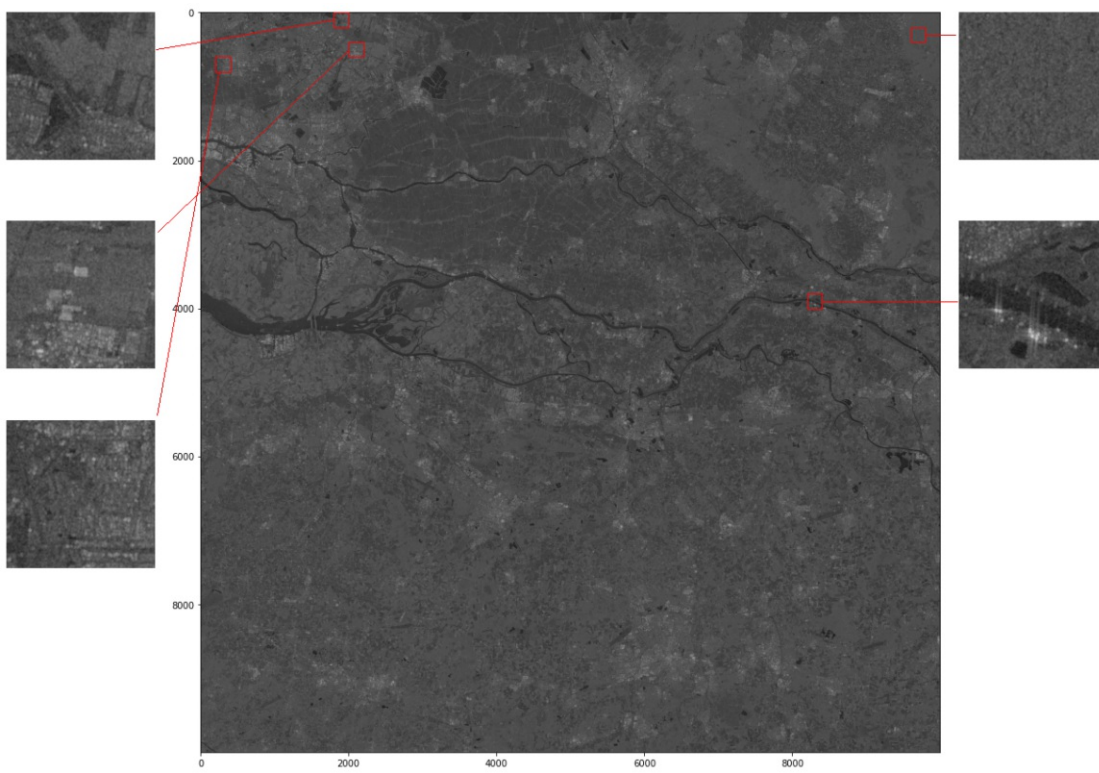


Figure A: Extent of the study site of the November scene.

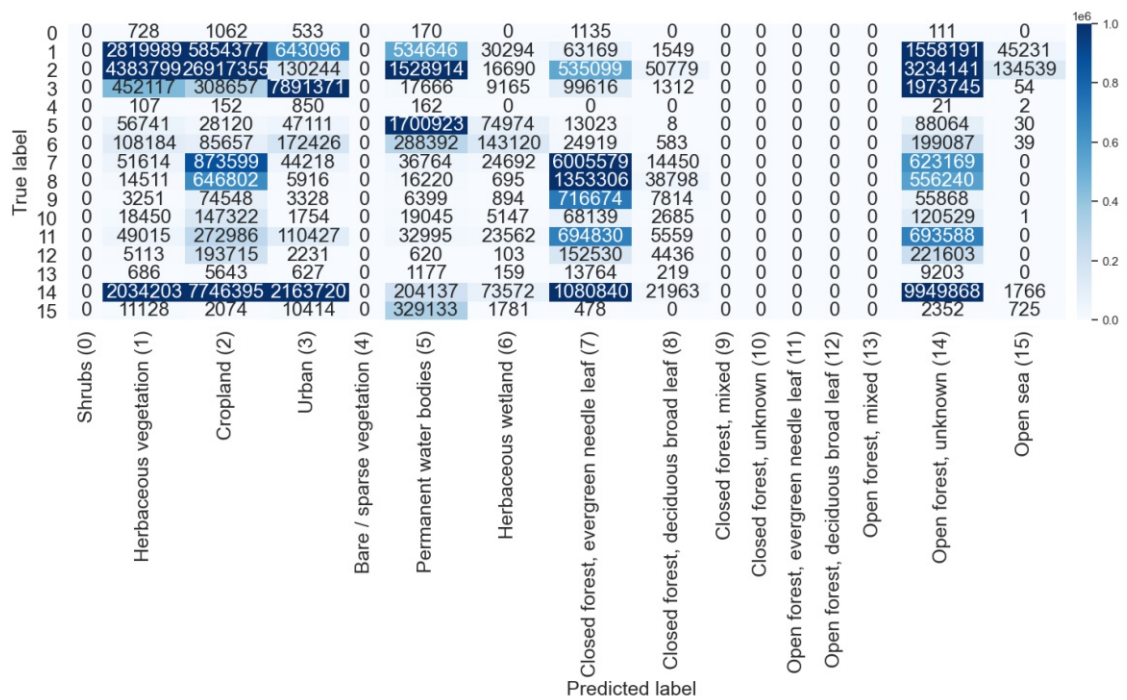


Figure B: Confusion matrix for the evaluation on the HR November scene.

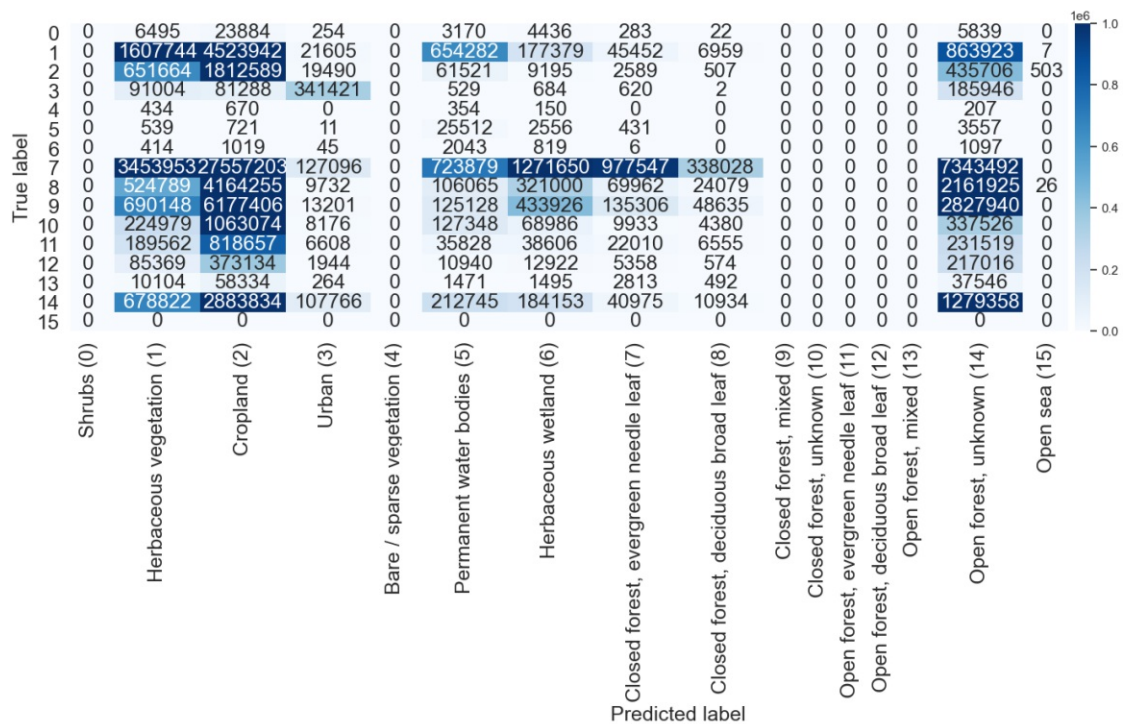


Figure C: Confusion matrix for the evaluation on the HR Mountains scene.

Class	Accuracy	Precision	Recall	IoU
Shrubs	1	NaN	0	0
Herbaceous vegetation	0.8407	0.2817	0.2441	0.1505
Cropland	0.7373	0.6237	0.7288	0.5062
Urban	0.938	0.7028	0.7338	0.56
Bare / sparse vegetation	1	NaN	0	0
Permanent water bodies	0.9667	0.3606	0.8467	0.3385
Herbaceous wetland	0.9886	0.3535	0.14	0.1115
Closed forest, evergreen needle leaf	0.9351	0.5549	0.7826	0.4808
Closed forest, deciduous broad leaf	0.9729	0.2584	0.0147	0.0141
Closed forest, mixed	0.9913	NaN	0	0
Closed forest, unknown	0.9962	NaN	0	0
Open forest, evergreen needle leaf	0.9812	NaN	0	0
Open forest, deciduous broad leaf	0.9942	NaN	0	0
Open forest, mixed	0.9997	NaN	0	0
Open forest, unknown	0.7733	0.5159	0.4275	0.3051
Open sea	0.9946	0.004	0.002	0.0013

Table A: List of classes with the corresponding segmentation metrics for the November scene based on the HR data.

Class	Accuracy	Precision	Recall	IoU
Shrubs	0.9995	NaN	0	0
Herbaceous vegetation	0.8402	0.1957	0.2035	0.1108
Cropland	0.3941	0.0366	0.6055	0.0357
Urban	0.9916	0.5192	0.4867	0.3355
Bare / sparse vegetation	1	NaN	0	0
Permanent water bodies	0.9743	0.0122	0.7655	0.0122
Herbaceous wetland	0.9686	0.003	0.1505	0.0003
Closed forest, evergreen needle leaf	0.4902	0.7444	0.0234	0.0232
Closed forest, deciduous broad leaf	0.9037	0.0546	0.0033	0.0031
Closed forest, mixed	0.8705	NaN	0	0
Closed forest, unknown	0.9772	NaN	0	0
Open forest, evergreen needle leaf	0.9833	NaN	0	0
Open forest, deciduous broad leaf	0.9912	NaN	0	0
Open forest, mixed	0.9986	NaN	0	0
Open forest, unknown	0.7674	0.0803	0.237	0.0638
Open sea	1	0	NaN	0

Table B: List of classes with the corresponding segmentation metrics for the Mountains scene based on the HR data.



# Bibliography

- [1] Michael Elad and Arie Feuer. Superresolution restoration of an image sequence: adaptive filtering approach. *IEEE Transactions on Image Processing*, 8(3):387–395, 1999.
- [2] Linwei Yue, Huanfeng Shen, Jie Li, Qiangqiang Yuan, Hongyan Zhang, and Liangpei Zhang. Image super-resolution: The techniques, applications, and future. *Signal Processing*, 128:389–408, 2016.
- [3] John F. Shanahan, James S. Schepers, Dennis D. Francis, Gary E. Varvel, Wallace W. Wilhelm, James M. Tringe, Mike R. Schlemmer, and David J. Major. Use of Remote-Sensing Imagery to Estimate Corn Grain Yield. *Agronomy Journal*, 93(3):583–589, 2001.
- [4] Yee Kit Chan and Voon Chet Koo. An introduction to synthetic aperture radar (SAR). *Progress In Electromagnetics Research*, 2:27–60, 2008.
- [5] Gotthard Meinel and Marco Neubert. A comparison of segmentation programs for high resolution remote sensing data. *International Archives of Photogrammetry and Remote Sensing*, 35(Part B):1097–1105, 2004.
- [6] Weixun Zhou, Shawn Newsam, Congmin Li, and Zhenfeng Shao. Learning low dimensional convolutional neural networks for high-resolution remote sensing image retrieval. *Remote Sensing*, 9(5):489, 2017.
- [7] Daiqin Yang, Zimeng Li, Yatong Xia, and Zhenzhong Chen. Remote sensing image super-resolution: Challenges and approaches. In *2015 IEEE International Conference on Digital Signal Processing (DSP)*, pages 196–200. IEEE, 2015.
- [8] Penghai Wu, Huanfeng Shen, Liangpei Zhang, and Frank-Michael Göttsche. Integrated fusion of multi-scale polar-orbiting and geostationary satellite observations for the mapping of high spatial and temporal resolution land surface temperature. *Remote Sensing of Environment*, 156:169–181, 2015.
- [9] A. Chaponniere, Philippe Maisongrande, Benoît Duchemin, L. Hanich, Gilles Boulet, Richard Escadafal, and S. Elouaddat. A combined high and low spatial resolution approach for mapping snow covered areas in the Atlas mountains. *International Journal of Remote Sensing*, 26(13):2755–2777, 2005.

- [10] Daniel Glasner, Shai Bagon, and Michal Irani. Super-resolution from a single image. In *2009 IEEE 12th International Conference on Computer Vision*, pages 349–356. IEEE, 2009.
- [11] Dinh-Hoan Trinh, Marie Luong, Francoise Dibos, Jean-Marie Rocchisani, Canh-Duong Pham, and Truong Q. Nguyen. Novel example-based method for super-resolution and denoising of medical images. *IEEE Transactions on Image Processing*, 23(4):1882–1895, 2014.
- [12] Jingwei Guan, Cheng Pan, Songnan Li, and Dahai Yu. SRDGAN: learning the noise prior for Super Resolution with Dual Generative Adversarial Networks. *arXiv:1903.11821*, 2019.
- [13] Grigorios Tsagkatakis, Anastasia Aidini, Konstantina Fotiadou, Michalis Giannopoulos, Anastasia Pentari, and Panagiotis Tsakalides. Survey of Deep-Learning Approaches for Remote Sensing Observation Enhancement. *Sensors*, 19(18):3929, 2019.
- [14] Xiaoran Shi, Feng Zhou, Shuang Yang, Zijing Zhang, and Tao Su. Automatic target recognition for synthetic aperture radar images based on super-resolution generative adversarial network and deep convolutional neural network. *Remote Sensing*, 11(2):135, 2019.
- [15] Xiaomin Yang, Wei Wu, Kai Liu, Pyoung Won Kim, Arun Kumar Sangaiah, and Gwanggil Jeon. Long-distance object recognition with image super resolution: A comparative study. *IEEE Access*, 6:13429–13438, 2018.
- [16] Muhammad Haris, Greg Shakhnarovich, and Norimichi Ukita. Task-driven Super Resolution: Object Detection in Low-resolution Images. *arXiv:1803.11316*, 2018.
- [17] Li Wang, Dong Li, Yousong Zhu, Lu Tian, and Yi Shan. Dual Super-Resolution Learning for Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3774–3783, 2020.
- [18] Thomas Vandal, Evan Kodra, Sangram Ganguly, Andrew Michaelis, Ramakrishna Nemani, and Auroop R. Ganguly. DeepSD: Generating high resolution climate change projections through single image super-resolution. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1663–1672, 2017.
- [19] Sheng Cao, Chao-Yuan Wu, and Philipp Krähenbühl. Lossless Image Compression through Super-Resolution. *arXiv:2004.02872*, 2020.
- [20] Vahid Naeimi, Stefano Elefante, Senmao Cao, Wolfgang Wagner, Alena Dostalova, and Bernhard Bauer-Marschallinger. Geophysical parameters retrieval from Sentinel-1 SAR data: a case study for high performance computing at EODC. In *Proceedings of the 24th High Performance Computing Symposium*, pages 1–8, 2016.



- [21] Ruben Fernandez-Beltran, Pedro Latorre-Carmona, and Filiberto Pla. Single-frame super-resolution in remote sensing: A practical overview. *International Journal of Remote Sensing*, 38(1):314–354, 2017.
- [22] Lukas Liebel and Marco Körner. Single-image super resolution for multispectral remote sensing data using convolutional neural networks. *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 41:883–890, 2016.
- [23] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.
- [24] Neeraj Kumar, Ruchika Verma, and Amit Sethi. Convolutional neural networks for wavelet domain super resolution. *Pattern Recognition Letters*, 90:65–71, 2017.
- [25] Jay Fussell, Donald Rundquist, and John Harrington. On defining remote sensing. *Photogrammetric Engineering and Remote Sensing*, 52(9):1507–1511, 1986.
- [26] Kamlesh Lulla Sabins Jr. *Remote Sensing: Principles and Interpretation*, volume 2. Taylor & Francis, 1987.
- [27] David P. Lusch. Introduction to microwave remote sensing. *Center for Remote Sensing and Geographic Information Science Michigan State University*, 1999.
- [28] Neha Joshi, Matthias Baumann, Andrea Ehammer, Rasmus Fensholt, Kenneth Grogan, Patrick Hostert, Martin Jepsen, Tobias Kuemmerle, Patrick Meyfroidt, Edward Mitchard, and et al. A Review of the Application of Optical and Radar Remote Sensing Data Fusion to Land Use Mapping and Monitoring. *Remote Sensing*, 8(1):70, 2016.
- [29] Merrill Ivan Skolnik. Introduction to radar. *Radar Handbook*, 2:21, 1962.
- [30] Kiyo Tomiyasu. Tutorial review of synthetic-aperture radar (SAR) with applications to imaging of the ocean surface. *Proceedings of the IEEE*, 66(5):563–583, 1978.
- [31] Alberto Moreira, Pau Prats-Iraola, Marwan Younis, Gerhard Krieger, Irena Hajnsek, and Konstantinos P. Papathanassiou. A tutorial on synthetic aperture radar. *IEEE Geoscience and Remote Sensing Magazine*, 1(1):6–43, 2013.
- [32] Jia Liu, Maoguo Gong, Kai Qin, and Puzhao Zhang. A deep convolutional coupling network for change detection based on heterogeneous optical and radar images. *IEEE Transactions on Neural Networks and Learning Systems*, 29(3):545–559, 2016.
- [33] Anthony Freeman. SAR calibration: An overview. *IEEE Transactions on Geoscience and Remote Sensing*, 30(6):1107–1121, 1992.

- [34] Charles E. Livingstone and Mark R. Drinkwater. Springtime C-band SAR backscatter signatures of Labrador Sea marginal ice: measurements versus modeling predictions. *IEEE Transactions on Geoscience and Remote Sensing*, 29(1):29–41, 1991.
- [35] Phillip M. Stepanian, Kyle G. Horton, Valery M. Melnikov, Dušan S. Zrnić, and Sidney A. Gauthreaux Jr. Dual-polarization radar products for biological applications. *Ecosphere*, 7(11):e01539, 2016.
- [36] Anthony Freeman, Yuhshyen Shen, and Charles L. Werner. Polarimetric SAR calibration experiment using active radar calibrators. *IEEE Transactions on Geoscience and Remote Sensing*, 28(2):224–240, 1990.
- [37] Alexis A. Mouche, Bertrand Chapron, Biao Zhang, and Romain Husson. Combined co-and cross-polarized SAR measurements under extreme wind conditions. *IEEE Transactions on Geoscience and Remote Sensing*, 55(12):6746–6755, 2017.
- [38] Dipanwita Haldar, Pooja Rana, Manoj Yadav, R.S. Hooda, and Manab Chakraborty. Time series analysis of co-polarization phase difference (PPD) for winter field crops using polarimetric C-band SAR data. *International Journal of Remote Sensing*, 37(16):3753–3770, 2016.
- [39] Francesco Mattia, Thuy Le Toan, Jean-Claude Souyris, Giacomo De Carolis, Nicolas Floury, Franco Posa, and Guido Pasquariello. The effect of surface roughness on multifrequency polarimetric SAR data. *IEEE Transactions on Geoscience and Remote Sensing*, 35(4):954–966, 1997.
- [40] Christophe Proisy, Eric Mougin, Eric Dufrêne, and Valérie Le Dantec. Monitoring seasonal changes of a mixed temperate forest using ERS SAR observations. *IEEE Transactions on Geoscience and Remote Sensing*, 38(1):540–552, 2000.
- [41] Katarzyna Dabrowska-Zielinska, Yoshio Inoue, Wanda Kowalik, and Maria Gruszczynska. Inferring the effect of plant and soil variables on C-and L-band SAR backscatter over agricultural fields, based on model analysis. *Advances in Space Research*, 39(1):139–148, 2007.
- [42] Jakob van Zyl, Bruce Chapman, Pascale Dubois, and Jiancheng Shi. The effect of topography on SAR calibration. *IEEE Transactions on Geoscience and Remote Sensing*, 31(5):1036–1043, 1993.
- [43] Fawwaz Ulaby, Percy Batlivala, and Myron Dobson. Microwave Backscatter Dependence on Surface Roughness, Soil Moisture, and Soil Texture: Part I-Bare Soil. *IEEE Transactions on Geoscience Electronics*, 16(4):286–295, 1978.
- [44] Marius Rüetschi, David Small, and Lars T. Waser. Rapid detection of windthrows using Sentinel-1 C-band SAR data. *Remote Sensing*, 11(2):115, 2019.

- [45] Lukas Novotny. Effective wavelength scaling for optical antennas. *Physical review letters*, 98(26):266802, 2007.
- [46] Nicolas Baghdadi, Mohammad Choker, Mehrez Zribi, Mohammad El Hajj, Simonetta Paloscia, Niko EC Verhoest, Hans Lievens, Frederic Baup, and Francesco Mattia. A new empirical model for radar scattering from bare soil surfaces. *Remote Sensing*, 8(11):920, 2016.
- [47] Y. Dong, B. Forster, and C. Ticehurst. Radar backscatter analysis for urban environments. *International Journal of Remote Sensing*, 18(6):1351–1364, 1997.
- [48] Shuai Xu, Zhixin Qi, Xia Li, and Anthony Gar-On Yeh. Investigation of the effect of the incidence angle on land cover classification using fully polarimetric SAR images. *International Journal of Remote Sensing*, 40(4):1576–1593, 2019.
- [49] Marcus E. Engdahl and Juha M. Hyyppä. Land-cover classification using multi-temporal ERS-1/2 InSAR data. *IEEE Transactions on Geoscience and Remote Sensing*, 41(7):1620–1628, 2003.
- [50] Andrea Puzzi Nicolau, Africa Flores-Anderson, Robert Griffin, Kelsey Herndon, and Franz J Meyer. Assessing SAR C-band data to effectively distinguish modified land uses in a heavily disturbed Amazon forest. *International Journal of Applied Earth Observation and Geoinformation*, 94:102214, 2021.
- [51] Paolo Ferrazzoli, Simonetta Paloscia, Paolo Pampaloni, Giovanni Schiavon, Simone Sigismondi, and Domenico Solimini. The potential of multifrequency polarimetric SAR in assessing agricultural and arboreal biomass. *IEEE Transactions on Geoscience and Remote Sensing*, 35(1):5–17, 1997.
- [52] Hao-Yu Liao and Tzai-Hung Wen. Extracting urban water bodies from high-resolution radar images: Measuring the urban surface morphology to control for radar’s double-bounce effect. *International Journal of Applied Earth Observation and Geoinformation*, 85:102003, 2020.
- [53] Paul A. Hwang, Derek M. Burrage, David W. Wang, and Joel C. Wesson. Ocean surface roughness spectrum in high wind condition for microwave backscatter and emission computations. *Journal of Atmospheric and Oceanic Technology*, 30(9):2168–2188, 2013.
- [54] Mariette Vreugdenhil, Wolfgang Wagner, Bernhard Bauer-Marschallinger, Isabella Pfeil, Irene Teubner, Christoph Rüdiger, and Peter Strauss. Sensitivity of Sentinel-1 backscatter to vegetation dynamics: An Austrian case study. *Remote Sensing*, 10(9):1396, 2018.
- [55] Amanda Veloso, Stéphane Mermoz, Alexandre Bouvet, Thuy Le Toan, Milena Planells, Jean-François Dejoux, and Eric Ceschia. Understanding the temporal behavior of crops using Sentinel-1 and Sentinel-2-like data for agricultural applications. *Remote Sensing of Environment*, 199:415–426, 2017.

- [56] Jean-Michel Martinez and Thuy Le Toan. Mapping of flood dynamics and spatial distribution of vegetation in the Amazon floodplain using multitemporal SAR data. *Remote Sensing of Environment*, 108(3):209–223, 2007.
- [57] Son Nghiem and George A. Leshkevich. Satellite SAR remote sensing of Great Lakes ice cover, Part 1. Ice backscatter signatures at C band. *Journal of Great Lakes Research*, 33(4):722–735, 2007.
- [58] Dorothy K. Hall, Daniel B. Fagre, Fritz Klasner, Gregg Linebaugh, and Glen E. Liston. Analysis of ERS 1 synthetic aperture radar data of frozen lakes in northern Montana and implications for climate studies. *Journal of Geophysical Research: Oceans*, 99(C11):22473–22482, 1994.
- [59] Gregory P. Asner. Cloud cover in Landsat observations of the Brazilian Amazon. *International Journal of Remote Sensing*, 22(18):3855–3862, 2001.
- [60] Andrea Meraner, Patrick Ebel, Xiao Xiang Zhu, and Michael Schmitt. Cloud removal in Sentinel-2 imagery using a deep residual neural network and SAR-optical data fusion. *ISPRS Journal of Photogrammetry and Remote Sensing*, 166:333–346, 2020.
- [61] Julien Travelletti, Christophe Delacourt, Pascal Allemand, J-P Malet, Jean Schmittbuhl, Renaud Toussaint, and Mickael Bastard. Correlation of multi-temporal ground-based optical images for landslide monitoring: Application, potential and limitations. *ISPRS Journal of Photogrammetry and Remote Sensing*, 70:39–55, 2012.
- [62] Fabrizio Argenti, Alessandro Lapini, Tiziano Bianchi, and Luciano Alparone. A tutorial on speckle reduction in synthetic aperture radar images. *IEEE Geoscience and Remote Sensing Magazine*, 1(3):6–35, 2013.
- [63] Felix Bachofer, Geraldine Quénéhervé, Thimm Zwiener, Michael Maerker, and Volker Hochschild. Comparative analysis of Edge Detection techniques for SAR images. *European Journal of Remote Sensing*, 49(1):205–224, 2016.
- [64] Sithara Kanakaraj, Madhu S. Nair, and Saidalavi Kalady. SAR image super resolution using importance sampling unscented Kalman filter. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(2):562–571, 2017.
- [65] Alexander Loew and Wolfram Mauser. Generation of geometrically and radiometrically terrain corrected SAR image products. *Remote Sensing of Environment*, 106(3):337–349, 2007.
- [66] Douglas J. Goering, Hao Chen, Larry D. Hinzman, and Douglas L. Kane. Removal of terrain effects from SAR satellite imagery of arctic tundra. *IEEE Transactions on Geoscience and Remote Sensing*, 33(1):185–194, 1995.

- [67] M. Sivakumar. Satellite remote sensing and GIS applications in agricultural meteorology. *Proceedings of the Training Workshop 7-11 July, 2003, World Meteorological Organisation*, 2003.
- [68] Taskin Kavzoglu and Ismail Colkesen. A kernel functions analysis for support vector machines for land cover classification. *International Journal of Applied Earth Observation and Geoinformation*, 11(5):352–359, 2009.
- [69] Ming-Hseng Tseng, Sheng-Jhe Chen, Gwo-Haur Hwang, and Ming-Yu Shen. A genetic algorithm rule-based approach for land-cover classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 63(2):202–212, 2008.
- [70] Sean Borman and Robert L. Stevenson. Super-resolution from image sequences-a review. In *1998 Midwest Symposium on Circuits and Systems (Cat. No. 98CB36268)*, pages 374–378. IEEE, 1998.
- [71] Mario Bertero and Patrizia Boccacci. *Introduction to Inverse Problems in Imaging*. CRC Press, 2020.
- [72] Masakazu Kojima. Strongly stable stationary solutions in nonlinear programs. In *Analysis and Computation of Fixed Points*, pages 93–138. Elsevier, 1980.
- [73] Vivek Bannore. Regularization for super-resolution image reconstruction. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pages 36–46. Springer, 2006.
- [74] Andreas Lugmayr, Martin Danelljan, Luc Van Gool, and Radu Timofte. Srflow: Learning the super-resolution space with normalizing flow. In *European Conference on Computer Vision*, pages 715–732. Springer, 2020.
- [75] Marcel C. Buhler, Andrés Romero, and Radu Timofte. DeepSEE: Deep Disentangled Semantic Explorative Extreme Super-Resolution. *Computer Vision - ACCV 2020 Lecture Notes in Computer Science*, pages 624–642, 2020.
- [76] Jan Allebach and Ping Wah Wong. Edge-directed interpolation. In *Proceedings of 3rd IEEE International Conference on Image Processing*, volume 3, pages 707–710. IEEE, 1996.
- [77] Hasan Demirel and Gholamreza Anbarjafari. Satellite image resolution enhancement using complex wavelet transform. *IEEE Geoscience and Remote Sensing Letters*, 7(1):123–126, 2009.
- [78] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2):295–307, 2015.

- [79] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1646–1654, 2016.
- [80] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.
- [81] Jin Wang, Bo Peng, and Xuejie Zhang. Using a stacked residual LSTM model for sentiment intensity prediction. *Neurocomputing*, 322:93–101, 2018.
- [82] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.
- [83] Oyebade K. Oyedotun, Djamila Aouada, Björn Ottersten, et al. Going deeper with neural networks without skip connections. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 1756–1760. IEEE, 2020.
- [84] Lena Wagner, Lukas Liebel, and Marco Körner. Deep Residual Learning For Single-Image Super-Resolution Of Multi-Spectral Satellite Imagery. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, IV-2/W7:189–196, 2019.
- [85] Ningbo Huang, Yong Yang, Junjie Liu, Xinchao Gu, and Hua Cai. Single-image super-resolution for remote sensing data using deep residual-learning neural network. In *International Conference on Neural Information Processing*, pages 622–630. Springer, 2017.
- [86] Zhihao Wang, Jian Chen, and Steven C.H. Hoi. Deep Learning for Image Super-Resolution: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10):3365–3387, 2021.
- [87] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1125–1134, 2017.
- [88] Jin Zhu, Guang Yang, and Pietro Lio. How can we make GAN perform better in single medical image super-resolution? A lesion focused multi-scale approach. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pages 1669–1673. IEEE, 2019.
- [89] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4681–4690, 2017.



- [90] Kui Jiang, Zhongyuan Wang, Peng Yi, Guangcheng Wang, Tao Lu, and Junjun Jiang. Edge-enhanced GAN for remote sensing image superresolution. *IEEE Transactions on Geoscience and Remote Sensing*, 57(8):5799–5812, 2019.
- [91] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. *Lecture Notes in Computer Science Computer Vision – ECCV 2018 Workshops*, pages 63–79, 2019.
- [92] Jakaria Rabbi, Nilanjan Ray, Matthias Schubert, Subir Chowdhury, and Dennis Chao. Small-Object Detection in Remote Sensing Images with End-to-End Edge-Enhanced GAN and Object Detector Network. *Remote Sensing*, 12(9):1432, 2020.
- [93] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A. Bharath. Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, 35(1):53–65, 2018.
- [94] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2472–2481, 2018.
- [95] Zhen Li, Jinglei Yang, Zheng Liu, Xiaomin Yang, Gwanggil Jeon, and Wei Wu. Feedback network for image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3867–3876, 2019.
- [96] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. Second-order attention network for single image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11065–11074, 2019.
- [97] Longgang Wang, Mana Zheng, Wenbo Du, Menglin Wei, and Lianlin Li. Super-resolution SAR image reconstruction via generative adversarial network. In *2018 12th International Symposium on Antennas, Propagation and EM Theory (ISAPE)*, pages 1–4. IEEE, 2018.
- [98] Ce Zheng, Xue Jiang, Ye Zhang, Xingzhao Liu, Bin Yuan, and Zhixin Li. Self-Normalizing Generative Adversarial Network for Super-Resolution Reconstruction of SAR Images. In *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 1911–1914. IEEE, 2019.
- [99] Naser Karimi and Mohammad Reza Taban. A convex variational method for super resolution of SAR image with speckle noise. *Signal Processing: Image Communication*, page 116061, 2020.
- [100] Sithara Kanakaraj, Madhu S. Nair, and Saidalavi Kalady. Adaptive Importance Sampling Unscented Kalman Filter With Kernel Regression for SAR Image Super-Resolution. *IEEE Geoscience and Remote Sensing Letters*, pages 1–5, 2020.



- [101] Huanfeng Shen, Liupeng Lin, Jie Li, Qiangqiang Yuan, and Lingli Zhao. A residual convolutional neural network for polarimetric SAR image super-resolution. *ISPRS Journal of Photogrammetry and Remote Sensing*, 161:90–108, 2020.
- [102] Liupeng Lin, Jie Li, Qiangqiang Yuan, and Huanfeng Shen. Polarimetric SAR Image Super-Resolution VIA Deep Convolutional Neural Network. In *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 3205–3208. IEEE, 2019.
- [103] Debora Pastina, Pierfrancesco Lombardo, Alfonso Farina, and Piero Daddi. Super-resolution of polarimetric SAR images of ship targets. *Signal Processing*, 83(8):1737–1748, 2003.
- [104] Krishna Kant Singh and Akansha Singh. A study of image segmentation algorithms for different types of images. *International Journal of Computer Science Issues (IJCSI)*, 7(5):414, 2010.
- [105] Jing Li, Xin Zhang, Jiehao Li, Yanyu Liu, and Junzheng Wang. Building and optimization of 3D semantic map based on Lidar and camera fusion. *Neurocomputing*, 409:394–407, 2020.
- [106] Mahesh Pal. Random forest classifier for remote sensing classification. *International Journal of Remote Sensing*, 26(1):217–222, 2005.
- [107] Peter M. Atkinson and P. Lewis. Geostatistical classification for remote sensing: an introduction. *Computers & Geosciences*, 26(4):361–371, 2000.
- [108] Pabitra Mitra, B. Uma Shankar, and Sankar K. Pal. Segmentation of multispectral remote sensing images using active support vector machines. *Pattern Recognition Letters*, 25(9):1067–1074, 2004.
- [109] M. Neubert and G. Meinel. Evaluation of segmentation programs for high resolution remote sensing applications. In *Proc. Joint ISPRS/EARSeL Workshop High Resolution Mapping from Space*, page 8. Citeseer, 2003.
- [110] Jia Deng, Alexander C Berg, Kai Li, and Li Fei-Fei. What does classifying more than 10,000 image categories tell us? In *European Conference on Computer Vision*, pages 71–84. Springer, 2010.
- [111] Lichao Mou and Xiao Xiang Zhu. RiFCN: Recurrent Network in Fully Convolutional Network for Semantic Segmentation of High Resolution Remote Sensing Images. *arXiv:1805.02091*, 2018.
- [112] Chengquan Huang, L. Davis, and J. Townshend. An assessment of support vector machines for land cover classification. *International Journal of Remote Sensing*, 23(4):725–749, 2002.

- [113] Mehdi Khoshboresh Masouleh and Reza Shah-Hosseini. Development and evaluation of a deep learning model for real-time ground vehicle semantic segmentation from UAV-based thermal infrared imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 155:172–186, 2019.
- [114] Wenxiu Wang, Yutian Fu, Feng Dong, and Feng Li. Semantic segmentation of remote sensing ship image via a convolutional neural networks model. *IET Image Processing*, 13(6):1016–1022, 2019.
- [115] Cindy Gonzales and Wesam Sakla. Semantic segmentation of clouds in satellite imagery using deep pre-trained U-nets. In *2019 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, pages 1–7. IEEE, 2019.
- [116] N. Venugopal. Automatic semantic segmentation with DeepLab dilated learning network for change detection in remote sensing images. *Neural Processing Letters*, pages 1–23, 2020.
- [117] Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. Segment-before-detect: Vehicle detection and classification through semantic segmentation of aerial images. *Remote Sensing*, 9(4):368, 2017.
- [118] Haiping Yang, Bo Yu, Jiancheng Luo, and Fang Chen. Semantic segmentation of high spatial resolution images with deep neural networks. *GIScience & Remote Sensing*, 56(5):749–768, 2019.
- [119] Björn Fröhlich, Erik Rodner, and Joachim Denzler. Semantic segmentation with millions of features: Integrating multiple cues in a combined random forest approach. In *Asian conference on computer vision*, pages 218–231. Springer, 2012.
- [120] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.
- [121] Benjamin Graham. Fractional Max-Pooling. *arXiv:1412.6071*, 2015.
- [122] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. *arXiv:1603.07285*, 2018.
- [123] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017.
- [124] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556*, 2015.
- [125] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

- [126] Zemin Han, Yuanyong Dian, Hao Xia, Jingjing Zhou, Yongfeng Jian, Chonghuai Yao, Xiong Wang, and Yuan Li. Comparing fully deep convolutional neural networks for land cover classification with high-spatial-resolution Gaofen-2 images. *ISPRS International Journal of Geo-Information*, 9(8):478, 2020.
- [127] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, Ben Glocker, and Daniel Rueckert. Attention U-Net: Learning Where to Look for the Pancreas. *arXiv:1804.03999*, 2018.
- [128] Mina Jafari, Ruizhe Li, Yue Xing, Dorothee Auer, Susan Francis, Jonathan Garibaldi, and Xin Chen. FU-net: multi-class image segmentation using feedback weighted U-net. In *International Conference on Image and Graphics*, pages 529–537. Springer, 2019.
- [129] Leon Gatys, Alexander S. Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. *Advances in Neural Information Processing Systems*, 28:262–270, 2015.
- [130] Walter Hugo Lopez Pinaya, Sandra Vieira, Rafael Garcia-Dias, and Andrea Mechelli. Convolutional neural networks. In *Machine learning*, pages 173–191. Elsevier, 2020.
- [131] P. Ittiyavirah Sibi, S. Allwyn Jones, and P. Siddarth. Analysis of different activation functions using back propagation neural networks. *Journal of Theoretical and Applied Information Technology*, 47(3):1264–1268, 2013.
- [132] Thamer M. Jamel and Ban Mohammed Khammas. Implementation of a sigmoid activation function for neural network using FPGA. In *13th Scientific Conference of Al-Ma'moon University College*, volume 13, 2012.
- [133] Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv:1803.08375*, 2018.
- [134] Andrew L. Maas, Awni Y. Hannun, Andrew Y. Ng, et al. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3. Citeseer, 2013.
- [135] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1026–1034, 2015.
- [136] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1874–1883, 2016.

- [137] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. IEEE, 2009.
- [138] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The Handbook of Brain Theory and Neural Networks*, 3361(10):1995, 1995.
- [139] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4700–4708, 2017.
- [140] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Citeseer*, 2009.
- [141] Tong Tong, Gen Li, Xiejie Liu, and Qinquan Gao. Image super-resolution using dense skip connections. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4799–4807, 2017.
- [142] Wei Zhang, Xiang Li, and Qian Ding. Deep residual learning-based fault diagnosis method for rotating machinery. *ISA Transactions*, 95:295–305, 2019.
- [143] Joseph J. Lim, C. Lawrence Zitnick, and Piotr Dollár. Sketch tokens: A learned mid-level representation for contour and object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3158–3165, 2013.
- [144] Jianbo Jiao, Wei-Chih Tu, Shengfeng He, and Rynson W.H. Lau. Formresnet: Formatted residual learning for image restoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 38–46, 2017.
- [145] Jinghui Chu, Jiaqi Zhang, Wei Lu, and Xiangdong Huang. A novel multiconnected convolutional network for super-resolution. *IEEE Signal Processing Letters*, 25(7):946–950, 2018.
- [146] Mehrdad Shoeiby, Antonio Robles-Kelly, Ran Wei, and Radu Timofte. PIRM2018 Challenge on Spectral Image Super-Resolution: Dataset and Study. *Lecture Notes in Computer Science Computer Vision - ECCV 2018 Workshops*, pages 276–287, 2019.
- [147] Jason Brownlee. *Deep Learning for Computer Vision: Image Classification, Object Detection, and Face Recognition in Python*. Machine Learning Mastery, 2019.
- [148] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456. PMLR, 2015.

- [149] Andrew M. Shackleton and Abdulrahman M. Altahhan. A Comparison Study of Deep Learning Techniques to Increase the Spatial Resolution of Photo-Realistic Images. In *International Conference on Neural Information Processing*, pages 341–348. Springer, 2019.
- [150] Wenzhe Shi, Jose Caballero, Lucas Theis, Ferenc Huszar, Andrew Aitken, Christian Ledig, and Zehan Wang. Is the deconvolution layer the same as a convolutional layer? *arXiv:1609.07009*, 2016.
- [151] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016.
- [152] Joan Bruna, Pablo Sprechmann, and Yann LeCun. Super-resolution with deep convolutional sufficient statistics. *arXiv:1511.05666*, 2015.
- [153] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 136–144, 2017.
- [154] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3883–3891, 2017.
- [155] Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard GAN. *arXiv:1807.00734*, 2018.
- [156] Jinhua Wang, Xuewei Li, and Hongzhe Liz. Exposure Fusion Using a Relative Generative Adversarial Network. *IEICE Transactions on Information and Systems*, 104(7):1017–1027, 2021.
- [157] Behzad Kamgar-Parsi and Azriel Rosenfeld. Optimally isotropic laplacian operator. *IEEE Transactions on Image Processing*, 8(10):1467–1472, 1999.
- [158] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 624–632, 2017.
- [159] Pierre Charbonnier, Laure Blanc-Feraud, Gilles Aubert, and Michel Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *Proceedings of 1st International Conference on Image Processing*, volume 2, pages 168–172. IEEE, 1994.
- [160] Feng Sun, Guanci Yang, Ansi Zhang, Yiyun Zhang, et al. Circle-U-Net: An Efficient Architecture for Semantic Segmentation. *Algorithms*, 14(6):159, 2021.

- [161] Marc'Aurelio Ranzato, Fu Jie Huang, Y-Lan Boureau, and Yann LeCun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *2007 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2007.
- [162] Thorsten Falk, Dominic Mai, Robert Bensch, Özgün Çiçek, Ahmed Abdulkadir, Yassine Marrakchi, Anton Böhm, Jan Deubner, Zoe Jäckel, Katharina Seiwald, et al. U-Net: deep learning for cell counting, detection, and morphometry. *Nature Methods*, 16(1):67–70, 2019.
- [163] Evert Attema, Pierre Bargellini, Peter Edwards, Guido Levrini, Svein Lokas, Ludwig Moeller, Betlem Rosich, Patrizia Secchi, Ramon Torres, Malcolm Davidson, and Paul Snoeij. The radar mission for GMES operational land and sea services. *ESA Bulletin*, 131:10–17, 2007.
- [164] Sentinel-1 Team. Sentinel-1 User Handbook. *European Space Agency*, GMES-S1OP-EOPG-TN-13-0001, 2013, <https://sentinel.esa.int/> (accessed on 8<sup>th</sup> of November, 2020).
- [165] Bernhard Bauer-Marschallinger, Daniel Sabel, and Wolfgang Wagner. Optimisation of global grids for high-resolution remote sensing data. *Computers & Geosciences*, 72:84–93, 2014.
- [166] Marcel Buchhorn, Myroslava Lesiv, Nandin-Erdene Tsendbazar, Martin Herold, Luc Bertels, and Bruno Smets. Copernicus global land cover layers—collection 2. *Remote Sensing*, 12(6):1044, 2020.
- [167] Marcel Buchhorn, Bruno Smets, Luc Bertels, Myroslava Lesiv, Nandin-Erdene Tsendbazar, and Linlin Li. Copernicus Global Land Service: Land Cover 100m: version 2 Globe 2015: Product User Manual, November 2019.
- [168] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 126–135, 2017.
- [169] Maria Carolina Monard and Geapa Batista. Learning with skewed class distributions. *Advances in Logic, Artificial Intelligence and Robotics*, 85:173–180, 2002.
- [170] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv:1712.04621*, 2017.
- [171] Connor Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48, 2019.
- [172] Sebastien C. Wong, Adam Gatt, Victor Stamatescu, and Mark D. McDonnell. Understanding data augmentation for classification: when to warp? In *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–6. IEEE, 2016.



- [173] Shuangting Liu, Jiaqi Zhang, Yuxin Chen, Yifan Liu, Zengchang Qin, and Tao Wan. Pixel level data augmentation for semantic image segmentation using generative adversarial networks. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1902–1906. IEEE, 2019.
- [174] Shounak Chakraborty, Jayashree Phukan, Moumita Roy, and Bidyut Baran Chaudhuri. Handling the class imbalance in land-cover classification using bagging-based semisupervised neural approach. *IEEE Geoscience and Remote Sensing Letters*, 17(9):1493–1497, 2019.
- [175] Georgios Douzas, Fernando Bacao, Joao Fonseca, and Manvel Khudinyan. Imbalanced learning in land cover classification: Improving minority classes’ prediction accuracy using the geometric smote algorithm. *Remote Sensing*, 11(24):3040, 2019.
- [176] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014.
- [177] Lutz Prechelt. Early stopping-but when? In *Neural Networks: Tricks of the trade*, pages 55–69. Springer, 1998.
- [178] Ying Da Wang, Ryan T. Armstrong, and Peyman Mostaghimi. Enhancing resolution of digital rock images with super resolution convolutional neural networks. *Journal of Petroleum Science and Engineering*, 182:106261, 2019.
- [179] Mehdi S. M. Sajjadi, Bernhard Scholkopf, and Michael Hirsch. Enhancenet: Single image super-resolution through automated texture synthesis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4491–4500, 2017.
- [180] Muhammad Waleed Gondal, Bernhard Schölkopf, and Michael Hirsch. The unreasonable effectiveness of texture transfer for single image super-resolution. In *European Conference on Computer Vision*, pages 80–97. Springer, 2018.
- [181] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *arXiv:1602.07261*, 2016.
- [182] Xiaying Wang, Lukas Cavigelli, Manuel Eggimann, Michele Magno, and Luca Benini. Hr-SAR-NET: A deep neural network for urban scene segmentation from high-resolution SAR data. In *2020 IEEE Sensors Applications Symposium (SAS)*, pages 1–6. IEEE, 2020.
- [183] Md Atiqur Rahman and Yang Wang. Optimizing intersection-over-union in deep neural networks for image segmentation. In *International Symposium on Visual Computing*, pages 234–244. Springer, 2016.
- [184] Xiwei Zhang, Guillaume Thibault, Etienne Decencière, Beatriz Marcotegui, Bruno Lay, Ronan Danno, Guy Cazuguel, Gwénolé Quéllec, Mathieu Lamard, Pascale



- Massin, et al. Exudate detection in color retinal images for mass screening of diabetic retinopathy. *Medical Image Analysis*, 18(7):1026–1043, 2014.
- [185] Margherita Grandini, Enrico Bagli, and Giorgio Visani. Metrics for multi-class classification: an overview. *arXiv:2008.05756*, 2020.
- [186] Prateek Singh, Deepak Chahal, and Latika Kharb. Predictive Strength of Selected Classification Algorithms for Diagnosis of Liver Disease. In *Proceedings of ICRIC 2019*, pages 239–255. Springer, 2020.
- [187] Alain Horé and Djemel Ziou. Is there a relationship between peak-signal-to-noise ratio and structural similarity index measure? *IET Image Processing*, 7(1):12–24, 2013.
- [188] Jiayuan Peng, Chengyu Shi, Eric Laugeman, Weigang Hu, Zhen Zhang, Sasa Mutic, and Bin Cai. Implementation of the structural SIMilarity (SSIM) index as a quantitative evaluation tool for dose distribution error detection. *Medical physics*, 47(4):1907–1919, 2020.
- [189] Jari Korhonen and Junyong You. Peak signal-to-noise ratio revisited: Is simple beautiful? In *2012 Fourth International Workshop on Quality of Multimedia Experience*, pages 37–38. IEEE, 2012.
- [190] Susu Yao, Weisi Lin, EePing Ong, and Zhongkang Lu. Contrast signal-to-noise ratio for image quality assessment. In *IEEE International Conference on Image Processing 2005*, volume 1, pages I–397. IEEE, 2005.
- [191] Wang Yuanji, Li Jianhua, Lu Yi, Fu Yao, and Jiang Qinzong. Image quality evaluation based on image weighted separating block peak signal to noise ratio. In *International Conference on Neural Networks and Signal Processing, 2003. Proceedings of the 2003*, volume 2, pages 994–997 Vol.2, 2003.
- [192] Sithara Kanakaraj, Madhu S. Nair, and Saidalavi Kalady. Adaptive importance sampling unscented kalman filter based SAR image super resolution. *Computers & Geosciences*, 133:104310, 2019.
- [193] Wei Li, Natasha MacBean, Philippe Ciais, Pierre Defourny, Céline Lamarche, Sophie Bontemps, Richard A Houghton, and Shushi Peng. Gross and net land cover changes in the main plant functional types derived from the annual ESA CCI land cover maps (1992–2015). *Earth System Science Data*, 10(1):219–234, 2018.
- [194] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32:8026–8037, 2019.

- [195] Changhui Yan, Drena Dobbs, and Vasant Honavar. A two-stage classifier for identification of protein–protein interface residues. *Bioinformatics*, 20(Suppl 1):i371–i378, 2004.
- [196] Yeong-Sun Song, Hong-Gyoo Sohn, and Choung-Hwan Park. Efficient water area classification using Radarsat-1 SAR imagery in a high relief mountainous environment. *Photogrammetric Engineering & Remote Sensing*, 73(3):285–296, 2007.