Universität Leipzig
Fakultät für Mathematik und Informatik
Institut für Informatik

Master Thesis

Hyperspectral satellite image sharpening using image fusion and Convolutional Neural Networks

Practical applications for Earth observation

Leipzig, September 2024

Marvin Müller
Informatik, Master of Science

Supervising Professor:
Prof. Dr. Hannes Feilhauer
Fakultät Physik und Erdsystemwissenschaften
Institut für Erdsystemwissenschaft und Fernerkundung

Second Supervisor:
Dr. Daniel Wiegreffe
Fakultät für Mathematik und Informatik
Institut für Informatik
Abteilung für Bild- und Signalverarbeitung

Further supervision:
Daniel Mederer
Fakultät Physik und Erdsystemwissenschaften
Institut für Erdsystemwissenschaft und Fernerkundung

Prof. Teja Kattenborn
Universität Freiburg
Fakultät für Umwelt und Natürliche Ressourcen
Professur für Sensorgestützte Geoinformatik (geosense)

Abstract

This work presents the development and application of a machine learning based technique for enhancing the spatial resolution of hyperspectral remote sensing data from the Environmental Mapping and Analysis Program (EnMAP) satellite from 30 m $\times$ 30 m to 10 m $\times$ 10 m. The research uses state-of-the-art techniques in Convolutional Neural Networks (CNNs) for image super-resolution and image fusion.

After an analysis and comparison of existing super-resolution CNNs, current limitations are outlined and requirements for the EnMAP sharpening tasks are defined. On this basis a novel CNN architecture using three- and two-dimensional convolutional filters was developed to use auxiliary high-resolution multispectral data from the Sentinel-2 satellites to reconstruct spatial details while taking into account the spectral context of the low-resolution hyperspectral data.

To collect and prepare data for model training, a comprehensive preprocessing pipeline was developed, including tasks such as scene cropping, Sentinel-2 data fetching, cloud masking, and co-registration. The model training was done using the Wald protocol. The preprocessing tools developed for data collection and preparation can be reused and provide a framework for further improvements to the model and other super-resolution approaches using EnMAP data.

Using Wald's strategy and an evaluation dataset, the model was shown to improve the quality of EnMAP data at higher resolution compared to interpolation and kernel sharpening methods. Experiments at the target resolution using an airborne high-resolution hyperspectral scene also showed improvements made by the model. While challenges such as limited access to high-resolution hyperspectral data and computational resource demands remain, the tools and techniques developed in this work provide a foundation for further research.

To Mary, Mom, Dad, Emi and Nico.
Your support was everything!

Contents

List of Figures

1 Introduction

Digital images are conventionally composed of three values per pixel representing the colors red, green and blue. In most camera lenses, the brightness for a wavelength that the human retina interprets as red, green, or blue is captured for each pixel location by an image sensor with a color filter. While three colors in the visible spectrum are sufficient for photorealistic representations, capturing values for more wavelengths inside and outside the visible spectrum has many useful applications. Especially in satellite remote sensing, multispectral images (MSI) can be used for geological, military, and agricultural purposes, for example. Satellites for such Earth observation tasks are equipped with sensors that mostly measure the intensity of reflected electromagnetic radiation of a specific region in a given spectrum. Therefore, a value does not refer to a single wavelength, but to a band covering multiple wavelengths. The NASA and USGS Landsat 8 satellite, launched in February 2013, carries two sensors that together capture eleven spectral bands [1]. Sentinel-2 is a mission of the European Commission and the European Space Agency that aims to launch a series of satellites to provide multispectral imagery [32]. Currently, two satellites, launched in 2015 and 2017, are operating with sensors covering 13 spectral bands from visible and near infrared to shortwave infrared. For simplicity, satellite missions such as Sentinel-2 list a center wavelength instead of the spectral range for each band [14].

The term multispectral imagery usually describes the use of 3 to 15 bands. There are satellite missions that aim to acquire much more spectral information for a captured scene. In 2019, the Italian Space Agency launched the PRISMA satellite, which is capable of capturing 239 distinct bands. Of these, 66 are located between 400 and 1010 nm, while the remaining 173 are situated between wavelengths of 920 and 2505 nm [2]. The Environmental Mapping and Analysis Program (EnMAP) is a German satellite mission designed to bridge the gap between the quantity and quality of space-based and airborne imaging spectroscopy data [10]. Currently, most hyperspectral applications and studies rely on airborne or non-imaging near-surface data. EnMAP covers the spectral range from 420 nm to 2450 nm in 246 bands and initially provides reflectance data for 224 bands [4].

Many applications in different scientific fields can benefit from the amount of spectral information acquired by hyperspectral space missions. In object detection, hyperspectral images (HSIs) can improve results and help to better distinguish objects from similar backgrounds such as camouflage [22]. Hyperspectral spaceborne images also provide geological information that can be extracted for

[1] https://www.usgs.gov/landsat-missions/landsat-8
[2] https://www.eoportal.org/satellite-missions/prisma-hyperspectral

ore exploration, lithological and mineralogical mapping and environmental geology [26]. Another use case is the remote sensing of plant canopies, which can be utilized to map multiple functional plant traits [5]. PRISMA and EnMAP provide data that is used to monitor ecosystems and draw conclusions about their healthiness, development and role in the Earth system. Spatial and temporal observations of plant traits and soil degradation processes are powerful tools in agricultural management aimed at increasing crop yield and quality [3].

Both mentioned hyperspectral satellite missions capture a wide range of spectral information. However, their sensors have a fixed coarse spatial resolution of 30 m $\times$ 30 m per pixel [4], [3], which limits the informative value of their data and e.g. the extracted plant traits [3] or geological information [26]. To overcome the hardware limitations of satellites, super-resolution techniques can be used to spatially sharpen the data to a finer resolution. For natural image super-resolution conventional statistical or dictionary-based techniques are often used. With the increasing amount of computing power available, Convolutional Neural Networks (CNNs), which learn a mapping function between a low-resolution image and a high-resolution image, are already outperforming conventional methods. While most CNNs use two-dimensional convolutions, recent studies make use of three-dimensional convolutions that are able to learn spectral correlations between adjacent bands of similar wavelength. Another super-resolution approach is to fuse the low-resolution image with a higher-resolution image. Pan-sharpening is a numerical method that sharpens an RGB image using a high-resolution panchromatic image. Some approaches using CNNs adapt this technique to fuse the HSI with a higher resolution MSI.

A number of the proposed methodologies have already demonstrated the ability to achieve good results in terms of spatial sharpening for MSIs and, more recently, for HSIs. However, the majority of the used models were trained and evaluated on well-curated datasets that avoid the issues inherent to authentic remote sensing data, such as noise caused by clouds or haze. Moreover, the majority of models are designed to enhance a specific sensor or a specific dataset, and they often perform less well in comparison to other models when tested under different conditions. Moreover, a considerable number of researchers do not make their code publicly accessible, nor do they publish the trained model or the full list of parameters used for training. This severely limits the ability to apply their proposed methods to different use cases.

This work aims to develop a tool that fetches EnMAP satellite data and automatically generates preprocessed model input data according to given parameters. In addition, a CNN architecture is proposed and used to train a model that spatially sharpens the EnMAP data. This model is evaluated on a different dataset afterwards. The architecture is orientated towards existing architectures using three-dimensional convolutions and adapted to serve this specific use case. As multispectral data of higher resolution from Sentinel-2 is readily accessible, the preprocessing tool is extended by a scraping mechanism for Sentinel-2 data that is

[3] https://www.eoportal.org/satellite-missions/prisma-hyperspectral

integrated with the EnMAP data by the model, thereby facilitating the sharpening process. Evaluations show that the model effectively sharpens the EnMAP data from a resolution of 30 m $\times$ 30 m to 10 m $\times$ 10 m per pixel and achieves better results than other conventional sharpening methods. All tools that are designed to scrape and spatially enhance EnMAP remote sensing data are user-friendly designed and are maintained in a public GitHub repository [4].

In the following chapter, an overview of goals and products of the EnMAP mission is given. Also conventional super-resolution techniques and Convolutional Neural Networks are briefly explained. An overview of recent studies on CNNs for super-resolution and hyperspectral super-resolution is given and the most important techniques for this work such as the Wald protocol and 3D convolutions are explained in Chapter 3. Moreover, this chapter outlines the limitations of existing studies and identifies the challenges that must be addressed in this work. In Chapter 4 the methods that were developed to scrape and preprocess the hyperspectral satellite data are described. This chapter also depicts how the network architecture was designed. The exact implementation details and parameters, which hyperparameters were compared, as well as the obtained and used datasets are presented in the following chapter. Chapter 6 describes the evaluations performed to validate the sharpening and to test the generalization ability of the trained model. In the final chapters, the potential as well as some problems and possible improvements of the model are discussed before a brief conclusion of the work is given.

2 Background

2.1 The Environmental Mapping and Analysis Program

"The main scientific goal of the hyperspectral EnMAP mission is to study environmental changes, investigate ecosystem responses to human activities and monitor the management of natural resources." [4, p. 6]

The satellite, which was launched on April 1, 2022, is equipped with a hyperspectral imager that records reflected solar radiation from the Earth's surface. At the time of writing, 224 of 246 recorded bands between 420 and 2450 nm are available as a data product. With the ability to record images 30° off the location directly under the satellite's track, called off-nadir pointing, EnMAP can reduce the revisit time of a location to four days. The nominal revisit time is 27 days. With a short revisit time and high spectral range observations, the mission aims to contribute to scientific questions about climate change impacts, land cover changes, biodiversity processes and more [4].

The data products from EnMAP that are accessible to the user community are classified into three different levels. So called Level 1B data consists of the raw data that is corrected by a processor for e.g. radiometric non-uniformities and is annotated with auxiliary information. This top-of-atmosphere radiance data is further corrected to reduce geometric errors, spectral and atmospheric distortions resulting in Level 1C data. Level 2A (L2A) data is the atmospheric-corrected bottom-of-atmosphere data derived from Level 1C, consisting of reflectance data, metadata, and various pixel masks for pixels covered by e.g. clouds, cloud shadows, or haze [4].

EnMAP has an open data policy, and after registering with the EnMAP Instrument Planning Portal, users can download existing data or request new data acquisitions for a specific site. Those requests are prioritized by user category [4]. In this work, data from the existing L2A database, which is hosted by the German Aerospace Center (DLR)[1], is utilized. This collection is updated daily with new data, and registered users can download L2A product files. In order to collect a reasonable dataset, a data scraping tool is developed that automatically downloads files that match given parameters and is described in Chapter 4.

[1] `https://geoservice.dlr.de/eoc/ogc/stac/v1/collections/ENMAP_HSI_L2A`

2.2 Image super-resolution

2.2.1 Sub-pixel based analysis

Many different approaches, which can be categorized into different groups, are aimed at the spatial sharpening of images. Sub-pixel based analysis methods such as spectral mixture analysis and super-resolution mapping [12], non-negative matrix factorization or maximum a posteriori estimation specifically target the enhancement of classification or object detection in low-resolution images [25]. This work aims to sharpen an input image without altering its dimensionality, domain or range of values, so that further processing applications, e.g. classification or calculation of vegetation indices, remain possible. Therefore, sub-pixel based methods are not considered further.

2.2.2 Single-image super-resolution

Single-image super-resolution techniques map a low-resolution input to a high-resolution output image. A common strategy for reconstructing an image from a lower to a higher resolution is interpolation, i.e. calculating the values between the input samples [9]. Nearest neighbor interpolation assigns each interpolated output pixel the value of its nearest sample point. Bilinear interpolation performs two linear interpolations to calculate the average of the four closest sample points to determine the output pixel value, while bicubic interpolation calculates the value of the 16 nearest pixels. Due to the larger number of sample points involved, the results of bicubic interpolation are more accurate than those of bilinear or nearest neighbor interpolation, but also have a higher computational complexity. There are other interpolation methods such as B-spline or Kriging interpolation [9], but all of them tend to blur the edges of the input image and can produce jagged or ringing effects [25].

Another approach is to achieve super-resolution by using signal processing methods, such as applying filters to the different frequency components of the input image signal. This is done, for example, to amplify the high-frequency components of an image, such as edges or other detail information, to prevent blurring effects [28]. To achieve this, the input signal is often convolved with a filter signal. The convolution of a function f and a function g produces a third function denoted as $f * g$. Filtering a two-dimensional image A with height i_a and width j_a by applying a convolution with another function B results in a filtered image C. This is expressed with the following formula [16, Formula 4]:

$$A[i_a, j_a] * B[i_b, j_b] = C[i_c, j_c] = \sum_{\tau_1=0}^{i_a-1} \sum_{\tau_2=0}^{j_a-1} A[\tau_1, \tau_2] B[i - \tau_1, j - \tau_1] \tag{2.1}$$

A and B can be thought of as two two-dimensional matrices, with B sliding over

each element of A and applying element-wise products. The sum of these products is the element in the resulting matrix C [16]. The matrix B is often referred to as a kernel, filter kernel, or filter matrix. In image processing, there are various kernels with effects such as horizontal or vertical edge detection with the Sobel operators K_h and K_v described in [16, Formula 5]. For image sharpening, the process of unsharp masking (UM) can be used, where the input is at first blurred to an unsharp image and then substracted from the original image [27]. An often used filter kernel for UM is K_{UM}:

$$K_{UM} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \tag{2.2}$$

The use of fixed filter kernels is an easy way for fast image processing, but it applies the same change to all regions of an image in all spatial orientations. Better results can be obtained with flexible operators that vary based on the unisotropic properties of an image, such as the Adaptive Unsharp Masking method proposed by Polesel et al. [28] or the Kriging-Weighted Laplacian Kernels by Pham [27].

Basic UM methods usually amplify the noise contained in the input signal as well as high frequency components, and the results usually suffer from a halo effect [6]. The Generalized Unsharp Masking algorithm by Deng [6] reduces these problems for RGB images. Most advanced UM methods are optimized for a specific multispectral domain, typically RGB, or ignore spectral context and correlations between bands.

A super-resolution tensor-based method for hyperspectral images that addresses both the spatial and spectral domains of HSIs is proposed by Wang et al. [35]. Like other complex optimization problems, it must be solved inefficiently in iterative steps [20].

With the increasing availability of computing power, deep machine learning methods, mostly Convolutional Neural Networks (CNNs), have recently become very popular in research for the sharpening of multi- and hyperspectral images. Deep learning models can be trained to find a complex nonlinear mapping function to create high-resolution images directly from low-resolution input images. There are various recent approaches for single-image super-resolution with CNNs such as the very deep network VDSR with 2D convolutions by Kim et al. [15], the mixed network MCNet by Li et al. using 2D and 3D convolutions [18], the mixed transformer and 3D convolutional network by Liu, Y. et al. [20], or the full 3D networks 3D-FCNN by Mei et al. [25] and F3DUN by Liu, Z et al. [21]. A brief functional explanation of CNNs is given in Section 2.3.

2.2.3 Image fusion

In contrast to single-image sharpening, another group of methods utilizes available auxiliary information, such as an additional image with higher spatial reso-

lution, but with different spectral information, usually a lower spectral resolution. A very common approach is the fusion of an MSI and a panchromatic (PAN) image, known as pan-sharpening. The PAN image is a single-band grayscale image encapsulating information from the three visible bands of an RGB image [17].

Different authors such as He et al., Masi et al. and Yuan et al. further classify conventional pan-sharpening techniques into component substitution (CS)-based and multiresolution analysis (MRA)-based methods [11], [24], [40]. CS-based methods transform the MSI into a suitable domain, replace the component containing the MSIs spatial information with the PAN image, up-sample the other components and then re-transform the MSI into the original domain [24], [40]. The fusion result is highly dependent on the correlation between the MS component and the PAN image [40] and often suffers from high spectral distortion [11],[24]. Instead of replacing components, details extracted from the PAN image are directly injected into the upsampled MSI by MRA-based methods [24], [40]. The basic principle of attaining the high-frequency details of the PAN image is to compute the difference between the PAN image and a low-pass filtered version of itself [24]. MRA refers to the collection of these details at different resolution levels of the PAN image [24], [40], e.g. with a Laplacian pyramid [1]. While MRA-based methods lead to less spectral distortions than CS-based methods, they suffer from spatial distortions, e.g. ringing artifacts, which depend on the quality of the co-registration of the MSI and the PAN image [24], [40].

Some authors also discuss model-based optimization (MBO) approaches, which further reduce spectral distortions, but rely on prior knowledge and lack robustness to images with different distributions and quality degradations [40]. Furthermore, inefficient iterative computations are required to minimize a cost function that optimizes the observation model between the ideal and input images [11]. Dictionary-based pan-sharpening methods rely on either a spectral or a spatial dictionary, and therefore cannot effectively preserve both types of high-quality information in the fused result [25].

Analogous to the latest research developments in the field of single-image super-resolution, deep machine learning methods are currently of great interest in the field of image fusion. Deep learning models overcome the limitations of the previously mentioned techniques, as their multiple layers and parameters allow for highly nonlinear transformations. These parameters are updated under the supervision of multiple training samples which reduces the amount of prior knowledge required and allows the model to learn transformations for different distributed inputs.

Recently lots of different architectures of CNNs for pan-sharpening have been put forward. Most of the authors make use of existing architectures and extend them with various modifications to overcome a specific limitation or to fit an explicit use case. The pan-sharpening CNN proposed by Masi et al. is a frequently cited work with fundamental ideas [24]. Wei et al. utilize residual learning as main improvement in their proposed CNN [37] and Yuan et al. introduce multiscale feature extraction [40]. Using these methods, Brook et al. develop a CNN architec-

ture that sharpens Sentinel-2 imagery with high-resolution drone data and prove the accuracy of this concept with in vivo plant measurements from a vineyard [3]. The basic idea of pan-sharpening can also be used to sharpen hyperspectral data by fusion with multispectral data. He et al. make use of 3D convolutions and detail injection layers to fuse the input images and propose a spectral-aware pan-sharpening neural network [11]. Lu et al. develop a coupled CNN where two CNN branches learn details from HS and MS images individually and are fused together in an additional convolutional layer [23]. These authors mostly compare their proposed CNN architectures with conventional sharpening methods and show that they achieve better spatial accuracy. Lu et al. and He et al. also demonstrate results with high quality in spectral fidelity for the sharpening of HSIs [11], [23].

Image fusion techniques are particularly well-suited to the task of hyperspectral image super-resolution, since low-resolution HSIs already contain the majority of spectral information of the targeted ground truth. High-resolution MSIs are employed to obtain the absent high-frequency spatial information, such as edges, and facilitate the reconstruction of high-quality results [23], [39]. In comparison to pan-sharpening techniques, Wei et al. describe the single image super-resolution process as highly ill-posed, as the blind prediction from a low-resolution input to a high-resolution image results in a significant loss of information [37]. The main drawback of image fusion methods is the need for an auxiliary corresponding image in addition to the given HSI. Since multispectral Sentinel-2 data has a high availability and is easily accessible, this work focuses on the development of an image fusion CNN that aims to sharpen EnMAP HSIs with the help of Sentinel-2 MSIs.

2.3 Convolutional Neural Networks

Machine learning has recently become very popular for many image processing tasks such as super-resolution, but also for e.g. object detection or classification. The most common type of deep neural networks used for image processing are Convolutional Neural Networks.

A CNN usually consists of multiple layers that sequentially process an input and pass the computed output to the next layer. The most important type of layer for this work is the convolutional layer, that performs a convolution of a given input matrix with a filter kernel, as already described in Section 2.2.1. For an image processing CNN the network input is an image, given as a three-dimensional matrix with a height, width, and number of bands. Depending on the filter kernel matrix, the first convolutional layer performs a convolution of the input image and its kernel. This can also be described as filtering for features, since the output of a convolutional layer is called a feature map. A filter kernel in a neural network is also described as weights of a layer W_l and the output of a layer Y_l is calculated as formulated by He et al. [11, Formula 2], where ϕ is an activation function and

B_l is a bias matrix:

$$Y_l = \phi(W_l * X_l + B_l) \tag{2.3}$$

The activation function calculates the output of a layer based on its input and parameters. This can be an identity function or a nonlinear function used to increase the nonlinearity of the entire network or for other purposes, such as reducing vanishing gradients. The weights, i.e. the filter kernel, and the bias matrix are the parameters P that are learned during the training process. This work focuses on a supervised learning approach, like most of the existing CNNs for image super-resolution. This means that during training a degraded input image X and a sharp version Y of that image is present. After processing the input image through multiple layers, the generated output image $\hat{Y}$ is compared to Y. In this way, a current error E of the network can be determined:

$$E = Y - \hat{Y} \tag{2.4}$$

In the training phase, the weights and biases of the network are adjusted with the goal of minimizing this error. An often used strategy to determine how to update the parameters, called the optimizer, is gradient descent [38]. The loss or error E of the network contributes to the parameters of a layer P_l at the next time step $t+1$ as follows:

$$(P_l)^{t+1} = (P_l)^t - \eta \frac{\partial E}{\partial (P_l)^t} \tag{2.5}$$

The partial derivative $\partial E / \partial P_l$, known as the gradient, indicates how E changes with respect to P_l. Moving P_l in the gradient's direction increases E, so to minimize the loss function, P_l should be updated in the opposite direction. To prevent a too large step in the negative direction of the gradient, which would also increase the loss, a learning rate parameter $0 < \eta < 1$ is introduced. Furthermore, the error is propagated backwards through the network, where each layer's contribution to the network loss is calculated. During learning, instead of a single training example, consisting of X and Y, a batch of multiple examples is presented to the network at once, to achieve a more stable loss function. After each batch, the gradient is calculated and the network parameters are updated. Typically, the mini-batch strategy, with batches containing a small subset of the entire training dataset, and stochastic gradient descent are used to reduce computation complexity. The term epoch is used to describe the process, where all training examples are presented to the network once [38].

With this procedure, the network learns which filters to use in each layer to extract the features that help to minimize the overall network error and produce the best output. Each layer can utilize more than one filter, with each filter corresponding to a feature map produced by that layer. The number of filters per layer, as well as the filter size, is a fixed given number that is determined before the training.

In contrast to other image processing networks, such as those for categorization or object detection, pooling or fully connected layers are generally avoided in image

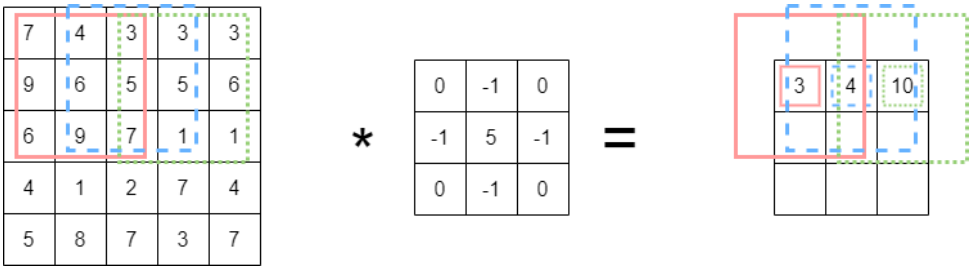


Figure 2.1: Convolution with a 3×3 filter kernel

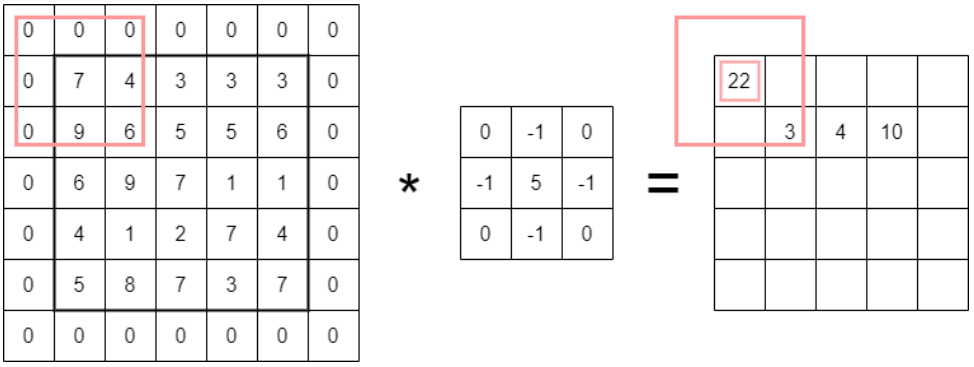


Figure 2.2: Convolution with a 3×3 filter kernel and zero padding

super-resolution and reconstruction, as it is crucial to preserve both the input image shape and the full amount of input information without compression. To ensure that an image has the same shape after a convolution is performed, stride and padding must be introduced. Convolving an input matrix with a filter kernel that is larger than 1×1 always results in a matrix with smaller dimensions. Figure 2.1 outlines that the convolution of a 5×5 input matrix with a common 3×3 filter kernel results in a 3×3 matrix. The filter kernel moves over the input matrix like a window, which is highlighted for the first row. Padding means to artificially enlarge the input matrix to receive a result with the original shape. This is most commonly done by adding 0-values around the edges of the input as shown in Figure 2.2. Since this can result in skewed values, there are alternative strategies, such as reflection padding, that may be employed to circumvent this issue. The second parameter stride s is the step size, that determines that a convolution is performed every s pixels in the horizontal and afterwards vertical direction [38]. In Figure 2.1 a stride of $s = 1$ is present.

The architecture of a deep CNN has many parameters that must be chosen before the training process starts, such as the number of layers, activation functions, the size and number of filter kernels per layer, the optimizer, the loss function, and many more. Some parameters are known to work well for specific applications, others may perform better or worse in different combinations without prior knowledge and could be determined by hyperparameter search. The design and implementation of the CNN architecture proposed in this work is further described in Chapters 4 and 5.

2.4 Reconstruction formulation

The super-resolution or image reconstruction problem addressed in this work can be formulated as a reconstruction g_r that attempts to approximate the ground truth G from a remote sensing observation f_i. This is done by a highly nonlinear function learned by a Convolutional Neural Network $CNN(f_i)$, that transforms the degraded input. The low-resolution hyperspectral observation f_{hs} is complemented by a high-resolution multispectral observation F_{ms} and used as the network input $X = (f_{hs}, F_{ms})$. In light of the above considerations, the problem aimed to be solved in this work is defined as follows:

$$CNN(f_{hs}, F_{ms}) = g_r \approx G \tag{2.6}$$

3 State of the Art

3.1 Fundamental architecture aspects

3.1.1 Image fusion CNNs in remote sensing

As shown in Chapter 2, CNNs have recently gained significant traction in the field of super-resolution, with research also exploring their potential in hyperspectral super-resolution. In particular, the work of Brook et al. demonstrates, that image fusion CNNs can effectively enhance the resolution of remote sensing data, which can then be used for further analysis, such as the observation of plant responses [3]. With independent high resolution multispectral images and in situ measurements, they were able to demonstrate that the fusion of PAN and MSIs with their trained network achieves high spatial and spectral resolution [3].

A frequently cited method is the pan-sharpening CNN by Masi et al. that uses three convolutional layers [24]. This is derived from Dong et al., who conducted a comparative analysis of a super-resolution architecture with deeper architectures and concluded that they do not exhibit superior performance compared to an architecture with three layers. Instead, the complexity and training time increase significantly [7]. A fundamental common ground among super-resolution CNNs is the fixed input size, which is a parameter that cannot be altered in a trained model. To satisfy this requirement and to reduce computational complexity, the input image must be divided into multiple tiles of fixed shape. Masi et al. describe that a low-resolution multispectral image is upscaled and interpolated to the same shape as the PAN image. Both images are afterwards stacked, tiled, and fed into the network, which thus operates at the target resolution and aims only to reconstruct the ground truth image from the interpolated input [24]. The stacked images, which have the same height h and width h, are represented as a matrix with a dimension of $h \times w \times b_i$. The variable b_i represents the number of bands present in the multispectral image b and the additional PAN band. In the architecture proposed by Masi et al., this matrix gets passed to the first convolutional layer, which has 64 filter kernels with a dimension of 9×9 and is activated by a ReLU function. The output of this layer has a dimensionality of $h \times w \times f_1$, where f_1 denotes the number of feature maps produced by the filter kernels, which is equal to 64. This output is passed to the next layer, which performs convolutions with $f_2 = 32$ filter kernels of dimension 5×5 and has a ReLU activation function. The final convolutional layer uses the resulting output and performs convolutions with $f_3 = 4$ filter kernels, which have a shape of 5×5. This is also the output layer of the network. An identity activation function is used and the number of

feature maps f_3 is equal to the number of bands of the multispectral input image b. Therefore, the output matrix, i.e., the reconstructed high-resolution image, has a dimension of $h \times w \times f_o$ with $f_o = b$ [24].

Inspired by the network proposed by Masi et al., Wei et al. introduce residual learning and a deeper architecture for the pan-sharpening task [37]. The extension of a network by several layers increases nonlinearity and can help to extract more features and increase accuracy, but also increases the risk of vanishing gradients during the backpropagation learning process. Residual learning directly addresses this problem as the networks hidden layers learn the mapping between the degraded input f_i and the difference of the input and ground truth G, called the residual $f_{res} \approx G - f_i$. Before the output layer a skip connection adds the input to the residual and the reconstructed ground truth image g_r is formed $f_i + f_{res} = g_r \approx G$. Most of the values in the residual are close to zero, which leads to a faster learning process, a smoother loss hypersurface and enables the use of more hidden layers [37]. Yuan et al. and Brook et al. utilize the architecture by Masi et al. for a shallow network branch and combine it with a deeper branch with skip connections and feature extraction blocks [3], [40].

These basic architectural concepts of super-resolution CNNs are used in the most proposed networks. Since the performance alters for different use cases, such as input data from different sensors with different spectral resolution, there is no existing superior combination of parameters. Therefore a hyperparameter search is an effective method to identify the optimal configuration. For example, Wei et al. train three models each with a different shape of filter kernels on the same training dataset. The models are then compared on two evaluation metrics to determine the best filter kernel shape [37].

3.1.2 Spectral context in 3D convolutions

In the majority of Convolutional Neural Networks, the filter kernels are two-dimensional matrices with a fixed height and width. In the context of multi- and hyperspectral image reconstruction, the input images are represented as three-dimensional matrices, since the third dimension is defined by the number of bands b. From the perspective of pixels arranged in a coordinate system, each pixel has an x- and a y-coordinate and b number of values. Filtering this image with a two-dimensional filter matrix is done by the consideration of each band as an individual two-dimensional matrix. The process can be thought of as a kernel sliding over a band, producing a filtered result band, before moving on to the next one. For a kernel with a size $h \times w$ greater than 1×1 and a stride $s < h$ and/or $s < w$, the filtering process considers spatial relationships as the neighbors of a pixel are taken into account for each step of the convolution. When the sliding kernel moves to the next band, no pixel values from other bands are considered and therefore possible relationships between adjacent bands are ignored.

To utilize spectral context in the filtering process, three-dimensional filter kernels are introduced. 3D convolutions proved useful for exploring volumetric data

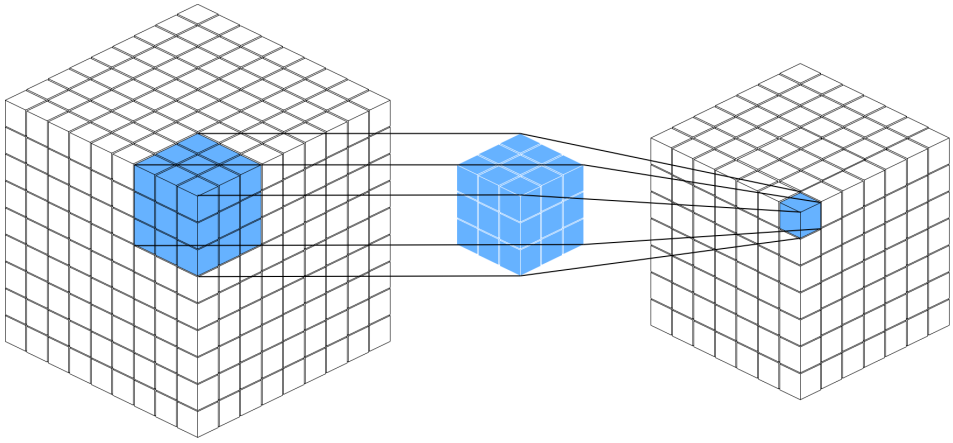


Figure 3.1: Convolution with a three-dimensional filter kernel

needed for tasks such as object detection in virtual 3D models [29]. Hyperspectral image data can also be interpreted as volumetric data, where the three dimensions of height, width, and number of bands correspond to an image. Using a three-dimensional filter kernel with a shape of $3 \times 3 \times 3$ enables the calculation of the filtered value of a pixel, derived from 27 input values. As illustrated in Figure 3.1, a 3×3 neighborhood of a given pixel in its corresponding band and in the two adjacent bands is investigated. The concepts of padding and stride remain unchanged. Compared to a convolutional layer using 2D convolutions, the output matrix of a 3D layer has a shape of $h \times w \times b \times f_l$. Rather than each feature map representing a single band of an image, the number of bands b of the input image remains constant throughout the network. Accordingly, each band can have multiple feature maps, corresponding to the number of filter kernels f_l used in a layer. For super-resolution tasks, the last convolutional layer should usually generate a single feature map, so that the output has a shape of $h \times w \times b$ and is identical to the input shape.

Mei et al. propose a CNN architecture, called 3D-FCNN, for hyperspectral single-image super-resolution that is similar to the SRCNN architecture from Dong et al. [7], but uses three-dimensional filter kernels [25]. In comparison with msiSRCNN, an extended version of SRCNN for multispectral imagery [19], 3D-FCNN shows better reconstruction results and less spectral distortion [25]. Subsequently, Li et al. propose an architecture with a mixture of 2D and 3D convolutional layers, called MCNet, which surpasses the performance of 3D-FCNN [18]. The F3DUN network by Liu et al. in turn only uses 3D convolutions and obtains better results than a mixed model with the same parameters and other models such as 3D-FCNN and MCNet [21]. The spectral-aware pansharpening neural network (SA-PNN) by He et al. uses two branches to achieve super-resolution for multispectral images [11]. The detail branch focuses on the extraction of spatial detail components on a stacked input of a high-resolution PAN image and a low-resolution multispectral image. In this branch, only 2D convolutions are used, while the approximation branch uses 3D convolutions. For the approximation branch, only the multispectral image is used as an input. After each layer the extracted features from the

detail branch are injected into the approximation branch. This is done by spatial feature transformations (SFTs), which adjust the spectra of a feature map present in the approximation branch. To achieve this, parameter pairs (α, β) are obtained from the feature maps of the current detail branch layer and used to scale and shift the feature maps of the corresponding approximation branch layer [11, Formula 3]:

$$SFT(F|\alpha, \beta) = \alpha \odot F + \beta \tag{3.1}$$

He et al. show that their proposed architecture achieves better results than the networks proposed by Masi et al. and Wei et al. and that the SA-PNN significantly reduces spectral distortion. The model was also evaluated on an HSI dataset in combination with a PAN image, which was simulated using the HSI bands that fall within the visible spectrum [11].

The mentioned works show that the use of spectral context in the filtering process can improve the super-resolution results for multi- and hyperspectral data and prevent spectral distortions. The main disadvantage of employing 3D rather than 2D filter kernels is the increased complexity and resource consumption resulting from the generation of four-dimensional matrices by each 3D convolutional layer, as opposed to three-dimensional matrices by layers with 2D kernels.

3.1.3 Wald's protocol

As described in Section 2.3 and Equation 2.4 during a supervised learning process, the parameters of a CNN are adopted to minimize the error between the network output $\hat{Y}$ and the reference data Y. In the context of super-resolution, the network output is a scientific image g_r that reconstructs the ground truth G from a degraded observation as described in Equation 2.6. However, a ground truth image does not exist and therefore no reference data for supervised learning is available. To overcome this, the Wald protocol, originally described by Wald et al. for the assessment of fused satellite images [34], is a frequently utilized strategy. Initially, the high-resolution and low-resolution input components are downsampled to a lower resolution. This is done by the ratio of the desired sharpening. The low-resolution components are then upsampled and interpolated to match the scale of the high-resolution components. Both components are afterwards stacked and fed into the network as the degraded input image. The original low-resolution components now have a higher resolution on the same scale as the input components and can be used as the reference Y_{ref}. In this way, the network learns its parameters on a different scale and can later be used to sharpen the original images. An example of the Wald protocol is illustrated in Figure 3.2 and Figure 3.3 depicts the network training process with an example scene from the EnMAP satellite and a corresponding scene from Sentinel-2. The network learns the sharpening of a degraded hyperspectral image f_{hs} with a size of $3 \times 3 \times b_{hs}$ and an auxiliary multispectral image F_{ms} using the original hyperspectral image as a reference. Afterwards the trained model can be used to reconstruct a sharp hyperspectral image from an upsampled and interpolated low-resolution hyper-

spectral image of size $9 \times 9 \times b_{hs}$ and an auxiliary high-resolution multispectral image of the same size.

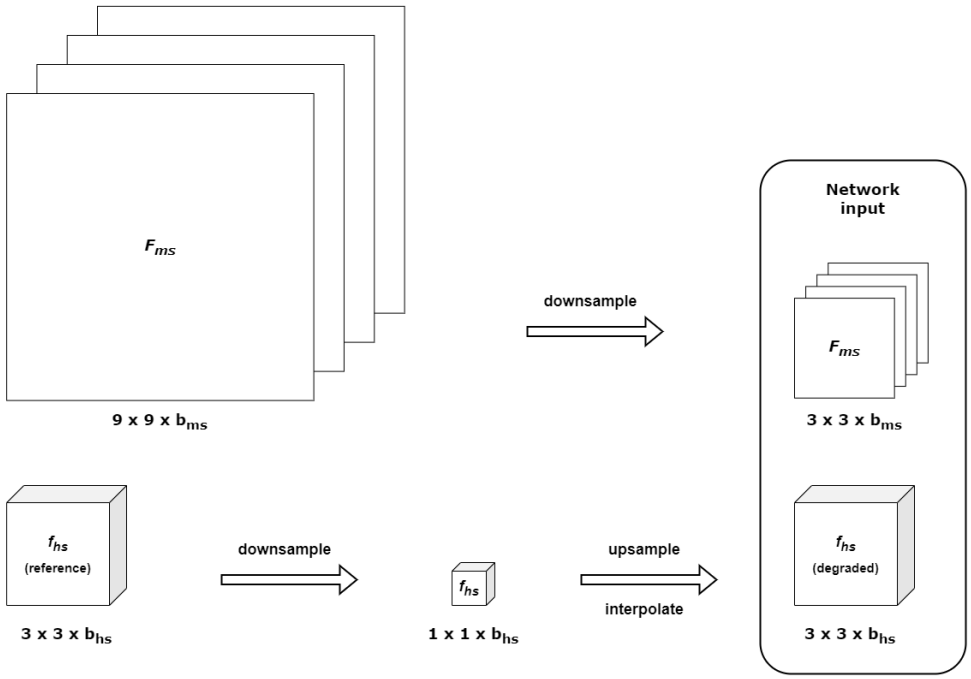


Figure 3.2: Wald protocol with high-resolution multispectral and low-resolution hyperspectral components

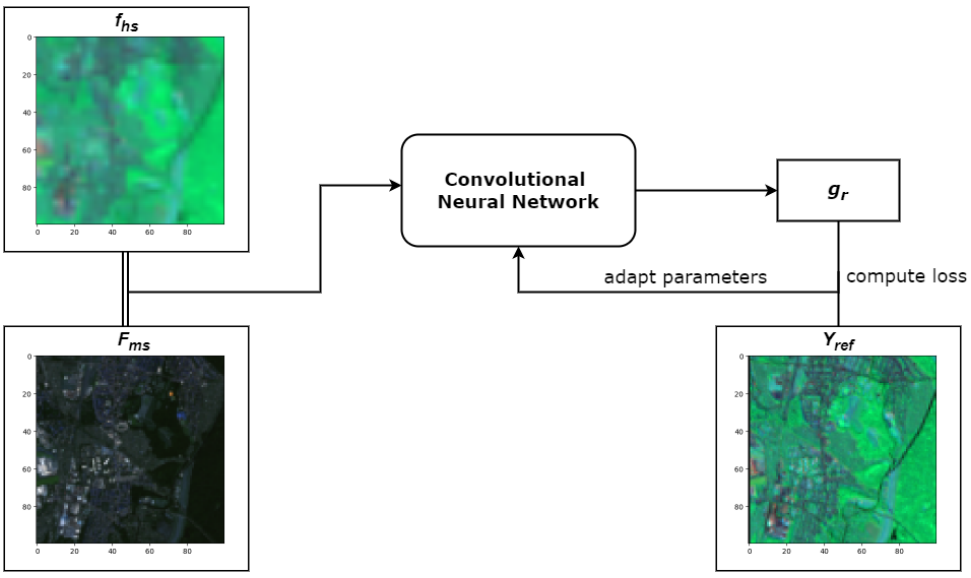


Figure 3.3: Network training with inputs and reference simulated by the Wald protocol

It is common practice in the field of super-resolution with supervised CNNs to use the Wald protocol for both training and assessment of the network. Some authors refer to their strategy as the Wald protocol [11], [24], while others do not use a name but employ a similar approach. For instance, [23], [25], [37], [39], [40].

3.2 Current limitations and resulting objectives

3.2.1 Comparison of recent architectures

For a better overview, the CNN architectures mentioned so far for multi- and hyperspectral image super-resolution are shown in Table 3.1. As mentioned in Section 2.2.2, single image super-resolution is a highly ill-posed process, and with Sentinel-2 imagery available as auxiliary data, this work focuses on an image fusion approach to spatially sharpen the resolution of hyperspectral EnMAP data. In Section 3.1.2, three-dimensional convolutions were introduced and shown to be useful for exploring spectral context. In particular, the SA-PNN approach of He et al. is very interesting for this work, as it follows the idea of obtaining and injecting spatial details from the auxiliary image while performing a spectral-aware reconstruction of the hyperspectral image with 3D convolutions [11].

SR category	Target	2D convolutions	Mixed	3D convolutions
Single-image	MSI	SRCNN [7], VDSR [15], msiSRCNN [19]		
	HSI		MCNet [18], Interact-former [20]	3D-FCNN [25], F3DUN [21]
Image fusion	MSI	PNN [24], DRPNN [37], MSDCNN [40], multiscale CNN [3]	SA-PNN [11]	
	HSI	Two-CNN-Fu [39], CpCNN [23]	SA-PNN [11]	

Table 3.1: CNN architectures for multi- and hyperspectral image super-resolution

3.2.2 Limitations of existing approaches

Despite the promising results it should be noted, however, that 3D convolutions are more complex and have a higher resource consumption. In addition, the SA-PNN uses four more 2D convolutions and additional transformations of each band of each feature map per SFT operation. The model was also tested on a 103-band hyperspectral dataset with reduced dimensions [11]. However, EnMAP images consist of 224 bands, resulting in higher resource consumption. Considering this, a more efficient approach may be necessary.

Another consideration is that the common methodology for most remote sensing super-resolution networks is to train and evaluate a model on only one or a few selected scenes from a specific region. A frequently used dataset was acquired with a ROSIS sensor during a flight campaign over the Italian city of Pavia and contains two scenes. The first scene captures the city center and has a spectral resolution of 102 bands, while the second scene captures the University of Pavia with 103 bands. At least one of these scenes has been used for the architectures SA-PNN [11], Interactformer [20], CpCNN [23] and 3D-FCNN [25]. CpCNN [23] and 3D-FCNN [25] also used a scene that captures the location around a mall in Washington DC, USA, has a spectral resolution of 191 bands and was captured with a HYDICE sensor. Instead of existing auxiliary data, a high-resolution PAN image was simulated for training of the SA-PNN by averaging the visible spectral bands of the HSI. The Two-CNN-Fu was trained once on four scenes with 162 bands from an Airborne Visible Infrared Imaging Spectrometer (AVIRIS) sensor, and once on a scene from a HyMap sensor with 244 bands [39]. In order to simulate the high-resolution multispectral image, the HSI bands were averaged with wavelengths that are covered by corresponding bands of the multispectral Landsat 5 TM sensor. Both models were tested on tiles from the respective scene, which were withheld from the training set. Afterwards another model was trained on a hyperspectral spaceborne scene from a Hyperion sensor carried by the Earth Observing-1 satellite and a corresponding multispectral Sentinel-2A scene. Due to noise, only 83 of the 242 bands captured by the Hyperion sensor were taken into account. MCNet [18] and F3DUN [21] have not been tested on remote sensing data. All the mentioned datasets have been selected with particular care and only partially reflect real remote sensing data.

3.2.3 Development goals

The goal of this work is to train a model that sharpens EnMAP data without imposing additional constraints. Consequently, the model must be capable of processing scenes from different locations, at varying times of the day and year, and must be able to handle noise introduced by atmospheric conditions such as clouds and haze, as well as spatial and temporal discrepancies between hyperspectral and auxiliary multispectral data. To ensure that the model has a good ability to generalize, a framework must be developed that fetches and preprocesses EnMAP and corresponding Sentinel-2 scenes to create an appropriate training dataset. This tools should also be capable of fetching and preprocessing data, which can be fed into the trained model. The model aims to reconstruct sharp EnMAP data with a resolution of 10 m $\times$ 10 m per pixel, according to the resolution of the Sentinel-2 data used. Moreover, the sharpening process should not result in any alteration to the dimensionality of the inputs. Additionally, the outputs should be georeferenced data. To ensure straightforward applicability, expandability, and comparability, it is essential to maintain all tools in a public repository.

4 Design and Methods

4.1 EnMAP data collection

To train a super-resolution model that has reasonable performance and the ability to generalize well to unknown data, a suitable dataset must be available. This should consist of enough unbiased samples to achieve convergence during training without overfitting. A collection of EnMAP L2A data, which is regularly updated with new data, is provided by the DLR[1]. Bachmann et al. illustrate the design of the analysis-ready L2A data product, which describes the composition of the spectral raster file as well as available auxiliary files. The latter are, e.g. quicklook images, binary masks, or classification files containing pixel flags for the classes “Land”, “Water”, “Background” and “Other” [2].

To efficiently build a training dataset, a data scraping tool was developed that automatically requests and downloads EnMAP scenes from the database. Initially, a *GET* request is sent to the available API, to receive a filtered list of items that match given parameters. Possible parameters are a maximum cloud cover and a spatial bounding box that should not be exceeded, as well as a specified time period within which the requested scenes should be recorded. The maximum cloud cover value refers to the sum of the cloud and cirrus cover values as specified in the metadata associated with the given scene. Subsequently, a separate *GET* request is initiated for each item on the list, with the objective of obtaining the corresponding spectral raster file, the associated metadata file, and the binary cloud mask and cloud shadow mask files. If the item in question has valid properties and all the necessary files are present in the database, the scenes are downloaded to a local device one at a time. In the spectral raster file, each pixel has a value per band corresponding to the reflectance measured at that coordinate. The data type is *int16*, the value -32 768 marks a pixel with no data, and values between 0 and 10 000 correspond to 0 to 100% reflectance on a linear scale. A detailed view of the obtained and used datasets is given in Chapter 5.

[1] `https://geoservice.dlr.de/eoc/ogc/stac/v1/collections/ENMAP_HSI_L2A`

4.2 Data preprocessing pipeline

4.2.1 Overview

In order to assemble and process a representative dataset that can be used for model training, a series of preprocessing steps on the initial dataset must be carried out. Therefore, a four-stage pipeline was designed to generate an input dataset from given EnMAP data. Each scene is sequentially passed through each stage of the pipeline and ends up as tiled model input data according to the Wald protocol, annotated with auxiliary Sentinel-2 information. The fact that the EnMAP and Sentinel-2 scenes may have been acquired at different times is a major challenge for the image fusion approach. This is because additional noise is introduced between the two scenes due to changing scenery, changing reflective properties such as fresh snow cover, or more commonly, small georeferencing errors, different shadows due to different times of day, or different cloud cover. The latter can be completely different, even if the time difference is only a few minutes. To reduce the differences between the two scenes, cloud masks from both scenes are merged in Stage 2 and applied to both rasters, and georeferencing errors are targeted in Stage 4. The pipeline can be reused for different EnMAP datasets to train different models, e.g. for specific tasks.

Figure 4.1 provides an overview of the preprocessing pipeline and depicts the output of each stage. The process begins with an initial EnMAP raster containing spectral information and a corresponding binary cloud mask. Based on these inputs, the model training inputs, designated as X and Y, are generated, with Y serving as the training reference.

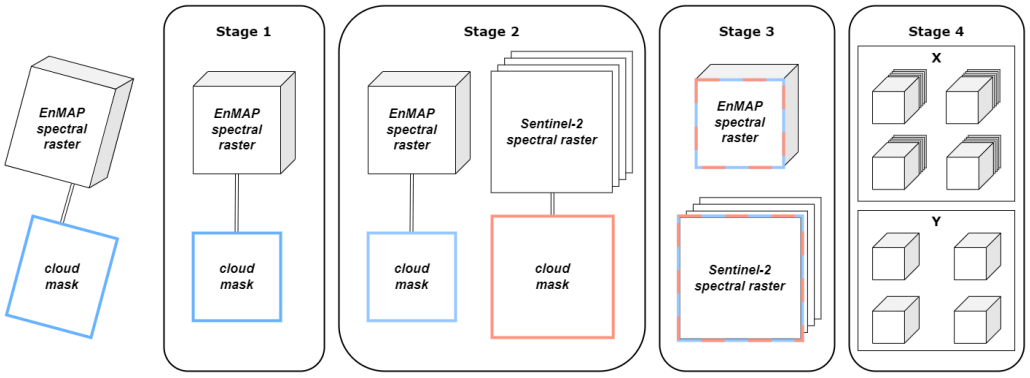


Figure 4.1: Overview of the preprocessing pipeline

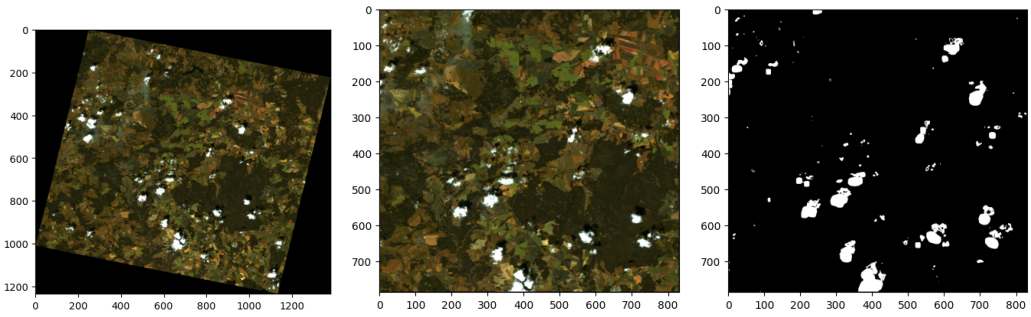
4.2.2 Stage 1: cropping the EnMAP scene

The first preprocessing step is to crop the EnMAP data to obtain rectangular rasters with axis-parallel edges. This is necessary because scenes have different shapes depending on the latitude at which they were acquired. Cropping each scene to a

rectangular shape without rotation simplifies co-registration with the corresponding Sentinel-2 scenes and subsequent tiling. The bounding box coordinates with longitude and latitude pairs of the spatial coverage of the data can be extracted from the metadata files. To match the coordinate system of the raster files, the coordinates must be transformed to UTM format. For clockwise rotated scenes, the upper left corner x-coordinate O_{ul_x} and the upper right corner y-coordinate O_{ur_y} mark the upper left corner coordinates of the new bounding box. The coordinates R_{ul}, R_{ur}, R_{lr}, R_{ll} of the new rectangle are composed of the original bounding box coordinates O as follows:

$$R_{ul} = (O_{ul_x}, O_{ul_x}), R_{ur} = (O_{lr_x}, O_{ur_y}), R_{lr} = (O_{lr_x}, O_{ll_y}), R_{ll} = (O_{ul_x}, O_{ll_y}) \quad (4.1)$$

An example scene from the training dataset is shown in Figure 4.1. It was recorded on June 24, 2023 at 10:32:15 UTC+0 in the western part of the Czech Republic. According to the metadata, the biom is described as temperate broadleaf and mixed forests. For visualization purposes, three bands with center wavelengths of ~435 nm, ~545 nm, and ~700 nm were selected, scaled, and high reflectance values discarded to increase brightness and simulate an RGB image. The axis scales indicate the number of pixels. According to the resolution of EnMAP data, each pixel is 30 meters long and 30 meters wide. Subfigure (a) shows a visualization of the original spectral file and (b) shows the cropped rectangular raster. The same cropping is applied to the binary cloud mask. This file indicates for each pixel in the raster whether it is covered by a cloud or not. Before cropping, the cloud mask is merged with the corresponding cloud shadow mask into a single file. Subfigure (c) shows the resulting cropped mask with white spots indicating a cloud or cloud shadow.



(a) Initial EnMAP raster **(b)** Cropped raster **(c)** Cropped cloud mask

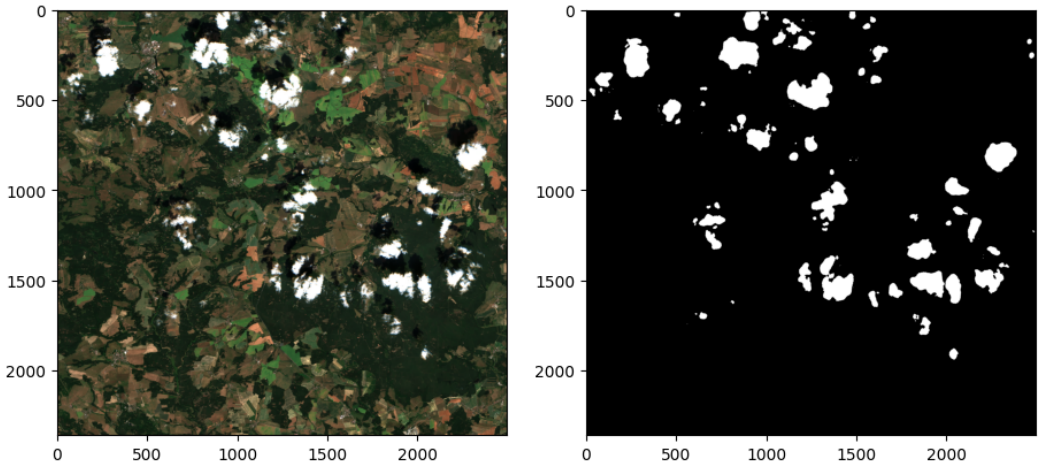
Figure 4.2: Example EnMAP scene before and after Stage 1

4.2.3 Stage 2: fetching Sentinel-2 data

The second stage of the pipeline was designed to download multispectral Sentinel-2 scenes with higher spatial resolution, which will later be used as auxiliary in-

formation in the image fusion process with the corresponding EnMAP scenes. To achieve this, the Sentinel Hub Process API[2] was utilized. After a registration process, 30 000 free requests or processing units are granted per user account per month. To receive a suitable scene for a given EnMAP raster, a *POST* request with certain parameters is sent to the Process API. The parameters consist of the bounding box coordinates and coordinate reference system from the cropped EnMAP raster and a time range of 15 days before and after the time and date the EnMAP scene was recorded. With a Sentinel-2 revisit time of five days in almost all regions [32], it is guaranteed that at least one, but most likely more, scenes are recorded by the Sentinel-2 satellites in the given time period. If multiple scenes are found with the given bounding box and time range parameters, another parameter defines that the scene with the least cloud cover should be selected. The request further specifies that the spectral raster included in the response should include only bands 2, 3, 4, and 8 of the Sentinel-2 imager, as these bands have a spatial resolution of 10 m $\times$ 10 m per pixel [32]. Since the Process API limits the requested raster size to 2 500 $\times$ 2 500 pixels, the cropped EnMAP scene must cover a maximum surface area of less than 25 km^2, which was considered in the design of Stage 1. The datatype of each value in the raster from the response is *uint16*. Values between 0 and 10 000 correspond to reflectance values between 0 and 100% on a linear scale, which is equivalent to EnMAP reflectance values. In addition to the spectral raster file, a binary cloud mask file is requested.

Figure 4.3 shows the spectral raster and the binary cloud mask retrieved from the Sentinel Hub Process API based on the procedure described and the initial EnMAP raster shown in Figure 4.2. For visualization purposes, three bands with center wavelengths of ~490 nm, ~560 nm, and ~665 nm were selected and brightened.



(a) Sentinel-2 spectral raster **(b)** Binary cloud mask

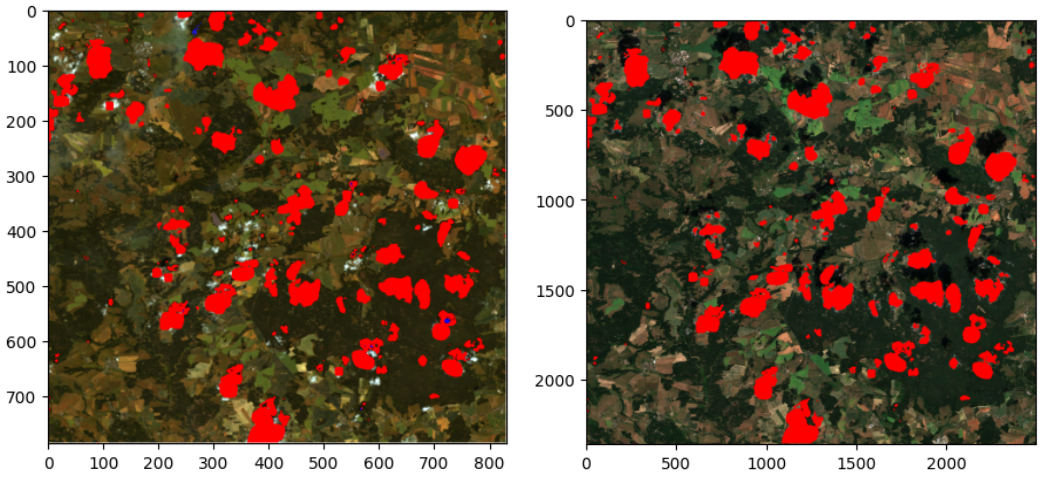
Figure 4.3: Sentinel-2 scene corresponding to the EnMAP scene shown in Figure 4.2

[2] https://docs.sentinel-hub.com/api/latest/reference/

4.2.4 Stage 3: masking clouds

To reduce noise between the EnMAP and Sentinel-2 raster pairs caused by clouds, Stage 3 was designed to harmonize all regions that are covered by clouds in each of the two rasters. Otherwise, the differences between the two images may be too large to form a useful pair for the learning process. The first step of harmonization is to merge the two binary cloud masks into a single mask. Each pixel value then indicates whether or not a pixel with the same coordinates is covered by a cloud in the Sentinel-2 image or by a cloud or cloud shadow in the EnMAP image. Due to the three times higher resolution of Sentinel-2, the EnMAP cloud mask file is upsampled and interpolated to the same size and resolution using a bilinear interpolator. The upsampled mask is then merged with the Sentinel-2 cloud mask, and the result is afterwards downsampled to the original EnMAP resolution. Therefore two merged masks in both resolutions are created. Each pixel in each raster is then set to zero if the corresponding mask at the same coordinates indicates cloud cover. This way, both rasters imply that there is no reflectance at those coordinates, and differences due to different cloud cover are largely neutralized. Using a negative or positive value would introduce unmeasured reflectance and could lead to spectral distortions. At this point, all raster values are also clipped between 0 and 10 0000. This is done to remove misleading values, such as the negative value indicating no available data in EnMAP rasters, and to slightly reduce the dimensionality to a range between 0 and 100% reflectance. Values outside this range may occur, for example, due to bundled reflections from reflective surfaces, but are not useful for further analysis and may unintentionally distort the data. This particular scaling of features is conducted at this stage, as it is necessary for all values of each band to be read into memory for masking purposes. Other processing steps, including resampling and cropping, can be performed effectively without the necessity of loading all values into memory.

Figure 4.4 depicts the example EnMAP and the Sentinel-2 rasters with the merged masks applied. For better visualization, masked pixels were colored red. It is recognizable that most clouds from both rasters are covered. However, some clouds are still visible. This may be due to the fact that the threshold for defining an existing cloud was not exceeded when the mask was created [2]. Other possibilities are that the visible white spots are classified as haze or thin cirrus clouds, which also do not exceed the threshold value. In order to avoid excessive modification of the data, these small deviations are not taken into account. In addition, cirrus clouds are already considered during data collection as described in Section 4.1 and should therefore not appear to a large extent in the EnMAP data. It can also be seen that cloud shadows are not masked in the Sentinel-2 scene, as there is no cloud mask file available. This should not be a major problem, as it is mainly high frequency components such as edges that should be extracted from the Sentinel-2 data, and these are still present even when covered by a cloud shadow.



(a) Masked EnMAP raster

(b) Masked Sentinel-2 raster

Figure 4.4: Spectral raster files masked with merged binary cloud masks

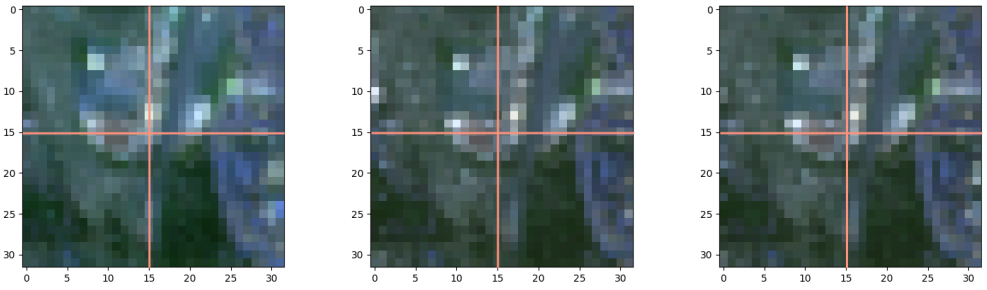
4.2.5 Stage 4: Wald protocol, alignment and tiling

The main purpose of the last stage is to generate model training data according to the Wald protocol described in Section 3.1.3. To do this, the EnMAP and Sentinel-2 raster pairs are first scaled down by a factor of three to resolutions of 90 m $\times$ 90 m and 30 m $\times$ 30 m. Subsequently, each EnMAP raster of a pair is resampled to its original resolution with the use of a bilinear interpolator. This process generates the degraded input and the original reference rasters.

The next step is to correct for any spatial shift between the EnMAP and Sentinel-2 scenes. Due to random and systematic errors in the process of capturing an image with all atmospheric influences to the analysis-ready data product, the assignment of a geolocation in a coordinate reference system to a pixel may have minor discrepancies. The April 2024 Sentinel-2 L1C Product Data Quality Report reports a 5 m error at 95.5% confidence for multiple scenes acquired on different dates for the same geographic area [31]. Figure 4.5 (a) shows an example of an EnMAP scene section. Subfigure (b) shows the corresponding Setinel-2 scene downscaled to the same resolution as (a). The red markers were added for better orientation and a slight shift of the scene is visible. For a better co-registration of both scenes, the Enhanced Correlation Coefficient Maximization method (ECC), as originally proposed by Evangelidis and Psarakis [8] and implemented in the Python package *OpenCV*[3], was utilized in this stage. To calculate the necessary transformation matrix, four bands of the EnMAP image with similar center wavelengths to the four bands of Sentinel-2 are selected. For each selected band of each raster, a gradient is computed using the horizontal and vertical Sobel operators. The ECC algorithm is then applied to each pair of bands. The resulting four translation

[3] `opencv.org`

matrices are averaged into the final matrix used to align the Sentinel-2 raster to the EnMAP raster using a linear transformation. Subfigure (c) shows the aligned scene. The described method does not lead to an absolute co-registration of both scenes. However, the noise between the two scenes introduced by georeferencing errors can be reduced.



(a) EnMAP raster **(b)** Sentinel-2 downscaled **(c)** Sentinel-2 aligned

Figure 4.5: Sentinel-2 raster with georeferencing errors in comparison to EnMAP raster and corrected Sentinel-2 raster

To achieve an efficient tiling process and to facilitate saving and loading data into the model, the aligned and downscaled bands of each Sentinel-2 raster are stacked on top of each corresponding degraded EnMAP raster along the spectral axis and saved as single raster files. Before saving, the combined rasters, as well as the original EnMAP rasters, are cropped according to the previous alignment operation, which may result in missing values near the edges of the Sentinel-2 image due to the geometric translation used.

In the last step, the stacked scenes are tiled into several small patches. This is done since the CNN can only be trained on a fixed input and output size, to increase the amount of training data, and to reduce the number of parameters that need to be trained for faster convergence during training. The patches are unique parts of the scene that do not overlap, which could lead to errors as noted by Lu et al. [23]. Tiles without a minimum ratio of 90% of pixels indicating a reflection are discarded. This usually happens when a tile is almost completely covered by a cloud and the covered pixel values have been set to zero in Stage 3. Instead of discarding an entire scene with a high cloud cover, some parts of the scene can be used in this way. The remaining stacked tiles are used as network input X and the corresponding original hyperspectral patches are used as reference Y during training.

4.3 Network design

The Convolutional Neural Network was designed to spatially sharpen hyperspectral remote sensing data from the EnMAP satellite mission. With respect to the considerations in Chapter 3, this process was guided by commonly used methods from existing approaches. In addition, the objectives of the development were that the model should consider the spectral context, overcome the current existing limitations outlined in Section 3.2.2, and meet the defined goals described in Section 3.2.3.

Based on the the SA-PNN of He et al. [11], the main network architecture was developed with two branches. One branch obtains details using 2D convolutions from the multispectral Sentinel-2 image, while the main branch reconstructs a target resolution image from the degraded EnMAP input image using 3D convolutions and processes injections from the detail branch. To identify an optimal configuration, the model was designed incrementally, with each step involving the adjustment of different parameters. The resulting configurations were evaluated using three evaluation metrics in addition to the loss function. The first two of the metrics used were the Mean Squared Error (MSE), given in Equation 4.2, and the Peak Signal-to-Noise Ratio, given in Equation 4.3, which assesses the quality of a reconstructed signal. MAX_I refers to the maximum value in the original image Y_{ref}. In addition, the structural similarity index measure (SSIM), which evaluates image quality in terms of the degree of degradation of structural information originally introduced by Wang et al. [36], was used.

$$MSE = \frac{1}{h \times w \times b} \sum_{i=0}^{h-1} \sum_{j=0}^{w-1} \sum_{k=0}^{b-1} \left[Y_{ref}(i,j,k) - g_r(i,j,k) \right]^2 \tag{4.2}$$

$$PSNR = 20 \times \log_{10}(MAX_I) - 10 \times \log_{10}(MSE) \tag{4.3}$$

Since the 3D convolutions are performed on five-dimensional matrices with a size of $batch\_size \times h \times w \times b \times f_l$ during the batch strategy learning phase, training a model to convergence is very time consuming. Therefore, many multiple models with different parameters were trained using only 40 bands and compared on a validation dataset. The validation dataset was created prior to training by splitting the initial training dataset by a given ratio, which was set to 80:20, into the actual training set and a small validation set. Unlike training samples, a validation set is not used to compute loss and adjust model parameters. Instead, after each epoch, the model is automatically evaluated on the validation set, which provides a better indication of current performance than the metrics evaluated on the training samples that have already been presented to the model. In this way, different configurations of model parameters were manually set, compared, and retained if they helped improve model performance on the validation set, as assessed by the metrics described above.

Using this strategy, a basic model architecture with two branches was determined. A simplified flowchart of the model is shown in Figure 4.6. The detail branch

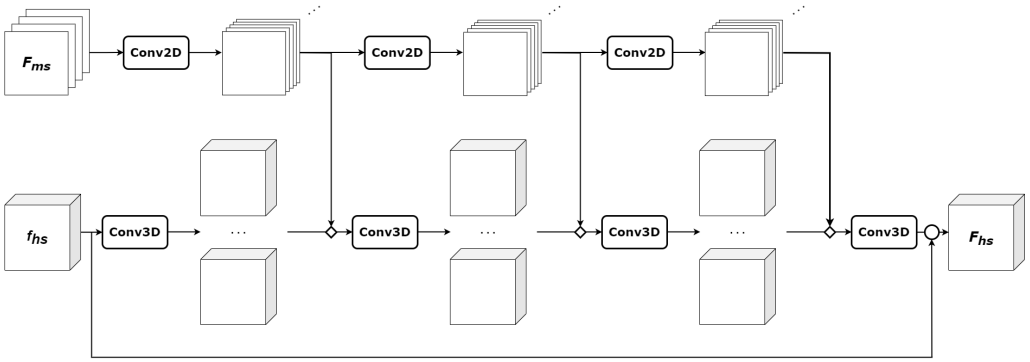


Figure 4.6: Simplified model graph

consists of three convolutional layers that perform 2D convolutions on the high-resolution multispectral input tiles, while the main branch performs three 3D convolutions on the low-resolution hyperspectral input tiles. Each convolutional layer in the detail branch creates f_{l_d} number of feature maps, each with a shape of $h \times w$. To utilize them for the image reconstruction done in the main branch, these feature maps, containing the obtained details from the high-resolution multispectral images, are injected into the main branch. This is done by adding them to the obtained feature maps from the 3D convolution at the same stage, which is illustrated in Figure 4.6 by the diamond symbols. As the feature maps from the detail branch contain mostly high-frequency components that are obtained independently from the spectral context, they are equally added to each band of the hyperspectral image processed in the main branch. Because each convolutional layer in the main branch creates f_{l_m} number of feature maps with a shape of $h \times w$ for each band b, the output shape of a detail injection operation is $h \times w \times (f_{l_m} + f_{l_d}) \times b$. For example, if each band has 64 corresponding feature maps after a 3D convolution, and in the same stage of the detail branch 9 feature maps were created, each band are assigned 73 feature maps after the detail injection and before the next convolutional layer. This method of stacking feature maps has proven successful for detail injection and is less resource intensive than the SFT-operation used in the SA-PNN [11].

In order to speed up network training and to avoid exploding output values, a batch normalization operation, originally introduced by Ioffe and Szegedy [13] and used for example in CpCNN [23], is performed after each convolutional layer in the detail branch. Also, the padding strategy was set to reflection padding only in the detail branch in order to preserve the spatial dimensions of the inputs. As explained by Ullah and Song, other padding strategies than zero padding can perform better in super-resolution CNNs [33]. The use of reflection padding in the detail branch has been shown to improve performance. For the main branch, a simple and less consuming zero padding strategy has been shown to work well.

After three hidden layers in each branch and the final detail injection operation, an additional 3D convolution is performed with a single kernel to match the output dimension of $h \times w \times b$. The activation function for this layer is the identity function, as proposed by Masi et al. [24]. After this layer, the degraded hyperspectral input

image is added to the output by a skip connection to introduce residual learning as described in Section 3.1.1. Instead of the ReLU activation function commonly used in super-resolution CNNs, such as MCNet [18], F3DUN [21], or 3D-FCNN [25], leakyReLU was set as the activation function for each hidden convolutional layer, similar to the SA-PNN model [11]. This way, the dying ReLU problem is avoided and each neuron can contribute to the output of a layer.

To determine all of the above mentioned parameters, a fixed set of filter kernel sizes and number of filters per layer was used. For the main branch, these were based on the parameters used in the 3D-FCNN. Slightly modified, this resulted in filter kernel sizes of $9 \times 9 \times 7$, $3 \times 3 \times 1$, $3 \times 3 \times 1$, $5 \times 5 \times 3$, and 64, 32, 9, 1 number of filters in the main branch. The filter kernel sizes in the detail branch were set to 3×3, as this is a common size for sharpening or edge detection kernels, and three filters were used per layer. Once a well-functioning architecture was found, a hyperparameter search was performed to determine the best shape of the filter kernels and the best number of filters used per convolutional layer for the current configuration. The best hyperparameters found were then evaluated on another data set to further test and compare the generalization capabilities. Once a satisfactory configuration was found, the model was trained using all 224 bands from the EnMAP data. A more detailed view of the parameters used, the hyperparameter search, and the training results is given in Section 5.3. After training and initial evaluation at a resolution of 30 m $\times$ 30 m per pixel, the model was tested and evaluated at the target resolution of 10 m $\times$ 10 m per pixel. This is described in more detail in Chapter 6.

5 Implementation and Experimental Setup

5.1 Preprocessing pipeline

The data preprocessing pipeline described in Section 4.2 was developed using the *Python*[1] programming language in version 3.11.9. For reading, writing, and processing geospatial raster data, such as the EnMAP or Sentinel-2 data, the library *rasterio*[2] in version 1.3.10 was utilized. This package provides tools for performing conventional matrix operations on multi-band rasters, as well as the ability to preserve the georeference of the given pixels and their values. In the pipeline, this package was used to read and write rasters, and for other tasks such as cropping, rescaling, and interpolation. For further matrix operations the package *NumPy*[3] in version 1.26.4 was used. Another important package that was used in the pipeline is *OpenCV*[4] in version 4.9.0.

To use the pipeline, a CLI script has been developed with several options, such as resource allocation for high-performance computing clusters. In this way, local EnMAP data, which can also be obtained using a provided CLI script for the data scraping tool described in Section 4.1, can be automatically annotated with Sentinel-2 data and transformed into model input data. Each stage of the pipeline can also be triggered separately. To achieve a reasonable number of training samples and to keep the processed five-dimensional matrices in the model at an acceptable size, the shape for each generated patch was set to 32×32. Furthermore, Ullah and Song demonstrated that this is a reasonable patch size for image super-resolution CNNs [33].

For the actual sharpening process, an additional stage was implemented that scrapes and aligns auxiliary Sentinel-2 data for given EnMAP data and proceeds to tile both. The resulting data is saved as model-ready, resource-efficient arrays without the overhead of a georeferenced raster. Instead, the metadata for an entire scene is saved as a single file for later reconstruction from the sharpened tiles.

[1] `https://python.org`
[2] `https://rasterio.readthedocs.io`
[3] `https://numpy.org`
[4] `https://opencv.org`

5.2 Datasets

5.2.1 Model training and validation set

For the purpose of model training, a set of EnMAP scenes was first obtained using the method described in Section 4.1. The constraints passed to the data scraping tool included a maximum cumulative cloud and cirrus cover of less than 10%. Furthermore, coordinates were defined for a bounding box containing scenes from Central Europe with a longitude of 5.84° to 28.3° and a latitude of 47.9° to 54.06° in the WGS 84 coordinate reference system only. Additionally, a time range from June 1, 2022 to September 31, 2022 and the same range in 2023 was passed. The latter two constraints were set to limit the differences between the samples according to the amount of data. The given time range also aims to avoid larger discrepancies between the EnMAP and Sentinel-2 auxiliary scenes, e.g. due to less rapidly changing vegetation and less risk of snowfall. With the given constraints, except for cloud cover, 770 matching scenes were found in the database. From this, 140 unique scenes with an acceptable cloud cover and all available auxiliary data files were downloaded. This resulted in ~58 GB of data, acquired in a time of 46 minutes and 5 seconds. The preprocessing of this dataset took 18:36 min in Stage 1 and 39:36 min in Stage 2. After this stage, the dataset was checked for possible broken scenes. To do this, the file sizes of each file were plotted as shown in Figure 5.1, with ten outliers marked.

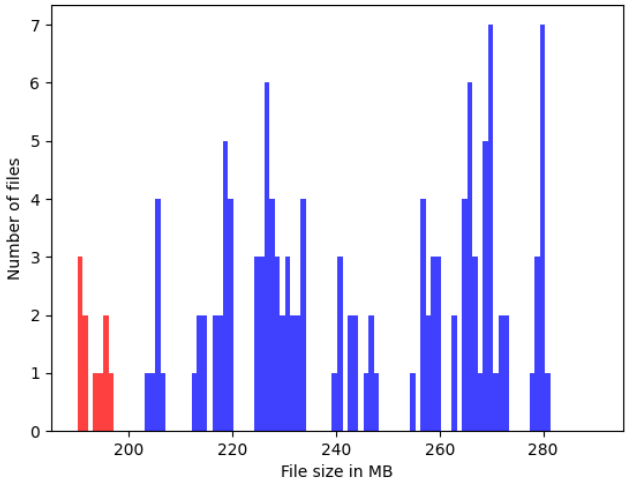
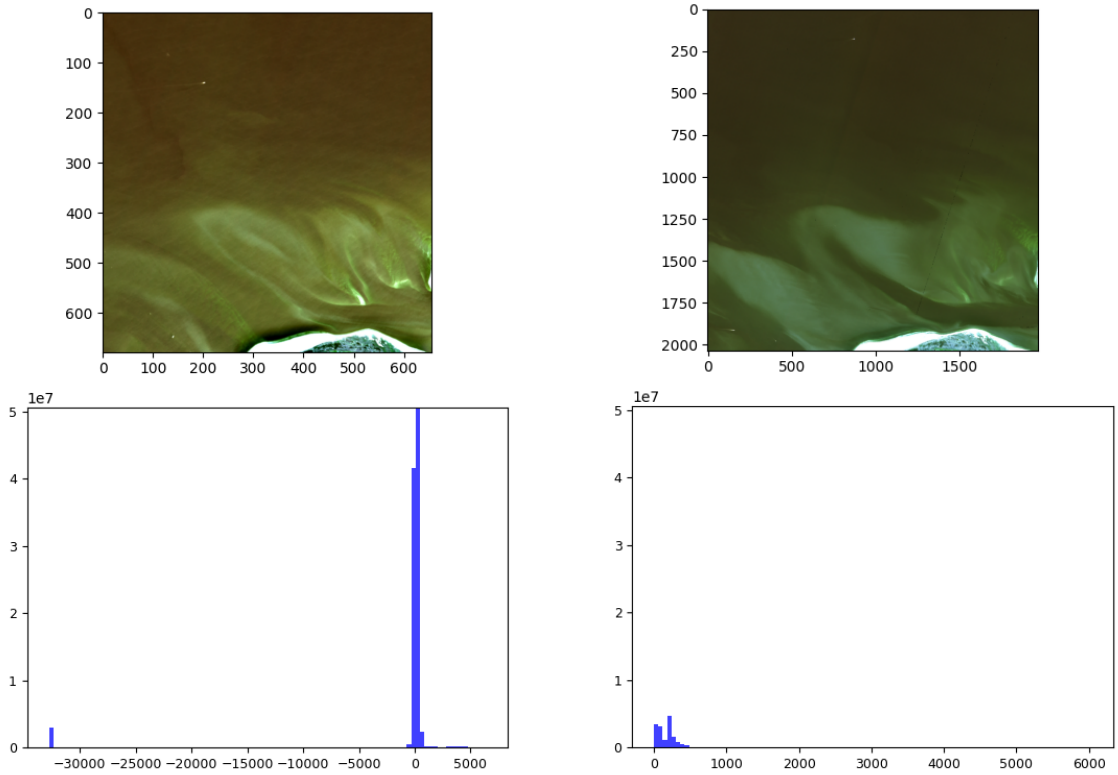


Figure 5.1: EnMAP files by size from the training dataset

Further examination of the scenes revealed that these scenes are not broken, but have many pixels with zero or very low reflectance values. Figure 5.2 visualizes an example outlier EnMAP scene with its corresponding Sentinel-2 scene, which has a large amount of water. The histograms below show the cumulative amount of each pixel value present in the scene. It can also be seen that some pixels across

all bands in the EnMAP data have a value of -32 768, indicating that no data was recorded. The histogram associated with the Sentinel-2 scene shows fewer values due to the smaller number of bands available. The same procedure was repeated for the file sizes of all acquired Sentinel-2 scenes. In contrast to other datasets that were obtained on a trial basis, no broken scenes could be detected. Therefore, no files had to be discarded before proceeding with Stage 3, which took 26:32 min. Stage 4 took 1:13:34 h and resulted in 49 021 model input tiles and the same number of reference tiles with an accumulated size of ~42 GB. Prior to training, the dataset was further split into a training set of 39 216 file pairs and a validation set of 9 805 pairs.



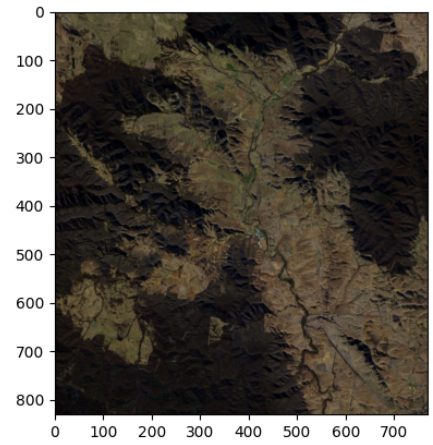
(a) EnMAP raster

(b) Sentinel raster

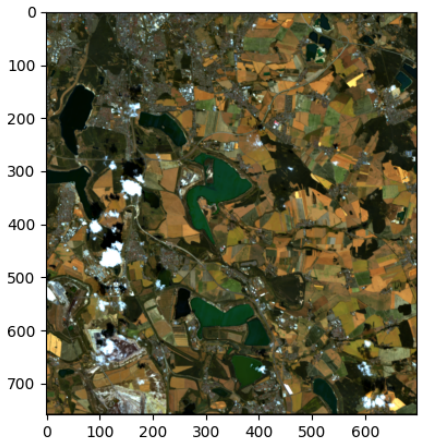
Figure 5.2: Outlier scene from the EnMAP training dataset

5.2.2 EnMAP evaluation set

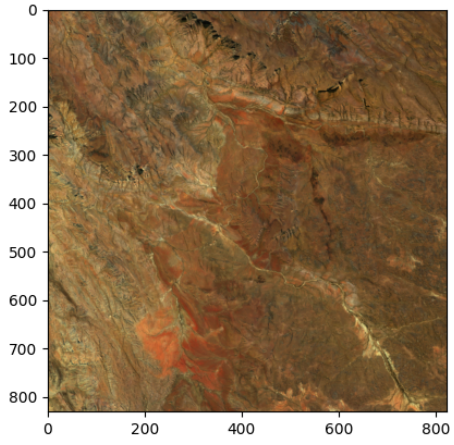
For a better evaluation of the models generalization abilities with Wald's protocol, another dataset with four scenes, visualized in Figure 5.3, not included in the training and validation datasets, was acquired. Except for Scene (b), all scenes are located in regions outside the bounding region of the training dataset. Furthermore, the scenes were chosen to contain different biomes and landscape types with potentially different reflectance properties. Scene (a) shows a river and a part of a large mountain system, Scene (b) is located in a mixed urban area with seas and agricultural fields, Scene (c) shows a part of a mountain system near a desert, and Scene (d) shows a large river and a tropical forest. Table 5.1 lists other basic characteristics of these scenes.



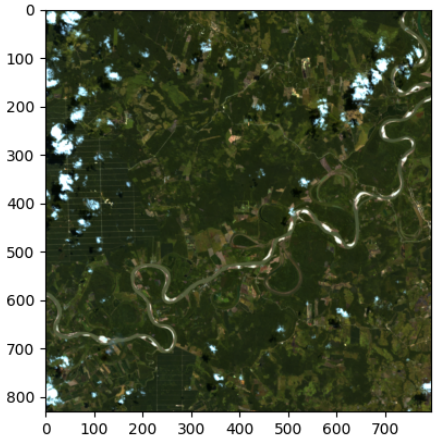
(a) Australia



(b) Germany



(c) Namibia



(d) Peru

Figure 5.3: EnMAP scenes for evaluation

Fig.	Acquisition date	Biome	Landmarks	Area
(a)	2024/05/16 10:47:29 (UTC+10)	Temperate Broadleaf and Mixed Forests	Victorian Alps, Tambo River	~576 km^2
(b)	2022/06/27 12:45:48 (UTC+2)	Temperate Broadleaf and Mixed Forests	South Leipzig, Neuseenland	~476 km^2
(c)	2024/06/11 12:03:11 (UTC+2)	Deserts and Xeric Shrublands	Baynes Mountains near Namib desert	~616 km^2
(d)	2024/06/02 10:58:32 (UTC-5)	Tropical and Subtropical Moist Broadleaf Forests	Amazon basin, Rio Aguaytia	~595 km^2

Table 5.1: Evaluation scenes comparison

5.2.3 Hyperspectral airborne evaluation scene

To evaluate the sharpening and reconstruction on the target resolution of 10 m$\times$ 10 m, an airborne hyperspectral scene acquired by the Helmholtz Centre for Environmental Research (UFZ) on June 12, 2023 is available. From 13:08 to 13:22 UTC+2, two airplane-carried sensors, HySpex SWIR-384 and HySpex VNIR-1800 from Norsk Electro Optikk, were used to obtain reflectance data for the area near the city of Leipzig, Germany. This scene has a spatial resolution of ~0.45 m $\times$ ~0.45 m and a spectral resolution of 455 bands, a maximum height of 7 550 m and a maximum width of 4 662 m. The scene covers a southern part of the Leipziger Auwald, a riparian forest near Leipzig, as well as some urban parts of the city. Figure 5.4 shows a visualization of the scene with three selected bands with center wavelengths of ~435 nm, ~535 nm, and ~701 nm.

5.3 EnMAP super resolution network

5.3.1 Training setup

The model training in the hyperparameter search, as well as the final model training were done with the Adam optimizer. The loss function used was the MS-SSIM + L1 loss proposed by Zhao et al. for image restoration tasks [42]. Furthermore, a learning rate with exponential decay from an initial value of $1e{-}3$ and a minimum value of $1e{-}5$ was determined. The training was performed on two NVIDIA RTX A6000 GPUs with equally divided batches.

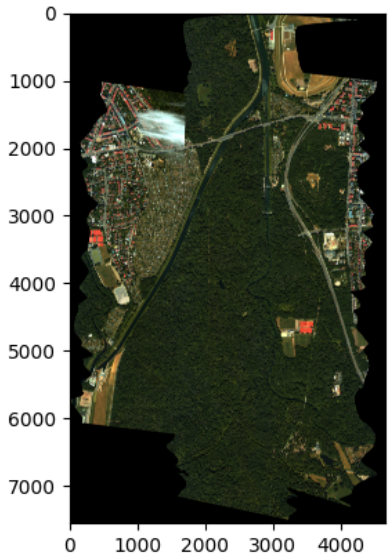


Figure 5.4: Airborne high-resolution hyperspectral scene

5.3.2 Hyperparameter search

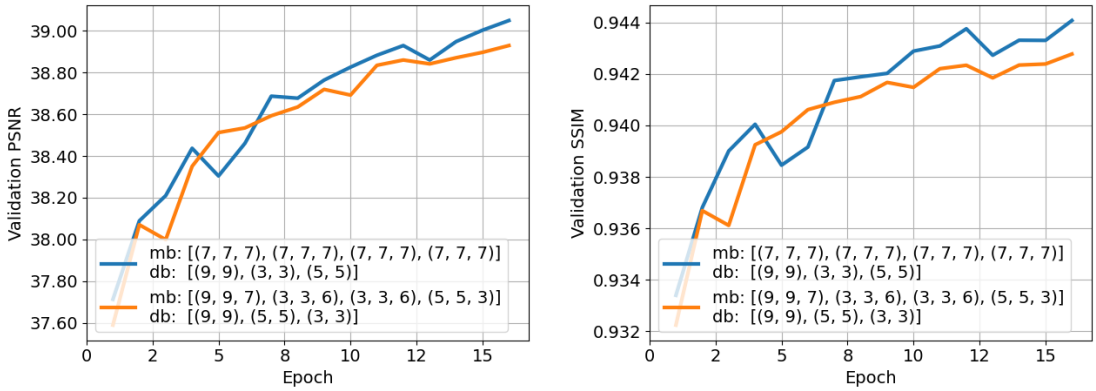
As described in Section 4.3, the shape and number of filter kernels used were determined by hyperparameter search. This was done by training several models for different hyperparameter combinations and comparing their performance on the validation set. The list of possible parameter values was created by combining the parameters used in SA-PNN [11], MCNet [18], 3D-FCNN [25], F3DUN [21]. Each model was trained with the EnMAP bands 20 to 60 for 16 epochs with a batch size of 32.

The first step was to find the best combination of filter kernel sizes. The lists of possible shapes for the main branch K_{mb} and the detail branch K_{db} were determined as written in (5.1) and (5.2). Each list contains several lists representing the kernel shapes used for a configuration. Each of these lists contains three or four kernel shapes, with the dimensions (height, width) or (height, width, depth), for each layer in the respective branch.

$$\begin{aligned} K_{mb} = [&[(3,3,3),(3,3,3),(3,3,3),(3,3,3)],\\ &[(7,7,7),(7,7,7),(7,7,7),(7,7,7)],\\ &[(9,9,7),(1,1,1),(1,1,1),(5,5,3)],\\ &[(9,9,7),(3,3,1),(3,3,1),(5,5,3)],\\ &[(9,9,7),(3,3,3),(3,3,3),(5,5,3)],\\ &[(9,9,7),(3,3,6),(3,3,6),(5,5,3)]] \end{aligned} \tag{5.1}$$

$$K_{db} = [[(3,3),(3,3),(3,3)], \\ [(7,7),(7,7),(7,7)], \\ [(9,9),(3,3),(5,5)], \\ [(9,9),(5,5),(3,3)]] \tag{5.2}$$

In this way, 24 models were trained with three filters in each layer of the detail branch and 64, 32, 9 number of filters in the main branch. The two best models achieved similar performance on the validation set, as shown in Figure 5.5. It can be seen, that the model, further referred to as Config 1, with kernel shapes of $K_{mb} = [(7,7,7),(7,7,7),(7,7,7),(7,7,7)]$ and $K_{db} = [(9,9),(3,3),(5,5)]$ could perform slightly better than the model labeled Config 2 with kernel shapes of $K_{mb} = [(9,9,7),(3,3,6),(3,3,6),(5,5,3)]$ and $K_{db} = [(9,9),(5,5),(3,3)]$.



(a) PSNR on validation set

(b) SSIM on validation set

Figure 5.5: Model performances with different kernel shapes

These two models were afterwards evaluated using Wald's protocol on the four scenes described in Section 5.2.2 to assess their generalization abilities. In addition to the metrics already presented, the Spectral Angle Mapper (SAM) algorithm was used to evaluate the ability to reconstruct spectral information. [41]. A lower value describes less spectral distortion between two images. All evaluation results of the outputs produced by Config 1, Config 2, and the initial upsampled and bilinear interpolated scenes are shown in Table 5.2. The displayed metrics were calculated, averaged, and rounded on each tile of the given scene. The best values for each scene are marked with an asterisk, and it can be seen that Config 2 performs best in almost all tests. Due to the larger filter kernels in the main branch, Config 1 also has a higher complexity. This led to a training time of 1:29:10 h, compared to 58:20 min for Config 2. With an average prediction time per tile of 653 ms, Config 1 is also slower than Config 2, where the prediction of a tile took an average of only 209 ms.

Based on these considerations, the hyperparameter search for the ideal number of filters per layer was carried out on Config 2. Eight models with different combinations of the number of filters in the main branch F_{mb} and in the detail branch

Kernels	Metric	Australia	Germany	Namibia	Peru
Config 1	MSE	691 968	41 697	19 428	11 060
	PSNR	25.65	29.90	29.03	33.38
	SSIM	0.85	0.87	**0.83***	**0.92***
	SAM	8.46	3.13	0.92	1.35
Config 2	MSE	**671 535***	**40 736***	**19 362***	**10269***
	PSNR	**25.78***	**30.03***	**29.08***	**33.71***
	SSIM	**0.86***	**0.87***	**0.83***	**0.92***
	SAM	**7.99***	**3.07***	0.82	**1.29***
Bilinear	MSE	875 555	55 623	24 672	17 940
	PSNR	24.41	28.37	27.22	31.39
	SSIM	0.77	0.78	0.67	0.85
	SAM	9.30	3.84	**0.80***	1.86

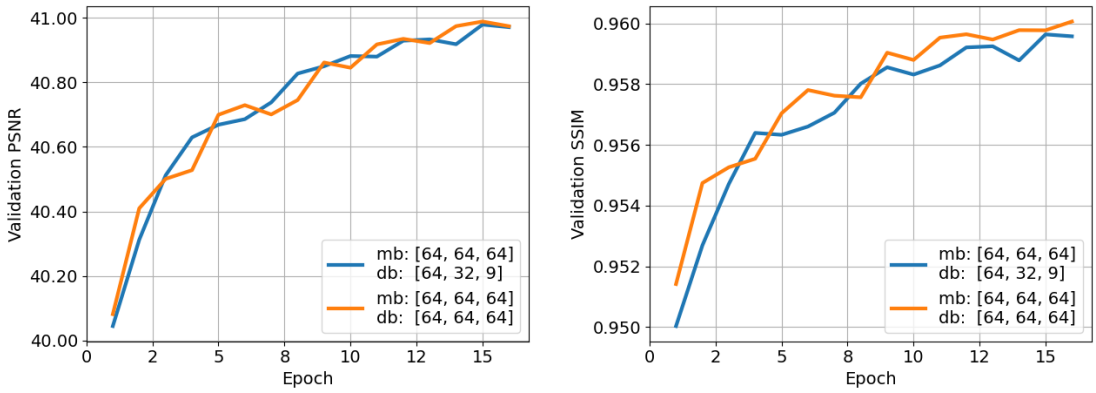
Table 5.2: Evaluation results of models with different kernel shapes

F_{db} were trained and compared. The possible parameter configurations are listed in (5.3) and (5.4), where each value refers to the number of filters used in the layer according to the position in the respective list. The fourth convolutional layer in the main brain was always set to have one filter only, as this forms the output residual image.

$$F_{mb} = [[64, 64, 64], [64, 32, 9]] \tag{5.3}$$

$$F_{db} = [[64, 64, 64], [64, 32, 9], [9, 6, 3], [3, 3, 3]] \tag{5.4}$$

The two models with the best performance on the validation set, visualized in Figure 5.6, were selected for further evaluation and comparison. The model with $F_{mb} = [64, 64, 64]$ and $F_{db} = [64, 32, 9]$ number of filters is further referred to as Config 3. The second model with $F_{mb} = [64, 64, 64]$ and $F_{db} = [64, 64, 64]$ number of filters is called Config 4. As illustrated in Table 5.3, both models show an improvement in all metrics, compared to the bilinear interpolated image, with Config 3 performing slightly better. The training of Config 3 took 1:55:50 h, while it took 2:19:00 h for Config 4. Config 3 also showed a faster prediction time, averaging 465 ms per tile compared to Config 4 averaging 580 ms per tile. With these considerations, the parameters of Config 3 were determined for the model to be trained on 224 EnMAP bands.



(a) PSNR on validation set

(b) SSIM on validation set

Figure 5.6: Model performances with different number of filters per layer

Filters	Metric	Australia	Germany	Namibia	Peru
Config 3	MSE	**650 646***	**42 431***	17 813	**10 963***
	PSNR	25.85	**29.78***	29.60	**33.49***
	SSIM	**0.86***	**0.86***	**0.85***	**0.92***
	SAM	7.59	**3.06***	0.77	**1.30***
Config 4	MSE	653 071	43 002	**17 786***	10 998
	PSNR	**25.90***	29.71	**29.63***	33.47
	SSIM	**0.86***	**0.86***	**0.85***	**0.92***
	SAM	**7.45***	3.10	**0.76***	1.31
Bilinear	MSE	875 555	55 623	24 672	17 940
	PSNR	24.41	28.37	27.22	31.39
	SSIM	0.77	0.78	0.67	0.85
	SAM	9.30	3.84	0.80	1.86

Table 5.3: Evaluation results of models with different number of filters per layer

5.3.3 Proposed architecture

The proposed Convolutional Neural Network model architecture for achieving super-resolution of hyperspectral EnMAP images, called supErMAPnet, was implemented in the *Python*[5] programming language version 3.11.9 and with the use of the packages *NumPy*[6] version 1.26.4 and *TensorFlow*[7] version 2.15.0.

The architecture consists of two branches, each with one input layer. Each branch has three hidden convolutional layers, and after each one, an injection from the

[5] `https://python.org`
[6] `https://numpy.org`
[7] `https://tensorflow.org`

detail branch is performed in the main branch. Then, a final layer performs a 3D convolution with one filter with a shape of $5 \times 5 \times 3$ and an identity activation function to create the final residual image. This is added to the degraded input image by a skip connection to form the reconstructed output image. The hidden layers in the main branch perform 3D convolutions with zero padding and 64 filters each. The filters of the first main branch layer have a shape of $9 \times 9 \times 7$, while the filters of following two layers have a shape of $3 \times 3 \times 6$. All layers use leakyReLU as their activation function. The detail branch has three hidden layers. They perform 2D convolutions on a reflection-padded input. Batch normalization is performed on the outputs of each detail layer before they are passed to the main branch and the next detail layer. The three detail layers use 64, 32, and 9 filters with a shape of 9×9, 5×5, 3×3. The complete *TensorFlow* model graph is shown in Figure A1 in the appendix.

5.3.4 Final model training

After all parameters were determined, a model was trained using all 224 bands of the EnMAP training data. The general training setup was the same as described in Section 5.3.1, but with a lower initial learning rate of $1e{-}4$, as loss explosions were experienced at higher initial learning rates. On account of the high resource consumption due to convolutions performed on five-dimensional matrices, the batch size had to be reduced to four. Training was stopped if the validation loss did not improve by at least 0.02 within five epochs. Then the best weights, in this case from epoch 31, within that five epoch period were restored. Training with the 39 216 pairs of input samples and 9 805 pairs of validation samples took 72:03:33 h. Figure 5.7 shows the training and validation loss up to epoch 31. It can be seen that both values tend to a limit, which shows that there are no excessive underfitting or overfitting phenomena.

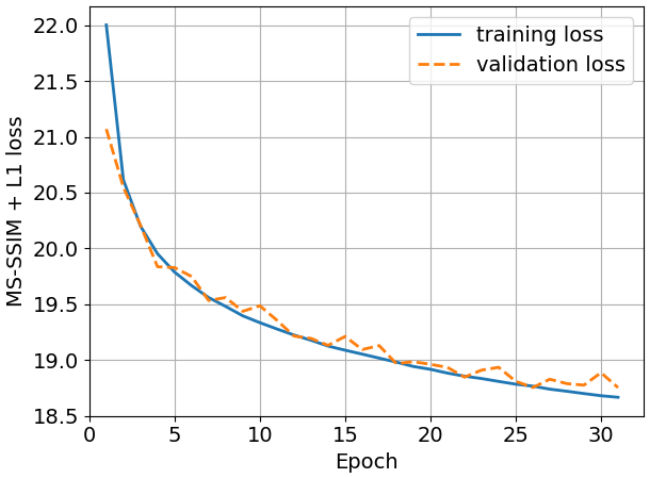


Figure 5.7: Training and validation loss of supErMAPnet

6 Results

6.1 Evaluation with Wald's protocol

Using Wald's evaluation strategy, supErMAPnet was evaluated on the dataset described in Section 5.2.2. The results listed in Table 6.1 show that supErMAPnet improves the quality of hyperspectral images scaled and interpolated from 90 m $\times$ 90 m resolution to 30 m $\times$ 30 m resolution. The supErMAPnet significantly improves the structural similarity between the reference image and the degraded input and helps to reduce the spectral distortions in the image reconstruction process. Additionally, Table 6.1 displays the average prediction time the model consumed to predict each tile. Prediction was performed on an Intel i7-10510U CPU running at 1.8GHz.

Method	Metric	Australia	Germany	Namibia	Peru
supErMAPnet	MSE	191 016	161 046	22 789	74 404
	PSNR	30.62	28.43	37.78	32.60
	SSIM	0.90	0.87	0.92	0.93
	SAM	4.66	4.41	1.41	2.19
	Time	3.32 s	2.84 s	3.21 s	2.94 s
Bilinear	MSE	261 836	177 369	39 524	111 989
	PSNR	28.86	28.10	35.13	30.94
	SSIM	0.82	0.83	0.84	0.86
	SAM	6.00	5.06	1.81	2.69

Table 6.1: Average evaluation results per tile

Since sharpening is usually desired without altering the original scenes, all individual tiles sharpened by the model must be stitched together and annotated with the original raster metadata with an updated pixel-to-coordinate mapping. A tool to automatically reconstruct scenes from multiple tiles has also been implemented and is available in the code repository. For further evaluation, the four sharpened evaluation scenes were reconstructed and compared to their originals, and to the same scenes sharpened with bilinear interpolation, cubic interpolation, and a bilinear interpolation combined with an unsharp masking filter kernel. The reconstruction was done on the Intel CPU mentioned above and took 27.4 s for the

scene from Australia, 44.3 s for the scene from Germany, 33.6 s for the scene from Namibia and 27.6 s for the scene from Peru. The averaged evaluation metrics in Table 6.2 show a clear advantage of supErMAPnet, especially in the improvement of structural similarity and the reduction of spectral distortions as indicated by the SAM value.

Method	MSE	PSNR	SSIM	SAM
Bilinear	3251	29.20	0.86	8.00
Cubic	3255	29.64	0.87	7.61
Bilinear + UM	3271	29.68	0.88	7.59
supErMAPnet	**3238***	**30.73***	**0.92***	**6.86***

Table 6.2: Average evaluation results of reconstructed scenes

To investigate possible discrepancies in the sharpening quality of supErMAPnet over different spectral bands, the coefficient of determination R^2 and the SSIM values between the sharpened scenes and their originals were calculated for each band and averaged over the four evaluation scenes. The resulting plot is visualized in Figure 6.1, and a boxplot with detailed R^2 values for each band can be found in the Appendix (Figure A2). While a comparatively low R^2 value can be observed in the visual spectrum from band 1 to 58, the structural similarity index value is remarkably high, indicating the enhancement of details and structures in this spectrum. Bands 130 to 134 fall in the electromagnetic absorption spectrum of water and take the maximum value for each pixel in the EnMAP data from the evaluation set. Because these bands are constant, no R^2 is computed, but the SSIM values show a good fit of supErMAPnet for this spectrum. Slightly worse values can be seen from band 164 to band 171. This may be due to unusual value spikes that can be observed in the original EnMAP data for these bands. There is also a larger spectral gap between band 163 with a center wavelength of ~1760 nm and band 164 with a center wavelength of ~1939 nm. The 20th percentile of all bands R^2 value is at 0.84. The 20 worst bands by this metric are bands 1 through 29, which cover a spectrum from ~418 nm to ~556 nm and are part of the visual spectrum, bands 87, 89, 91, 92 of the near-infrared spectrum, the aforementioned bands 164 through 171 without band 165, and the bands 219, 221, 222, and 223 of the shortwave infrared spectrum. To investigate whether the supErMAPnet sharpening results are biased by a good fit of the water absorption bands, the evaluation and comparison was performed again without these bands. The results listed in Table 6.3 indicate more spectral distortions in the sharpened scenes compared to the results listed in Table 6.2, but still better values obtained by supErMAPnet compared to other methods. The SSIM and PSNR values are almost unchanged.

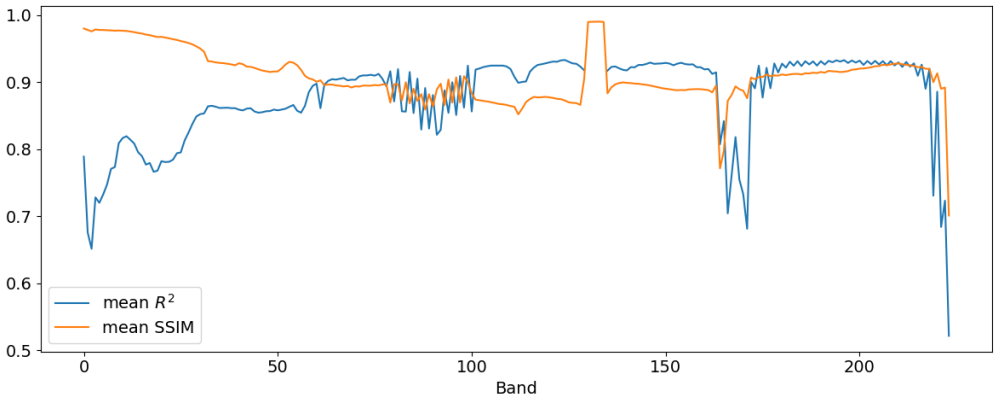


Figure 6.1: Mean R^2 and SSIM per band

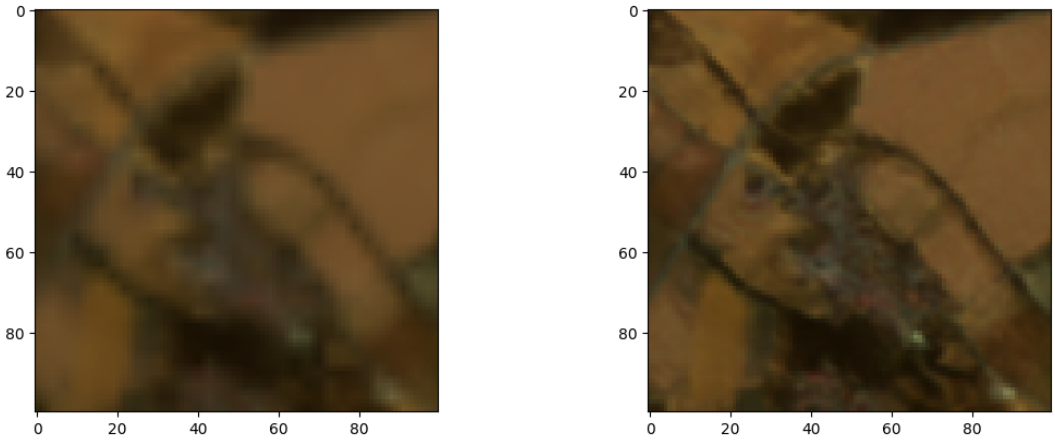
Method	MSE	PSNR	SSIM	SAM
Bilinear	3316	29.24	0.85	9.62
Cubic	3328	29.68	0.87	9.15
Bilinear + UM	3346	29.71	0.87	9.12
supErMAPnet	**3309***	**30.72***	**0.92***	**8.28***

Table 6.3: Average evaluation results of reconstructed scenes without water reflection bands

6.2 Sharpening on Sentinel-2 resolution

The results in Section 6.1 show that the proposed model can effectively sharpen hyperspectral EnMAP data to a three times higher resolution on a degraded scale. To show that this also works at the target resolution of 10 m $\times$ 10 m, a section of the upsampled and sharpened scene from Germany by supErMAPnet at this resolution is visualized and compared to the bilinear interpolated scene in Figure 6.2. Further comparisons of the sharpened sections from the other evaluation scenes can be found in the Appendix under figure numbers A3 to A6.

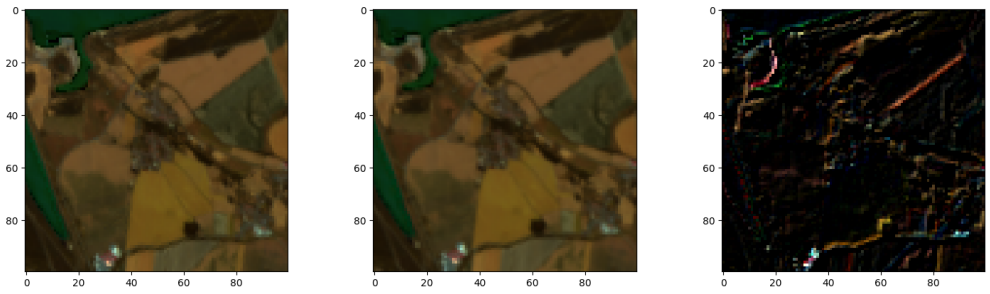
According to Wald et al., synthetic images should be as identical as possible to the original image when they are degraded back to the original resolution [34]. This is tested by downsampling the reconstructed scene from Germany, described in Section 5.2.2, from a resolution of 10 m $\times$ 10 m per pixel back to a resolution of 30 m $\times$ 30 m per pixel. Figure 6.3 shows the original scene, the downsampled prediction, and the computed differences between the three bands used for visualization. The values of Subfigure (c) have been increased by a factor of ten for better visibility. Since the selected patches have a width and height of 100 pixels, they cover several individually sharpened patches of the model. In Subfigure (c), no border effects are visible that might have been introduced by the tiling and reconstruction process. Instead, there are differences in the edge components of the scene, which can be explained by the injected details, some of which are still



(a) Bilinear interpolation **(b)** supErMAPnet reconstruction

Figure 6.2: Area of 1 km^2 from scene Germany at 10 m $\times$ 10 m per pixel

present at the original resolution. The boxplot in Figure 6.4 shows the distribution of R^2 values between the original and the downsampled prediction for all four evaluation scenes. For clarity, only the data points of every second band from band 2 to band 60 have been visualized. With the exception of a few outliers, there is an overall high median, indicated by the orange line.



(a) Original scene **(b)** Prediction downscaled **(c)** Differences

Figure 6.3: Area of 3 km^2 from scene Germany at 30 m $\times$ 30 m per pixel

To better evaluate the sharpening of EnMAP images to the same spatial resolution as Sentinel-2 data with the proposed model, co-registered high spatial resolution hyperspectral data must be available. The hyperspectral airborne scene described in Section 5.2.3 covers an area that is also covered by the evaluation scene from Germany from the dataset described in Section 5.2.2. Both scenes were taken around noon in June, so the vegetation and shadows should be similar. However, the aerial scene was captured a year later, which could lead to greater differences. Unfortunately, there are no available EnMAP scenes for this location in the DLR database that were recorded in the same year. After cropping both scenes to a co-registered section, 224 bands with center wavelengths close to the EnMAP bands

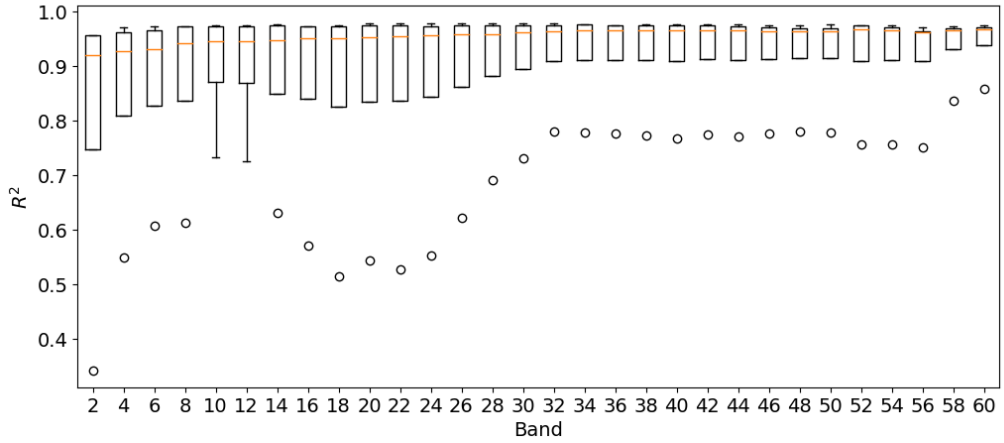


Figure 6.4: R^2 values for selected bands between original and downsampled reconstruction for four evaluation scenes

were selected from the aerial scene. The afterwards computed evaluation metrics are displayed in Table 6.4. It can be seen that for the selected area of the image no significant improvement was achieved using supErMAPnet, but also that there is a high overall discrepancy between the EnMAP and the airborne scene, as can be evidenced by the high SAM value.

Method	MSE	PSNR	SSIM	SAM
Bilinear	1959	**10.63***	0.39	**50.05***
supErMAPnet	**1941***	10.55	**0.40***	50.19

Table 6.4: Evaluation results using an airborne and EnMAP scene

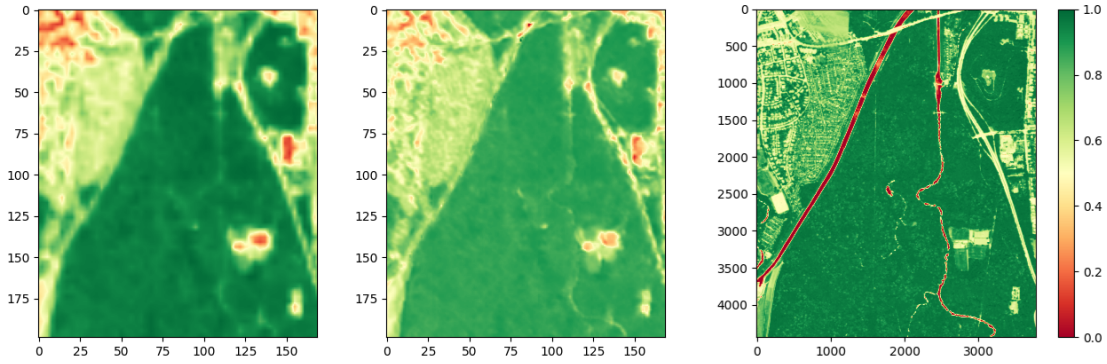
Another evaluation using the high-resolution hyperspectral scene was done by calculating and comparing different vegetation indices. Figure 6.5 shows the normalized difference vegetation index (NDVI) for the cropped scene recorded by the UFZ in comparison to the same area from the above mentioned EnMAP scene, once bilinear interpolated and once sharpened with supErMAPnet to a resolution of 10 m $\times$ 10 m. The NDVI is commonly used to assess the health or density of vegetation using remote sensing data [30]. The visualization shows that structures and boundaries of dense vegetation are clearer in the sharpened image than in the interpolated image. Red spots with smaller values are also less blurred and affect lesst adjacent pixels. Although the small river arms are slightly more structured in the sharpened EnMAP image, they are not as clearly visible as in Subfigure (c), where they are colored red due to very low NDVI values. However, it is noticeable that the NDVI in the area of dense vegetation in the center of the image is lower in the sharpened EnMAP scene than in the interpolated and the UFZ scene. In addition to the NDVI, the green normalized difference vegetation index (GNDVI), which is used to estimate photosynthetic activity, and the normalized difference infrared index (NDII), which can be used to interpret e.g. moisture

storage deficits in vegetation root zones, were also calculated. In (6.1) to (6.3) the formulas of NDVI, GNDVI and NDII are shown. The selection of suitable EnMAP bands was based on the center wavelengths of the Sentinel-2 bands used by Brook et. al to calculate these indices [3]. To take advantage of EnMAP's higher spectral resolution, several bands with a similar wavelengths to the corresponding Sentinel-2 band were selected and averaged. Bands 26 to 34 were used for the $Green$ variable, bands 48 to 55 for Red, bands 68 to 76 for NIR and bands 145 to 153 for $SWIR$.

$$NDVI = \frac{NIR - Red}{NIR + Red} \tag{6.1}$$

$$GNDVI = \frac{NIR - Green}{NIR + Green} \tag{6.2}$$

$$NDII = \frac{NIR - SWIR}{NIR + SWIR} \tag{6.3}$$



(a) Bilinear interpolated **(b)** supErMAPnet sharpened **(c)** UFZ scene

Figure 6.5: NDVI map showing parts of the Auwald, Leipzig

Subsequently, bands with similar wavelengths were selected from the hyperspectral UFZ scene, the same vegetation indices were calculated and compared with the indices calculated from the bilinear interpolated and the sharpened EnMAP scene. A comparison of the resulting MSE values is shown in Table 6.5. For all three indices, the sharpened scene with supErMAPnet achieves a better performance. It is important to note that the resulting values are dependent on the selected bands and further tests with better temporal matching scenes need to be carried out for a more meaningful value.

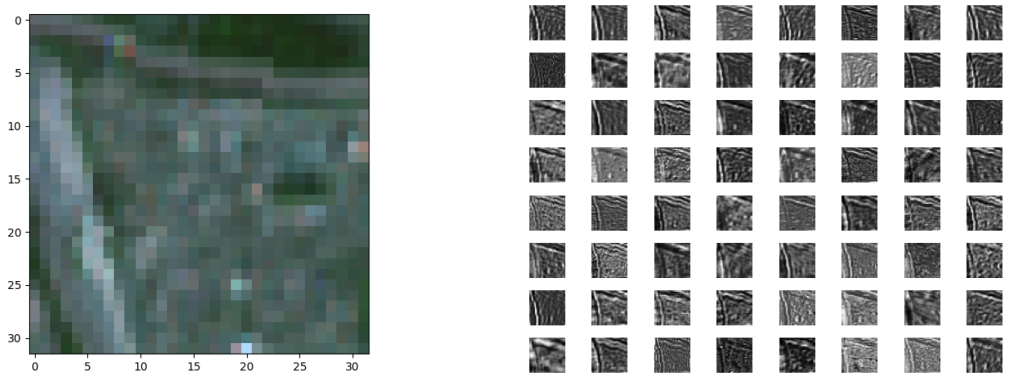
Method	NDVI	GNDVI	NDII
MSE(Bilinear, UFZ)	0.0384	0.0302	0.0693
MSE(supErMAPnet, UFZ)	**0.0340***	**0.0268***	**0.0462***

Table 6.5: MSE of vegetation indices in comparison to the UFZ scene

6.3 Model insights

For a better understanding of how the model works, the outputs of different layers have been visualized during a single prediction run. Figure 6.6 (a) shows the visualization of a multispectral input passed to the detail branch. Subfigures (b) to (d) show the feature maps created by each layer of the detail branch. It can be seen that most feature maps emphasize details of the input as expected. Some feature maps only exhibit edges from a certain direction. This is what directional Sobel operators do, for example, in a simplified way. The output of each layer in the detail branch is analogous to (2.3) but with a reflection padding pad_r performed on the layer inputs and a batch normalization $norm_b$ performed before the layer activation ϕ function is applied. This can be defined as follows:

$$Y_l = \phi(norm_b(W_l * pad_r(X_l) + B_l)) \tag{6.4}$$



(a) Input tile

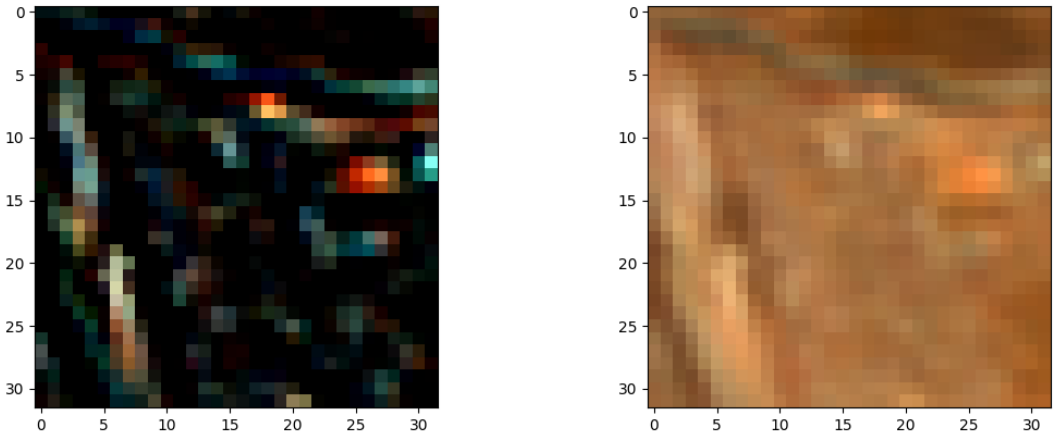
(b) Feature maps from first detail layer

(c) Feature maps from third detail layer

(d) Feature maps from third detail layer

Figure 6.6: Multispectral input and created feature maps in the detail branch

Figure 6.7 (a) shows the residual image generated by the last convolutional layer of the main branch, while Subfigure (b) visualizes the network output created by adding the residual image to the low-resolution hyperspectral input image. Most of the residual image values tend to zero, which helps to speed up the network and avoid vanishing gradients, as explained in Section 3.1.1.



(a) Residual image **(b)** Output image

Figure 6.7: Image in the main branch before and after the skip connection

7 Discussion

The results of this work demonstrate that the trained model sharpened EnMAP data with less spectral distortion and better structural improvement than various interpolation and kernel sharpening methods, which was demonstrated on a smaller scale. Experiments at the target resolution also showed reasonable sharpening results. By downsampling the sharpened results to their original resolution, it was shown that no unwanted phenomena such as edge effects, overemphasized jaggies, or ringing effects were introduced by the model. While the trained model showed good overall sharpening results, some bands had worse results than others. This may be due to the fact that most of the architectural parameters were only set and tested for bands 20 to 60, and to characteristics of the EnMAP data product, such as a larger spectral gap between bands 163 and 164.

The trained model spatially sharpens all 224 spectral bands of the EnMAP data and showed good generalization qualities, while other recently proposed models are scene dependent and need to be retrained for every location. Another contrast is, that the developed supErMAPnet was trained and tested on EnMAP L2A products instead of popular hyperspectral scenes that were specifically selected and prepared for analysis tasks.

By sharpening low-resolution hyperspectral EnMAP data, this work has the potential to contribute to the improvement of remote sensing based analysis, such as object detection or environmental monitoring tasks. The developed preprocessing tools allow for automatic data acquisition of EnMAP and corresponding Sentinel-2 remote sensing data, as well as consistent data preprocessing that enables retraining of the model architecture with different parameters. In this way, future improvements of the model and research on super-resolution for hyperspectral remote sensing data will have a better comparability.

The analysis of sharpening at the target resolution was limited by the low availability of high-resolution hyperspectral data and the difficult accessibility of temporally and spatially co-registered EnMAP data. By combining the DLR database, which is now regularly updated with the latest EnMAP data, with the preprocessing tools developed, it will be easier to obtain EnMAP data for future analysis. A drawback of the proposed methodology is that the original EnMAP scene must be cropped to a rectangular shape whose dimensions must be a multiple of the model input tile size. In addition, data acquisition, preprocessing, and model training are tasks that consume large amounts of memory and computing power due to the high dimensionality of the data used. However, utilizing the trained model to sharpen individual scenes is less resource intensive and does not require high-performance computing infrastructure. Furthermore, due to the chosen image

fusion approach, clouds from the Sentinel-2 auxiliary images prevent the EnMAP data from being sharpened in these regions.

Some of the researchers who proposed models for hyperspectral image super-resolution simulated missing auxiliary data by averaging the hyperspectral bands. This approach could also be used to fill in cloud-covered regions with artificial data. With the given framework, this and other experimental approaches can be easily carried out and compared. This could also be the repetition of the hyperparameter search, which was done under the assumption that an appropriate model parameter configuration, except for trainable weights and biases, is little dependent on the size of the input spectrum. As explained before, performing a hyperparameter search with all 224 bands of EnMAP is a resource intensive task. The growth in available computational power and developments in the area of big data over the next few years may therefore facilitate future research into hyperspectral image super-resolution.

8 Conclusion

The proposed and implemented methods have successfully realized the objective of developing a technique to achieve enhanced spatial resolution for hyperspectral remote sensing data acquired by the Environmental Mapping and Analysis Program satellite. Hyperspectral EnMAP images acquired at different locations and times of day and year can be sharpened to a spatial resolution of 10 m $\times$ 10 m using the same trained model. The chosen machine learning based image fusion approach proved to be appropriate to achieve the sharpening goal, and a novel Convolutional Neural Network architecture with a mix of three- and two-dimensional filter kernels was designed that utilizes auxiliary multispectral data for the reconstruction of degraded details. Moreover, the tools that were developed in this work enable the further improvement of the proposed techniques and help to acquire and process data for other research purposes. Although the proposed methodology has some limitations, this work was able to achieve initial results in the field of super-resolution for hyperspectral EnMAP data. These developments can contribute to improved analysis and interpretation of EnMAP satellite data and can serve as a basis for applying super-resolution techniques to other satellite missions and remote sensing applications.

Bibliography

[1] Alparone, Luciano *et al.*: *A pyramid-based approach to multisensor image data fusion with preservation of spectral signatures*. In *Future Trends in Remote Sensing*, pages 419–426. AA BALKEMA PUBLISHERS, 1998. 8

[2] Bachmann, Martin *et al.*: *Analysis-ready data from hyperspectral sensors—the design of the enmap card4l-sr data product*. Remote Sensing, 13(22):4536, 2021. 21, 25

[3] Brook, A. *et al.*: *A smart multiple spatial and temporal resolution system to support precision agriculture from satellite images: Proof of concept on aglianico vineyard*. Remote Sensing of Environment, 240:111679, 2020, ISSN 00344257. 2, 9, 13, 14, 18, 46

[4] Chabrillat, Sabine *et al.*: *Enmap science plan*. EnMAP Technical Report, 2022. 1, 2, 5

[5] Cherif, Eya *et al.*: *From spectra to plant functional traits: Transferable multi-trait models from heterogeneous and sparse data*. Remote Sensing of Environment, 292:113580, 2023, ISSN 00344257. 2

[6] Deng, Guang: *A generalized unsharp masking algorithm*. IEEE transactions on image processing : a publication of the IEEE Signal Processing Society, 20(5):1249–1261, 2011. 7

[7] Dong, Chao *et al.*: *Image super-resolution using deep convolutional networks*. IEEE transactions on pattern analysis and machine intelligence, 38(2):295–307, 2016. 13, 15, 18

[8] Evangelidis, Georgios D. and Emmanouil Z. Psarakis: *Parametric image alignment using enhanced correlation coefficient maximization*. IEEE transactions on pattern analysis and machine intelligence, 30(10):1858–1865, 2008. 26

[9] Fadnavis, Shreyas: *Image interpolation techniques in digital image processing: An overview*. International Journal Of Engineering Research and Application, 4:2248–962270, 2014. 6

[10] Guanter, Luis *et al.*: *The enmap spaceborne imaging spectroscopy mission for earth observation*. Remote Sensing, 7(7):8830–8857, 2015. 1

[11] He, Lin *et al.*: *A spectral-aware convolutional neural network for pansharpening*. Applied Sciences, 10(17):5809, 2020. 8, 9, 15, 16, 17, 18, 19, 28, 29, 30, 36

[12] Huang, Huijuan, Jing Yu, and Weidong Sun: *Super-resolution mapping via multi-dictionary based sparse representation*. In *2014 IEEE International Con-*

ference on Acoustics, Speech and Signal Processing (ICASSP), pages 3523–3527. IEEE, 2014, ISBN 978-1-4799-2893-4. 6

[13] Ioffe, Sergey and Christian Szegedy: *Batch normalization: Accelerating deep network training by reducing internal covariate shift*, 2015. 29

[14] Jaramaz, Darko *et al.*: *The esa sentinel-2 mission vegetation variables for remote sensing of plant monitoring*. In *Conference Proceedings - RESPAG 2nd International Scientific Conference*, pages 950–961, May 2013. 1

[15] Kim, Jiwon, Jung Kwon Lee, and Kyoung Mu Lee: *Accurate image super-resolution using very deep convolutional networks*, 2015. 7, 18

[16] Kim, Sung and Riley Casper: *Applications of convolution in image processing with matlab*. University of Washington, pages 1–20, 2013. 6, 7

[17] Kohonen, Oili: *Multiresolution-based pansharpening in spectral color images*. In *Conference on Colour in Graphics, Imaging, and Vision*, volume 5, pages 535–540. Society of Imaging Science and Technology, 2010. 8

[18] Li, Qiang, Qi Wang, and Xuelong Li: *Mixed 2d/3d convolutional network for hyperspectral image super-resolution*. Remote Sensing, 12(10):1660, 2020. 7, 15, 18, 19, 30, 36

[19] Liebel, L. and M. Körner: *Single-image super resolution for multispectral remote sensing data using convolutional neural networks*. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLI-B3:883–890, 2016. 15, 18

[20] Liu, Yaoting *et al.*: *Interactformer: Interactive transformer and cnn for hyperspectral image super-resolution*. IEEE Transactions on Geoscience and Remote Sensing, 60:1–15, 2022, ISSN 0196-2892. 7, 18, 19

[21] Liu, Ziqian *et al.*: *Rethinking 3d-cnn in hyperspectral image super-resolution*. Remote Sensing, 15(10):2574, 2023. 7, 15, 18, 19, 30, 36

[22] Lone, Zubair Ahmad and Alwyn Roshan Pais: *Object detection in hyperspectral images*. Digital Signal Processing, 131:103752, 2022, ISSN 10512004. 1

[23] Lu, Xiaochen *et al.*: *Coupled convolutional neural network-based detail injection method for hyperspectral and multispectral image fusion*. Applied Sciences, 11(1):288, 2021. 9, 17, 18, 19, 27, 29

[24] Masi, Giuseppe *et al.*: *Pansharpening by convolutional neural networks*. Remote Sensing, 8(7):594, 2016. 8, 13, 14, 17, 18, 29

[25] Mei, Shaohui *et al.*: *Hyperspectral image spatial super-resolution via 3d full convolutional neural network*. Remote Sensing, 9(11):1139, 2017. 6, 7, 8, 15, 17, 18, 19, 30, 36

[26] Peyghambari, Sima and Yun Zhang: *Hyperspectral remote sensing in lithological mapping, mineral exploration, and environmental geology: an updated review*. Journal of Applied Remote Sensing, 15(03), 2021, ISSN 1931-3195. 2

[27] Pham, Tuan D.: *Kriging-weighted laplacian kernels for grayscale image sharpening*. IEEE Access, 10:57094–57106, 2022. 7

[28] Polesel, A., G. Ramponi, and V. J. Mathews: *Image enhancement via adaptive unsharp masking*. IEEE transactions on image processing : a publication of the IEEE Signal Processing Society, 9(3):505–510, 2000. 6, 7

[29] Qi, Charles R. *et al.*: *Volumetric and multi-view cnns for object classification on 3d data*. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 15

[30] Rouse Jr, John W *et al.*: *Monitoring the vernal advancement and retrogradation (green wave effect) of natural vegetation*. Technical report, Texas A&M University Remote Sensing Center, 1974. 45

[31] S2 MSI ESL Team: *Data quality report: Sentinel-2 l1c msi: April 2024*, 2024. 26

[32] Spoto, Francois *et al.*: *Overview of sentinel-2*. In *2012 IEEE International Geoscience and Remote Sensing Symposium*, pages 1707–1710. IEEE, 2012, ISBN 978-1-4673-1159-5. 1, 24

[33] Ullah, Safi and Seong Ho Song: *Srresnet performance enhancement using patch inputs and partial convolution-based padding*. Computers, Materials & Continua, 74(2):2999–3014, 2023, ISSN 1546-2226. 29, 31

[34] Wald, Lucien, Thierry Ranchin, and Marc Mangolini: *Fusion of satellite images of different spatial resolutions: Assessing the quality of resulting images*. Photogrammetric Engineering and Remote Sensing, 63:691–699, 1997. 16, 43

[35] Wang, Yao *et al.*: *Hyperspectral image super-resolution via nonlocal low-rank tensor approximation and total variation regularization*. Remote Sensing, 9(12):1286, 2017. 7

[36] Wang, Zhou *et al.*: *Image quality assessment: from error visibility to structural similarity*. IEEE transactions on image processing : a publication of the IEEE Signal Processing Society, 13(4):600–612, 2004. 28

[37] Wei, Yancong *et al.*: *Boosting the accuracy of multispectral image pansharpening by learning a deep residual network*. IEEE Geoscience and Remote Sensing Letters, 14(10), 2017. 8, 9, 14, 17, 18

[38] Wu, Jianxin: *Introduction to convolutional neural networks*. National Key Lab for Novel Software Technology. Nanjing University. China, 5(23):495, 2017. 10, 11

[39] Yang, Jingxiang, Yong Qiang Zhao, and Jonathan Cheung Wai Chan: *Hyperspectral and multispectral image fusion via deep two-branches convolutional neural network*. Remote Sensing, 10(5):800, 2018. 9, 17, 18, 19

[40] Yuan, Qiangqiang *et al.*: *A multiscale and multidepth convolutional neural network for remote sensing imagery pan-sharpening*. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 11(3):978–989, 2018, ISSN 1939-1404. 8, 14, 17, 18

[41] Yuhas, Roberta H., Alexander F. H. Goetz, and Joe W. Boardman: *Discrimination among semi-arid landscape endmembers using the spectral angle mapper (sam) algorithm*. In *JPL, Summaries of the Third Annual JPL Airborne Geoscience Workshop. Volume 1: AVIRIS Workshop*, 1992. 37

[42] Zhao, Hang *et al.*: *Loss functions for neural networks for image processing*. CoRR, abs/1511.08861, 2015. 35

Appendix

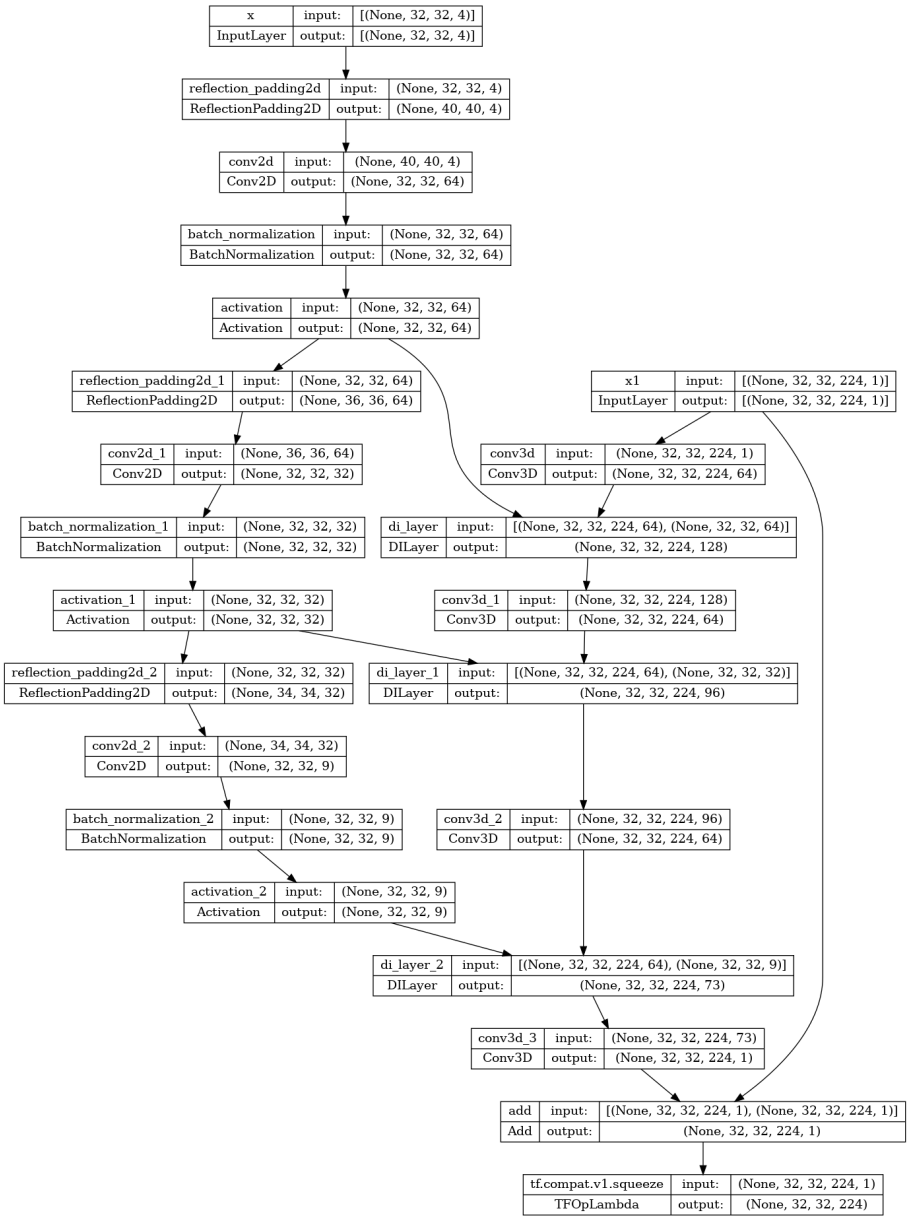
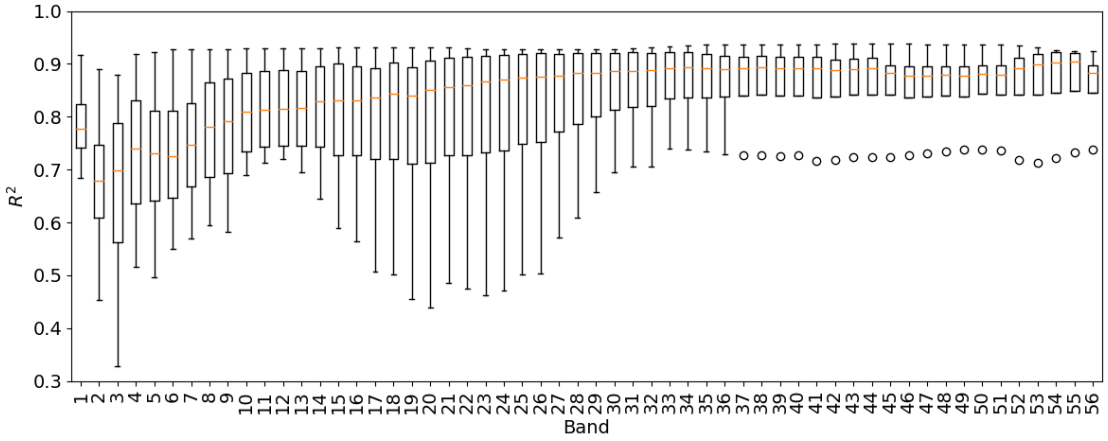
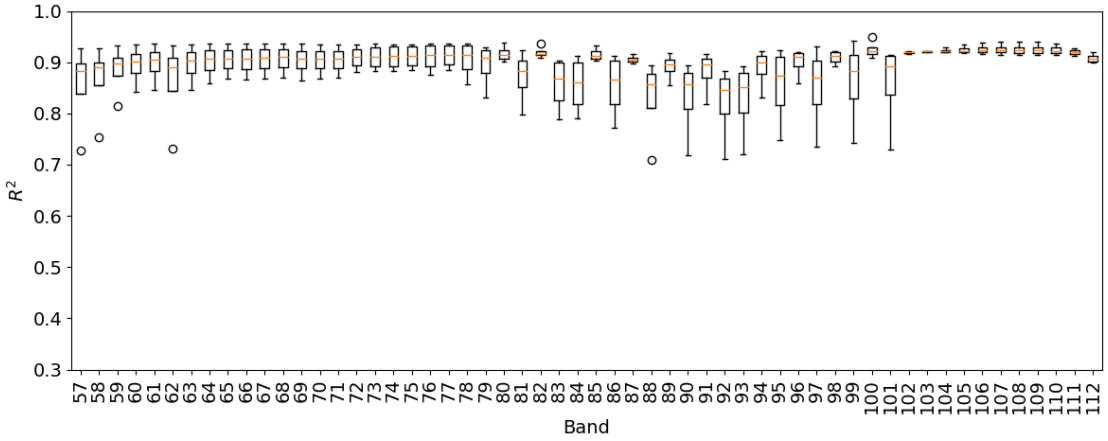
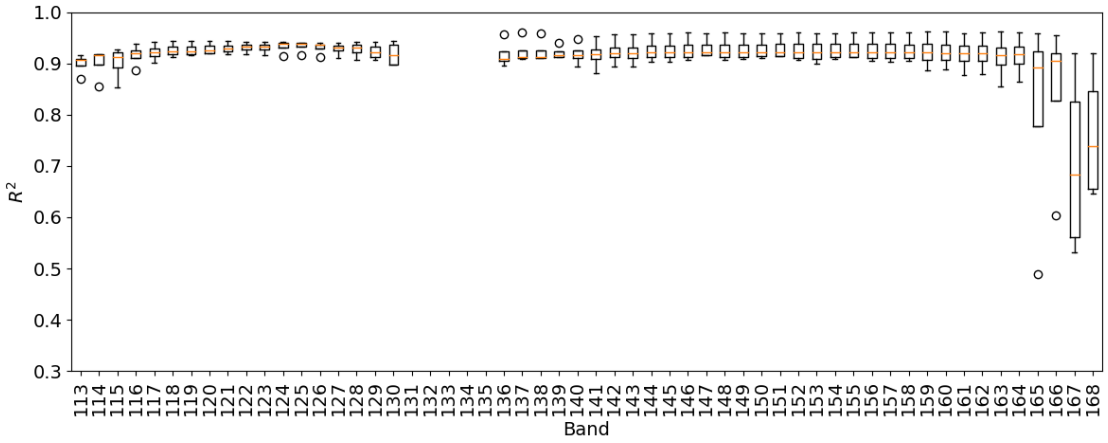


Figure A1: TensorFlow graph of supErMAPnet


1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
R^2
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56
Band


1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
R^2
57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112
Band


1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
R^2
113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168
Band

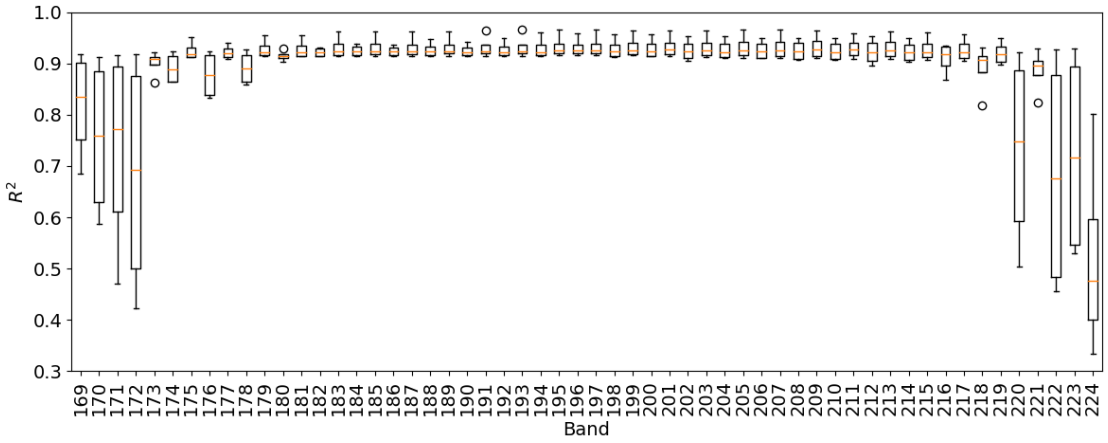
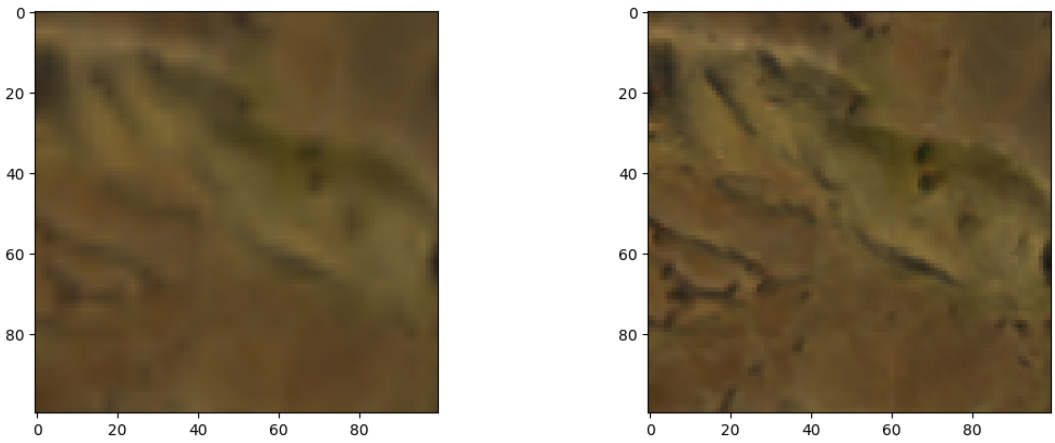
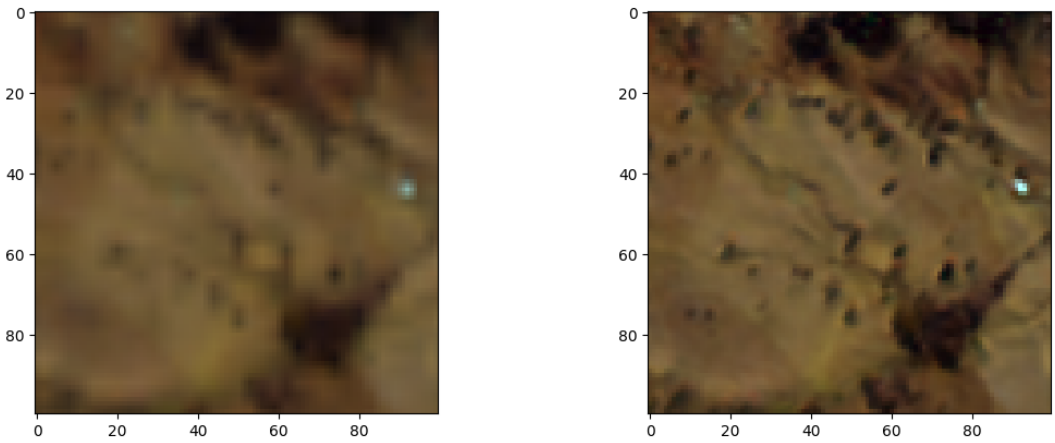


Figure A2: R^2 per band on four evaluation scenes



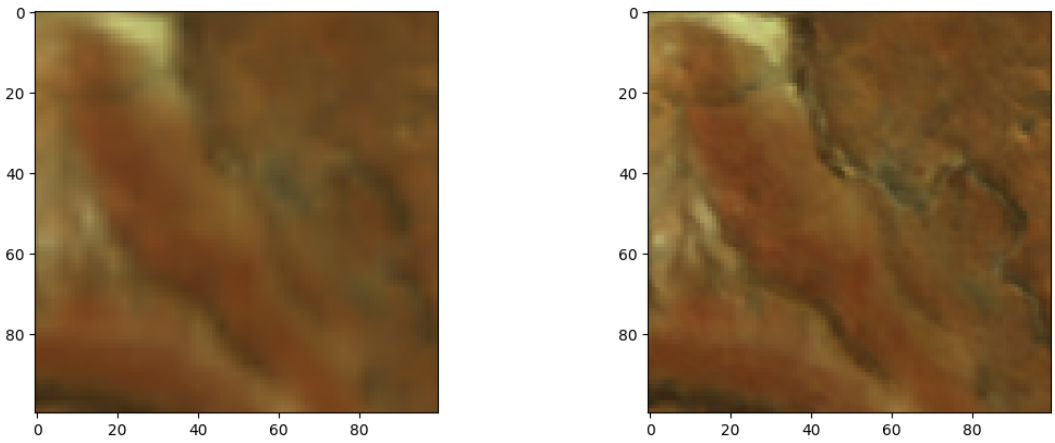
(a) Bilinear interpolation **(b)** supErMAPnet reconstruction

Figure A3: Area of 1 km^2 from scene Australia at 10 m $\times$ 10 m per pixel



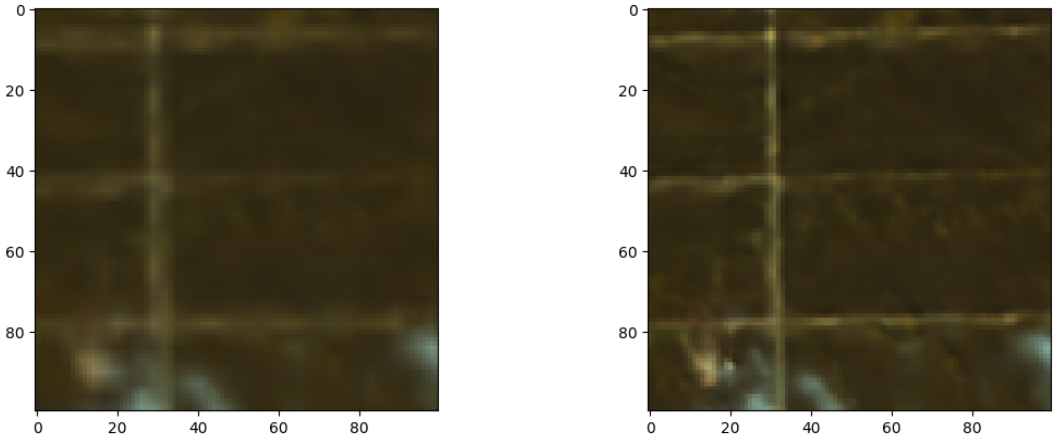
(a) Bilinear interpolation **(b)** supErMAPnet reconstruction

Figure A4: Area of 1 km^2 from scene Australia at 10 m $\times$ 10 m per pixel



(a) Bilinear interpolation **(b)** supErMAPnet reconstruction

Figure A5: Area of 1 km^2 from scene Namibia at 10 m $\times$ 10 m per pixel



(a) Bilinear interpolation **(b)** supErMAPnet reconstruction

Figure A6: Area of 1 km^2 from scene Peru at 10 m $\times$ 10 m per pixel

Declaration

"Ich versichere, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe, insbesondere sind wörtliche oder sinngemäße Zitate als solche gekennzeichnet. Mir ist bekannt, dass Zuwiderhandlung auch nachträglich zur Aberkennung des Abschlusses führen kann. Ich versichere, dass das elektronische Exemplar mit den gedruckten Exemplaren übereinstimmt."

Ort	Datum	Unterschrift