POLITECNICO DI TORINO

Master's Degree in Computer Engineering

A.Y. 2024/2025

Enhancing Sentinel-2 images using Super-Resolution

Supervisor
Prof. Paolo Garza

Candidate
Nicolò Vallania

Co-Supervisor
Dott. Edoardo Arnaudo

Graduation Session December 2025

Abstract

High-resolution satellite imagery plays a crucial role in remote sensing, enabling a wide range of applications, such as land-use monitoring, urban planning, precision agriculture, and environmental change detection. Super-resolution is a computer vision task aimed at reconstructing high-resolution images from their low-resolution counterparts. Applying it to multi-temporal, multi-spectral satellite imagery, such as that provided by Sentinel-2, can yield promising results; however, it is often not suitable as input for downstream tasks, such as land-cover segmentation.

Multi-task learning offers a viable alternative, providing improved regularization and better preservation of spatial structures when generating high-resolution land-cover maps. Moreover, information carried by shared feature representations proves more effective than relying on super-resolved RGB images.

This work makes use of two datasets: one specifically designed for super-resolution, and another suited for land-cover segmentation. After properly adapting the FLAIR-2 dataset, it investigates super-resolution architectures with the goal of extending and employing them to jointly perform segmentation in a single forward pass. The main objective is to obtain 4× super-resolved land-cover maps, supported by weighted combined loss functions.

The experiments involve fully convolutional networks (SRCNN, RCAN), generative adversarial networks (ESRGAN), and vision transformers (SwinIR). The study also explores the performance of pre-trained models, evaluates the benefits of parameter-efficient fine-tuning techniques (such as LoRA), and examines adversarial multi-task learning strategies.

Finally, quantitative and qualitative results are presented, showing that the proposed multi-task approach improves segmentation performance over single-task baselines, particularly in the preservation of fine spatial structures. A discussion of the advantages, limitations, and future research directions for this approach is also provided.

Contents

List of Tables

List of Figures

List of Acronyms

CLIP Contrastive Language-Image Pretraining. A model that learns visual concepts from natural language supervision. 52, 79

CNN Convolutional Neural Network. A deep learning model commonly used for image processing and computer vision tasks, capable of learning hierarchical features. 26

CORINE Coordination of Information on the Environment. European land cover classification program providing standardized nomenclature for land cover mapping. 35

cPSNR Color Peak Signal-to-Noise Ratio. Extension of PSNR to evaluate color images more accurately by considering multiple channels. 6, 7, 53, 58, 60–63

ESA European Space Agency. An intergovernmental organization dedicated to space exploration and Earth observation. 20, 33

ESRGAN Enhanced Super-Resolution Generative Adversarial Network. GAN-based SR model improving visual realism and high-frequency details. 37, 41, 44, 45, 57, 77

GAN Generative Adversarial Network. Deep learning architecture consisting of a generator and discriminator competing to produce realistic outputs. 26, 39, 77

GIS Geographic Information System. A system for storing, analyzing, and visualizing spatial and geographic data. 37

GT Ground Truth. Reference data used for evaluation of algorithms or models. In this work it refers to HR images and LC labels. 33, 50

HPC High-Performance Computing. Systems and techniques for large-scale computational tasks. 38

HR High Resolution. Refers to images with a high number of pixels, capturing fine spatial details. 7, 22, 34, 36, 40, 59, 63, 64, 77, 78

InSAR Interferometric Synthetic Aperture Radar. Technique using phase differences between SAR images to measure topography or deformation. 20, 78

IoU Intersection over Union. Metric for evaluating segmentation accuracy by comparing overlap between predicted and ground truth masks. 55, 64

JSON JavaScript Object Notation. Lightweight data-interchange format widely used for metadata storage and exchange. 34, 35

L1C Level-1C Sentinel-2 product. Top-Of-Atmosphere reflectance product from Sentinel-2 satellites. 33

LC Land Cover. Classification of Earth's surface types such as vegetation, water, or urban areas. 10, 40, 48, 65, 68

LLM Large Language Model. Neural network trained on massive text corpora to perform a wide range of natural language processing tasks. 67

LoRA Low-Rank Adaptation. Efficient technique for adapting large models with fewer parameters. 67, 68, 71, 78

LPIPS Learned Perceptual Image Patch Similarity. A perceptual metric measuring similarity between images based on deep features. 54

LR Low Resolution. Images with fewer pixels, often requiring super-resolution techniques for enhancement. 9, 10, 22, 25, 35, 38, 39, 43, 48–50, 57, 59, 62–64, 68, 77, 78

MAE Mean Absolute Error. Metric measuring the average magnitude of errors between predictions and targets. 50

MISR Multi-Image Super-Resolution. Technique that reconstructs a high-resolution image from multiple low-resolution observations. 19, 22

MLP Multi-Layer Perceptron. A fully connected neural network used for regression, classification, and feature transformation. 49

MS-SSIM Multi-Scale Structural Similarity Index Measure. Extension of SSIM considering image structures at multiple scales for quality assessment. 54

MSE Mean Squared Error. Metric measuring the average squared difference between predicted and actual values. 50

MSI Multi-Spectral Imaging. Acquisition of images in multiple wavelength bands for enhanced analysis. 21

MTL Multi-Task Learning. A paradigm where a single model learns multiple related tasks simultaneously, improving generalization. 10, 19, 23, 31, 40, 42, 48, 65, 68, 69

NAIP National Agriculture Imagery Program. U.S. program providing high-resolution aerial imagery for agriculture and land monitoring. 9, 33, 34

NIQE Naturalness Image Quality Evaluator. No-reference image quality metric based on natural scene statistics. 27, 54

NIR Near Infrared. Electromagnetic spectrum band used in remote sensing to detect vegetation and water content. 6, 21, 34, 37, 41, 61, 62, 68

NN Nearest Neighbour. Simple interpolation method using the value of the closest pixel. 25

PEFT Parameter-Efficient Fine-Tuning. Family of methods (e.g., LoRA, adapters) that adapt large models by training only a small subset of parameters. 67

PSNR Peak Signal-to-Noise Ratio. A common metric for image quality, measuring the ratio between maximum possible power and the power of noise. 6, 7, 53, 54, 58, 60–63, 77

RCAB Residual Channel Attention Block. Building block in RCAN architecture for enhancing features via attention mechanisms. 45, 46

RCAN Residual Channel Attention Network. Uses residual blocks with channel attention to improve feature representation for super-resolution. 41, 45, 65, 78

ReLU Rectified Linear Unit. Activation function introducing non-linearity while avoiding vanishing gradients. 66

RGB Red, Green, Blue color channels. Standard color representation used in imaging and display devices. 6, 10, 33, 34, 39, 41, 46, 48, 58, 60–62

RIR Residual in Residual. Architectural design to improve feature propagation and network stability. 26

RRDB Residual in Residual Dense Block. Architecture used in ESRGAN to improve feature propagation and stability. 44, 67, 78

RS Remote Sensing. The acquisition of information about Earth's surface and atmosphere from a distance, typically using satellite or aerial sensors. 30

RSTB Residual Swin Transformer Block. Core building block in SwinIR for image restoration tasks. 28

SAR Synthetic Aperture Radar. Active remote sensing technique using microwaves to capture surface structure independent of daylight. 9, 20, 78

SGD Stochastic Gradient Descent. Optimization algorithm commonly used to train machine learning and deep learning models by iteratively updating parameters based on random mini-batches. 61

SISR Single-Image Super-Resolution. The task of reconstructing a high-resolution image from a single low-resolution input. 19, 21

SLURM Simple Linux Utility for Resource Management. Workload manager for scheduling and running jobs on HPC clusters. 38

SMP Segmentation Model PyTorch. Library that provides implementations of several semantic segmentation networks using the PyTorch framework. 48

SR Super-Resolution. Task of enhancing the spatial resolution of images using computational techniques. 19, 25, 28, 30, 39, 48, 61, 65, 68, 72, 77

SRCNN Super-Resolution Convolutional Neural Network. One of the first deep learning approaches for single-image super-resolution, using three convolutional layers. 25, 41, 43, 57, 65, 68, 78

SSIM Structural Similarity Index Measure. A metric comparing local patterns of pixel intensities to assess perceived image quality. 51, 54, 77

SWIR Short-Wave Infrared. Electromagnetic spectrum band useful for material identification and environmental monitoring. 21, 41, 68

TCI True Color Image. Image composed of red, green, and blue bands, representing natural colors. 33

UTAE U-Net with Temporal Attention Encoder. A neural architecture for multi-temporal satellite image analysis, combining a U-shaped encoder with temporal self-attention; commonly used as a complete model for semantic segmentation or land cover classification. 7, 42, 49, 57, 59, 64, 68

VGG Visual Geometry Group network. CNN architecture known for deep layers and use in image classification. 51, 54

VHR Very High Resolution. Images with extremely fine spatial detail, often sub-meter. 7, 34, 38, 50, 62, 63, 78

Chapter 1

Introduction

Because of the wide range of potential applications and research areas, among which geology, climatology, urban planning, precision agriculture and defense, the necessity of high-resolution images of the Earth's surface has been increasing over the years.

Unfortunately, free high-resolution images are rare, often outdated or limited in coverage and quality. Therefore, efforts have focused on producing larger datasets and advanced techniques to overcome these issues, starting from freely available low-resolution images, without relying on commercial providers. The task that serves this purpose is called super-resolution.

The main goal of this thesis is to delve into the current state-of-the-art for super-resolution (SR), starting from single-image approaches (SISR) and extending the analysis to multi-image techniques (MISR), which leverage time series. Different architectures will be compared and some improvements will be proposed and evaluated, particularly focusing on multi-task learning (MTL).

This introduction aims to support the reader by providing an overview of the key terms and concepts.

1.1 The satellites

The Sentinels are a fleet of satellites delivering imagery to Copernicus [1], the Earth observation component of the European Union Space programme.

1.1.1 Sentinel-1

The Sentinel-1 mission [2] is part of the Copernicus Programme, developed by the European Space Agency (ESA). It consists of a constellation of C-band synthetic aperture radar (SAR) satellites: Sentinel-1A (launched in 2014) and Sentinel-1B (launched in 2016, currently inactive).

Unlike optical sensors, Sentinel-1 acquires data regardless of weather conditions or daylight, making it particularly valuable for continuous Earth observation. It provides high-resolution radar imagery with applications in land monitoring, agriculture, forestry, emergency response, flood mapping, and surface deformation studies through interferometric SAR (InSAR), with a revisit time of about 6 days at the equator (Figure 1.1).

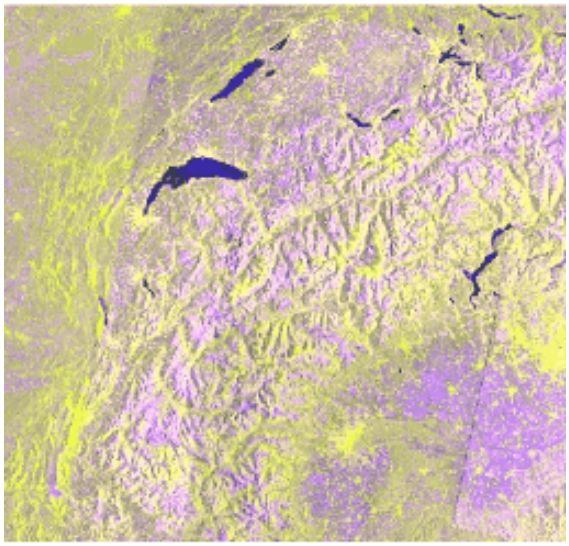

Figure 1.1. An example of SAR data sourced from Sentinel-1.

For the sake of this project, data sourced from Sentinel-1 will not be exploited.

1.1.2 Sentinel-2

The Sentinel-2 mission [3] complements Sentinel-1 with optical imaging capabilities. It includes Sentinel-2A (launched in 2015) and Sentinel-2B (launched in 2017), forming a twin-satellite constellation providing multi-spectral data for Earth surface monitoring.

Figure 1.2. An example of optical data sourced from Sentinel-2.

Sentinel-2 carries a MultiSpectral Instrument (MSI) with 13 spectral bands (from 443 nm to 2202 nm), spanning the visible, near-infrared (NIR), and short-wave infrared (SWIR) regions (Table 1.1). It is primarily designed for land monitoring, offering data for applications such as vegetation mapping, land cover classification, agriculture, water bodies, and disaster management. Its revisit time is approximately 5 days. An example is shown in Figure 1.2.

1.2 The tasks

As previously stated, throughout this work we will mainly focus on enhancing super-resolution, and exploiting it to improve land cover segmentation in a multi-task setting. The following subsections provide a clearer framework following this brief introduction.

1.2.1 Single-Image Super-Resolution

The task of Single-Image Super-Resolution (SISR) is aimed at reconstructing a high-resolution image from its low-resolution counterpart. Given a degradation process $\mathbf{I}^{LR} = D(\mathbf{I}^{HR})$, the objective is to learn a mapping $F_\theta : \mathbf{I}^{LR} \mapsto \hat{\mathbf{I}}^{HR}$ such that $\hat{\mathbf{I}}^{HR} \approx \mathbf{I}^{HR}$. As mentioned, developing efficient

Table 1.1. Spectral characteristics of Sentinel-2A and Sentinel-2B bands: central wavelength and bandwidth are expressed in nm, whereas spatial resolution is in m.

Band	**Sentinel-2A**		**Sentinel-2B**		**Spatial Res.**
	Central Wav.	Bandwidth	Central Wav.	Bandwidth	
1	442.7	20	442.3	20	60
2	492.7	65	492.3	65	10
3	559.8	35	558.9	35	10
4	664.6	30	664.9	31	10
5	704.1	14	703.8	15	20
6	740.5	14	739.1	13	20
7	782.8	19	779.7	19	20
8	832.8	105	832.9	104	10
8a	864.7	21	864.0	21	20
9	945.1	19	943.2	20	60
10	1373.5	29	1376.9	29	60
11	1613.7	90	1610.4	94	20
12	2202.4	174	2185.7	184	20

and effective super-resolution algorithms has become increasingly important in computer vision, because of its wide range of applications [4].

1.2.2 Multi-Image Super-Resolution

Multi-Image Super-Resolution (MISR) deals with datasets where multiple low-resolution (LR) images correspond to the same high-resolution (HR) patch. In this scenario, the model can extract more information and mitigate issues related to cloudy weather or low image quality.

1.2.3 Semantic Segmentation and Land Cover Segmentation

Semantic segmentation involves classifying each pixel of an image into predefined categories. In the context of remote sensing, land cover segmentation

aims to identify and map different surface types such as vegetation, water bodies, urban areas, and bare soil. Accurate land cover segmentation is crucial for applications in environmental monitoring, urban planning, and resource management. Similarly to super-resolution, revisits may be useful to deal with differences related to weather, seasons (especially for labels such as vegetation and agricultural lands) and brightness.

1.2.4 Multi-Task Learning

Multi-task learning (MTL) is a paradigm where a single model is trained to perform multiple related tasks simultaneously [5]. By sharing representations across tasks, MTL can improve generalization, leverage complementary information, and reduce overfitting. In remote sensing and computer vision, MTL is often used to jointly learn tasks such as semantic segmentation and object detection. This approach allows models to benefit from shared features while maintaining task-specific predictions via dedicated output heads. In the present work, MTL will be applied to super-resolution and land cover segmentation.

Chapter 2

Related works

This chapter serves as a comprehensive review and summary of the related literature. Starting from super-resolution foundations, different approaches and architectures will be presented, focusing solely on supervised learning, which is the core of the thesis.

2.1 Super-Resolution

Despite the existence of several algorithms based on sparse coding or interpolation, such as Nearest Neighbour (NN) and Bilinear, Bicubic, and Lanczos, deep learning and neural networks have proven to be more successful for SR.

The pioneer work was the Super-Resolution Convolutional Neural Network (SRCNN) proposed by Dong et al. [6] in 2014, relying on a simple architecture made of three convolutional layers (for patch extraction, non-linear mapping and reconstruction).

In 2016, Dong et al. [7] proposed a re-designed version called Fast Super-Resolution Convolutional Neural Network (FSRCNN), achieving an acceleration of more than 40 times: this was made possible by removing the upscaling bicubic interpolation, by exploiting an hourglass architecture made of shrinking and expanding layers, and by using a final deconvolution layer to obtain HR images, as depicted in Figure 2.1.

In the same year, Shi et al. [8] proposed ESPCN (Figure 2.2), which includes a novel sub-pixel convolution layer, instead of using deconvolution, to super-resolve LR images into HR space with very little additional computation.

In 2017, a paper by Lim et al. [9], inspired by ResNet, used residual blocks and claimed the benefit of removing batch normalization layer for SR task,

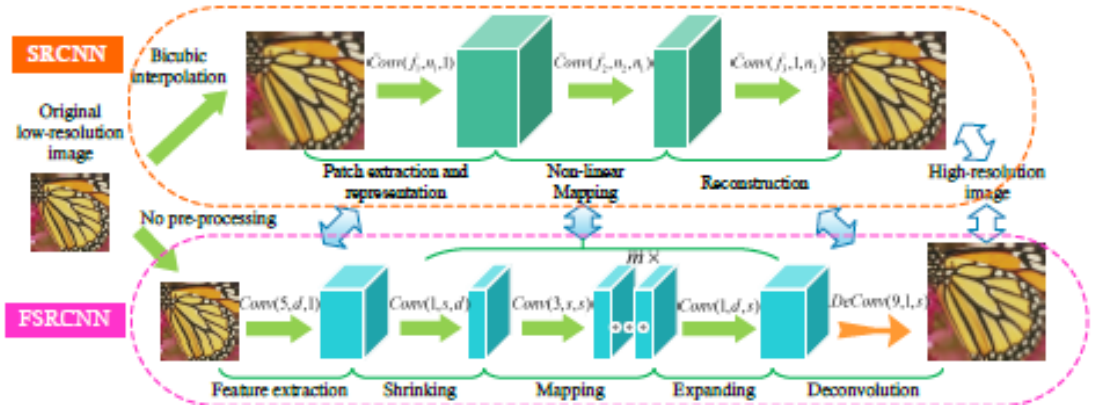

Figure 2.1. Comparison between SRCNN and FSRCNN architectures.

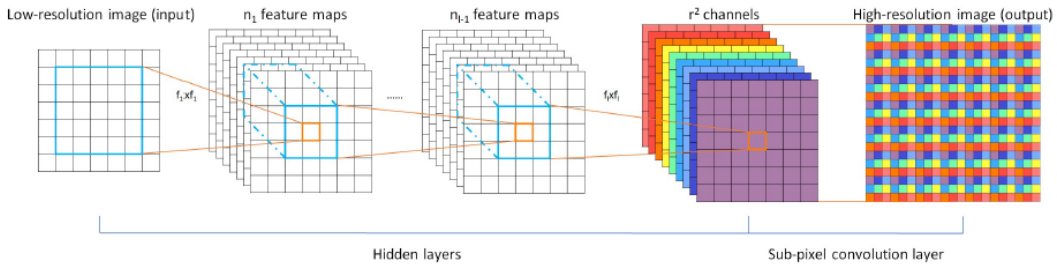

Figure 2.2. ESPCN network by Shi et al.

introducing EDSR (Figure 2.3).

In 2018, Ahn et al. [10] presented CARN (Figure 2.4), leveraging on cascading blocks to reduce the weight of computation.

Another excellent contribution came in the same year by Zhang et al. [11], who exploited Channel Attention mechanism to rescale channel-wise features by considering interdependencies among channels, and a RIR structure to bypass low-frequency information through skip connections.

Beyond CNN-based architectures, Generative Adversarial Networks (GAN) are regarded promising due to the increased realism of their outputs [12]. This is mainly due to the role of the discriminator, which has to distinguish real images and those produced by the generator: to do so, adversarial loss is employed.

After SRGAN (Figure 2.5), proposed in 2017 by Ledig et al. [13], again inspired by ResNet with skip-connection, Wang et al. [14] improved the former version by introducing ESRGAN, based on Residual-in-Residual Dense

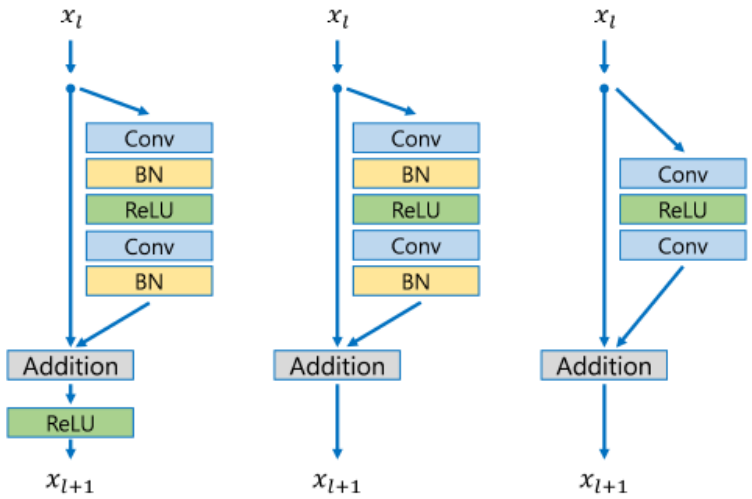

Figure 2.3. Comparison among residual blocks in ResNet, SRResNet and EDSR.

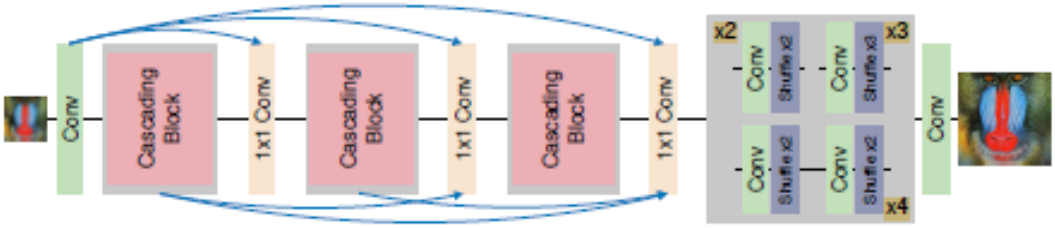

Figure 2.4. CARN architecture.

Blocks (without batch normalization), prediction of relative realness and improvement in perceptual loss, in order to reduce hallucinations.

For the following years, we cite Real-ESRGAN (Figure 2.6) by Wang et al. [15] and A-ESRGAN by Wei et al. [16]: the former aims to increase discriminator capability, stabilize training and enhance details, while removing artifacts, thanks to a U-Net discriminator with spectral normalization regularization, whereas the latter proposes a multi-scale attention U-Net discriminator and achieves better visual quality and results on NIQE metrics than contemporary state-of-the-art.

Another important approach in the field is represented by transformer-based architectures. Following the groundbreaking paper "Attention Is All You Need" by Vaswani et al. [17], transformers (which are depicted in Figure 2.7) gained popularity in computer vision thanks to Dosovitskyi et al.

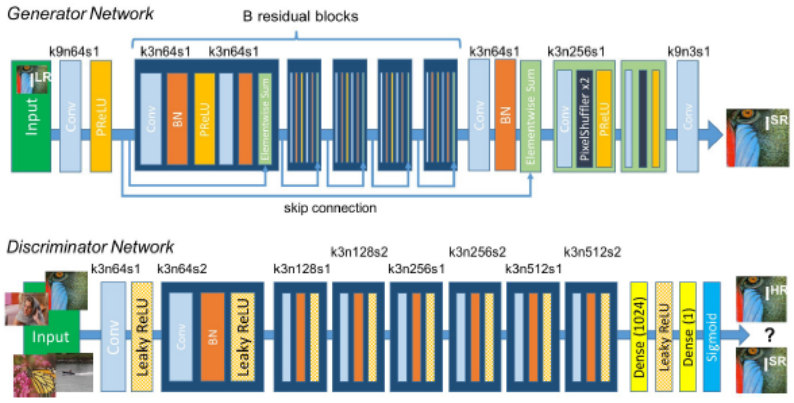

Figure 2.5. SRGAN architecture, leveraging adversarial learning and the contribution of the discriminator.

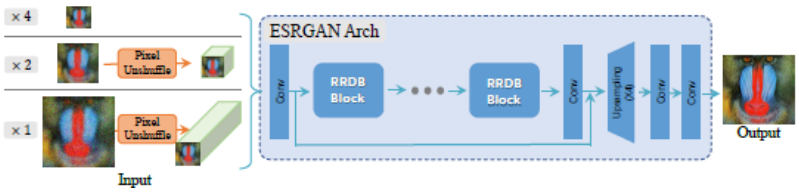

Figure 2.6. Real-ESRGAN architecture by Wang et al.

[18] in 2020.

One year later, SwinIR was presented by Liang et al. [19] as a new state-of-the-art model for image restoration. Based on the Swin Transformer, this new architecture consists of three parts: shallow feature extraction, deep feature extraction (made of RSTB, Residual Swin Transformer Blocks) and high-quality image reconstruction. The same approach was improved by Zhang et al. [20] in 2023, with their SwinFIR: among their contributions, they made the model faster by using Fast Fourier Convolution.

After SwinIR, many other methods focused on transformers for SR, such as ACT [21], which combines transformer and convolutional branches and introduces a cross-scale attention module, RGT [22], which combines recursive-generalization self-attention and local self-attention (hybrid adaptive integration), HAT [23], which combines channel attention and self-attention to activate more pixel for reconstruction (Figure 2.8), DRCT [24], mitigating spatial information loss and stabilizing information flow, and SRFormer [25], introducing permuted self-attention.

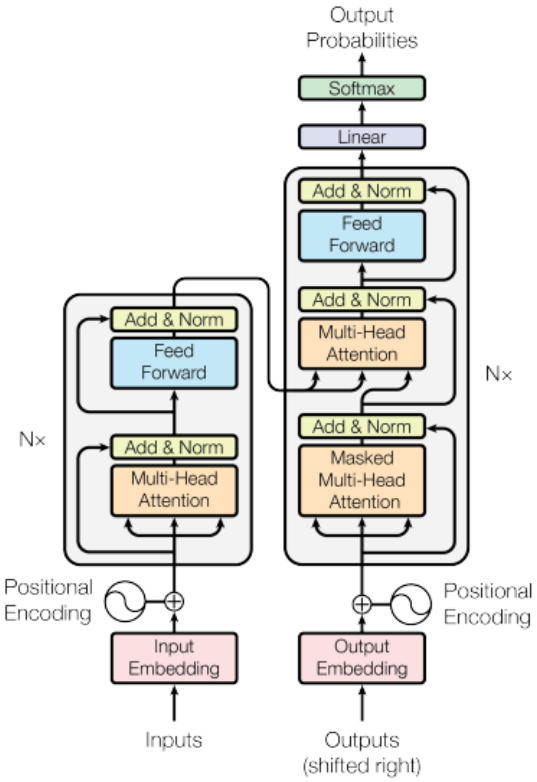

Figure 2.7. Transformer model architecture, by Vaswani et al.

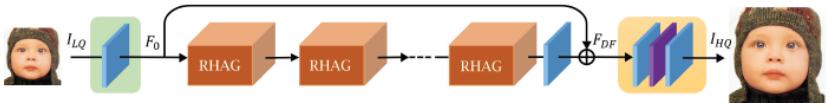

Figure 2.8. HAT architecture by Chen et al.

Finally, due to the increasing popularity of Diffusion Models in image generation, some studies delved into applying these architectures to super-resolution. For example, SR3 by Saharia et al. [26], in 2021, achieved highly photo-realistic outputs and visual quality, despite suffering from bias issues.

2.2 Super-Resolution for Remote Sensing

The increasing interest in Earth observation led to the need for higher-resolution imagery. SR for remote sensing poses its own challenges, due to multispectrality, scarcity of free ground-truth images, attention to spectral and geometrical preservation.

In literature, both single-image and multi-image approaches have been applied to RS; here, we cite some methods specifically tailored to Sentinel-2.

In 2018, Lanaras et al. [27] described an EDSR network to super-resolve the lower-resolution (20 m and 60 m) bands to 10 m, guided by higher-resolution bands themselves.

In 2021, Gong et al. [28] proposed Enlighten-GAN, outperforming state-of-the-art methods thanks to the enlighten blocks (Figure 2.9), and employing cropping-and-clipping strategy to avoid the seam line.

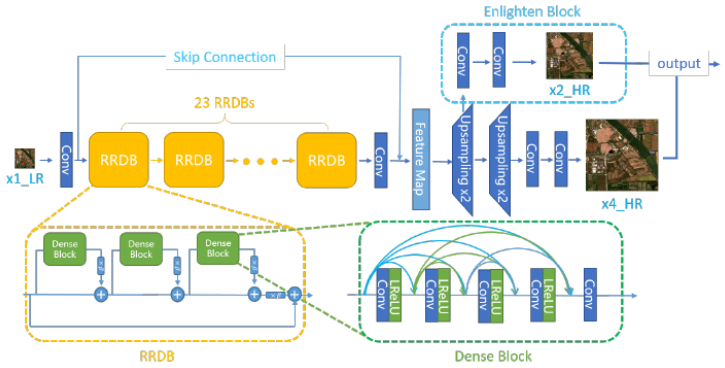

Figure 2.9. Enlighten-GAN architecture

In 2023, a method proposed by Tarasiewicz et al. [29] demonstrated how combining both multitemporal and multispectral data leads to more effective super-resolution of Sentinel-2 images.

2.3 Semantic Segmentation and Land Cover

Among the foundational works in semantic segmentation, the U-Net architecture [30] introduced the encoder-decoder structure with skip connections (Figure 2.10), which has since been widely adopted in remote sensing, as well.

More recently, Transformer-based models such as Swin-Unet [31] have

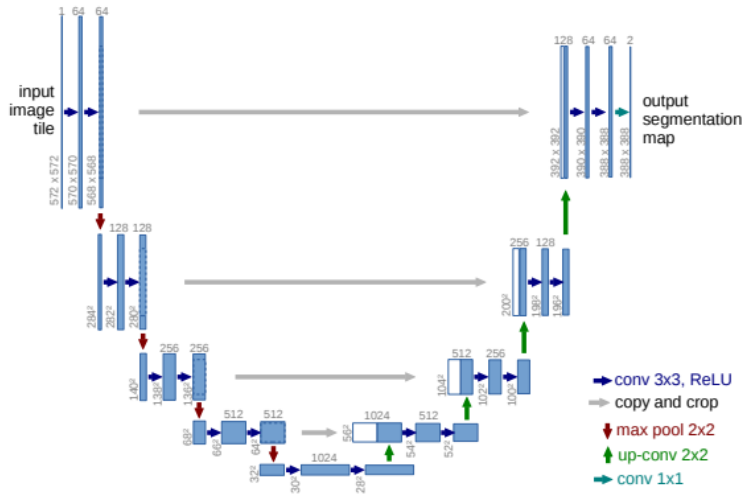

Figure 2.10. The original U-Net architecture

gained popularity, leveraging hierarchical self-attention to better capture spatial dependencies. Another promising approach came from Garnot et al. [32], who proposed a method for panoptic segmentation of satellite image time series (SITS). Their model leverages a temporal self-attention encoder to capture spatio-temporal patterns.

2.4 Multi-task Learning

Multi-task learning (MTL) was introduced by Caruana [5], who demonstrated that jointly learning multiple related tasks in a single network can improve generalization by leveraging shared representations. Ruder [33] provides a comprehensive survey of MTL, discussing several key benefits: improved generalization by transferring knowledge across related tasks, reduced overfitting, enhanced ability to learn robust features, and often faster convergence during training. Additionally, another approach introduces task-dependent uncertainty weighting for MTL, allowing a network to automatically balance losses for heterogeneous tasks, as shown by Kendall et al. [34].

Chapter 3

Dataset

As stated previously, the quantity of extensive, freely available datasets for remote sensing application, and particularly to perform both super-resolution and land cover segmentation, is limited. The present work was carried out using two different datasets: S2NAIP [35] and FLAIR-2 [36].

3.1 S2NAIP

The former consists of aligned NAIP, Sentinel-2, Sentinel-1, and Landsat images spanning the entire continental US. NAIP is the high-resolution RGB target, having a resolution of 512×512 pixels. For the scope of this work, only NAIP and Sentinel-2 were used, since the land cover ground truth (GT) provided by this dataset is at a lower resolution than required (i.e. 10 m/pixel, whereas the goal is to obtain a land cover map with a resolution of 2.5 m/pixel). Because of the huge number of samples in this dataset, exceeding 44 million, the provided urban subset was used, cutting the total to about 1.1 million samples. This allowed for faster training and reduced the number of rural areas, which would have generated a strong bias. Some examples can be observed in Figure 3.1.

Sentinel-2: This dataset uses the Sentinel-2 L1C imagery. Models that accept RGB as input rely on the TCI files provided by ESA, which contain an 8-bit image that was normalized by ESA to the 0-255 range. The image is then pre-processed by dividing the 0-255 RGB values by 255, and retaining the RGB order. On the other hand, for non-TCI bands, the 16-bit data is divided by 8160 and then clipped to 0-1.

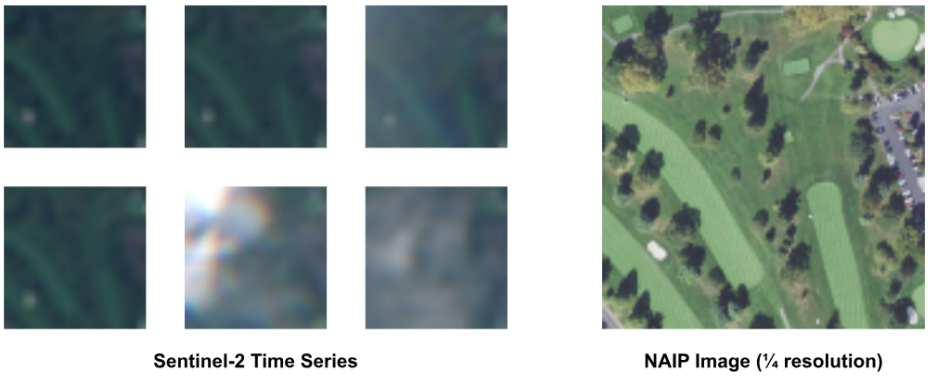

Figure 3.1. Samples of Sentinel-2 revisits and NAIP high-resolution target from S2NAIP by AllenAI.

NAIP: The NAIP images, which represent the HR ground truth in this dataset, are 25% of the original NAIP resolution; this means that each image (consisting of RGB channels only) is downsampled to 128×128px.

3.2 FLAIR-2

This dataset is sampled in France, countrywide, and is composed of over 20 billion annotated pixels of very high resolution (VHR) aerial imagery at 0.2 m/pixel, acquired over three years and different months. Aerial imagery patches consist of 5 channels (RGB, NIR and aerial) and have corresponding annotations for 19 semantic classes, 13 of which are used for this work. For low-resolution, Sentinel-2 time series with 10 spectral band are provided as 40×40 super-patches, allowing for wider spatial and temporal context, as well. More than 50,000 Sentinel-2 acquisitions with 10 m/pixel are available.

Aerial imagery (IMG): Each aerial patch contains five channels: blue, green, red, near-infrared (NIR), and elevation. These are stored as floating point values between 0 and 1. Metadata for each patch (acquisition time, location, altitude, camera type) is provided in a dedicated JSON file.

Sentinel-2 imagery (SEN2): For each aerial patch, corresponding Sentinel-2 super-areas are provided as time series of reflectance data in 10 spectral bands. These are stored in 4D NumPy arrays of shape $T \times C \times H \times W$, where T is the number of acquisitions, C the number of spectral channels,

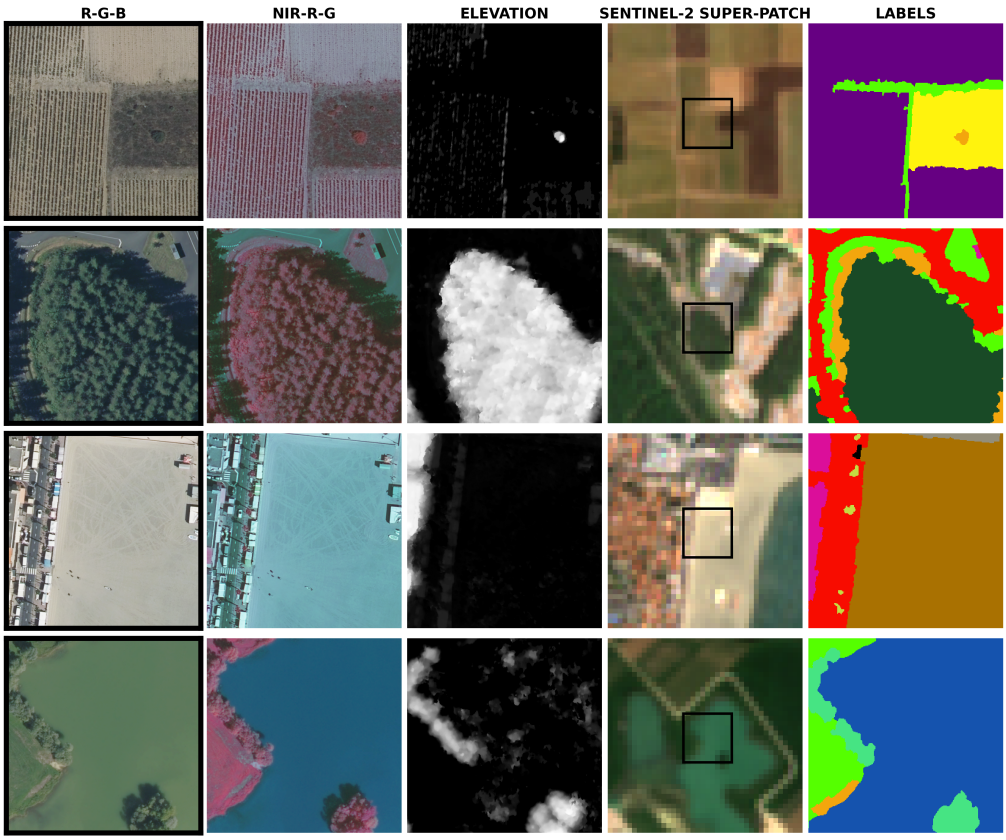

Figure 3.2. Samples from FLAIR-2 dataset: LR patch is the central crop of Sentinel-2 super-patch.

and H, W the spatial dimensions. Data are stored in 16-bit unsigned integer format. Associated files include:

– *Masks:* snow and cloud probability maps (0 - 100 scale).

– *Products:* text files listing acquisition details (platform, date, orbit, tile).

– *Centroids:* JSON files mapping aerial patch IDs to Sentinel-2 coordinates, used for extracting super-patches (of default size 40×40 pixels, but 32×32 was used for the present work).

Annotations (MSK): Semantic segmentation masks are provided as single-channel 8-bit images, with values ranging from 1 to 19 (Table 3.1). These represent land cover classes, consistent with the CORINE Land Cover nomenclature. Annotations are limited to aerial patch boundaries and correspond to the acquisition date of the aerial imagery. Temporal inconsistencies may occur for dynamic features (e.g., riverbanks), as changes may happen between

aerial and satellite acquisitions.

The dataset is mainly designed for land cover semantic segmentation and multi-modal learning, leveraging both HR aerial imagery and multi-temporal, multi-spectral Sentinel-2 data. Some samples are shown in Figure 3.2.

Table 3.1. Class distribution in the considered dataset (derived from FLAIR-2 challenge). Classes from 13 to 18 were not evaluated, due to their extremely low frequency, and were treated as 'other'.

Class	**Value**	**Freq. (%)**
building	1	8.14
pervious surface	2	8.25
impervious surface	3	13.72
bare soil	4	3.47
water	5	4.88
coniferous	6	2.74
deciduous	7	15.38
brushwood	8	6.95
vineyard	9	3.13
herbaceous vegetation	10	17.84
agricultural land	11	10.98
plowed land	12	3.88
swimming pool	13	0.01
snow	14	0.15
clear cut	15	0.15
mixed	16	0.05
ligneous	17	0.01
greenhouse	18	0.12
other	19	0.14

Chapter 4

Methodology

This chapter describes neural networks proposed to address super-resolution, land cover segmentation and multi-task learning, as well as assumptions, metrics and techniques that were adopted in order to mitigate issues related to image quality and intrinsic problems of the tasks.

Because of the high number of samples provided by S2NAIP, leading to long training time, this dataset was mainly used to compare super-resolution networks, image quality and investigate the performance boost due to the NIR channel. Moreover, since pre-trained ESRGAN models based on RRDB-Net were available for download, they offered the chance to easily evaluate fine-tuning on FLAIR-2.

On the other hand, U-Net and UTAE were used as baselines for land cover segmentation of low resolution images, derived from Sentinel-2 40×40 patches. The network outputs were then cropped to obtain a 10×10 map.

4.1 Tools, software and libraries

In order to perform data processing, model development and training, and geospatial analysis, several technologies were employed.

Firstly, QGIS (Quantum GIS) was used to visualize rasters, both for multispectral data and for segmentation labels, providing an interface to work with satellite imagery. Additionally, Rasterio was essential to read and manipulate multispectral data in the Python environment.

Tasks and experiments were mainly carried out using Python and PyTorch Lightning, which provides abstractions and wrappers to avoid writing boilerplate code for data modules, training, validation, checkpointing, testing, offering many options for customization. Moreover, other Python libraries,

such as Numpy, Matplotlib, OpenCV, Sklearn, Scipy, were employed for array manipulation (e.g., reshaping, flattening), visualization (e.g., histograms, image grids) and image processing (e.g., interpolation, resizing).

Computation related to experiments was executed on a HPC cluster, provided by LINKS Foundation. Specifically, jobs ran on a NVIDIA GeForce RTX 2080 Ti GPU, with 11 GB of memory, and 64 GB of RAM. Furthermore, SLURM was used as a workload manager to schedule tasks efficiently on the cluster. These tools altogether ensured fast data processing, model training and inference, which was essential to handle the large number of experiments.

4.2 Addressing dataset limitations and preprocessing

Satellite imagery usually comes with specific challenges, and, despite being qualitatively and quantitatively fine and complete, the present dataset is no exception.

Cloud filtering: Clouds often have a negative influence on the model ability to perform computer vision tasks. Cloud filtering is applied to the Sentinel patches using the associated cloud and snow masks. Each patch contains a mask indicating the presence of clouds or snow for each pixel and acquisition date. Therefore, observations affected by clouds or snow are excluded from the dataset, retaining only cloud-free images. This ensures that the model is trained and evaluated on clean satellite imagery, while further processing such as monthly averaging or data augmentation is performed only on these filtered observations. In other cases where this is not feasible, using revisits may be greatly beneficial in improving the quality of the reconstruction.

Cropping: While segmentation labels match VHR ground-truth, LR super-areas are larger, meaning they can provide context for our application, but eventually need cropping. As mentioned, among the provided metadata, there are centroids, which are particularly useful to obtain the correctly centered crop and, finally, stitch patches for visualization. To sum up, super-areas of variable size are first cropped to get a 32×32 patch, which is fed to the network. The output is then cropped again, to match the target

size (10×10 for LR tasks and 40×40 for SR tasks). An example is shown in Figure 4.1.

Figure 4.1. An example of cropped output: the original super-resolved 128×128 image (on the left) and the cropped 40×40 version (on the right).

Normalization and histogram matching: In order to comply with the guidelines for pre-processing Sentinel-2 data, bands values were divided by a factor 10000. Because of the evident differences in brightness and colours distribution when compared to S2NAIP imagery (Figure 4.2, 4.3, 4.4), further pre-processing was needed when using pre-trained GAN models. In particular, after computing mean and standard deviation for each RGB channel for the two datasets, a transformation was applied to match colour distribution: this ensured a better correspondence and led to more visually pleasing and coherent results when testing pre-trained ESRGAN. The result of the transformation is shown in Figure 4.5.
Let x be an input tensor with c channels, assume we know the per-channel statistics (mean and standard deviation) of the two datasets *FLAIR* and *S2NAIP*:

$$\mu^c_{\text{flair}},\ \sigma^c_{\text{flair}},\quad \mu^c_{\text{s2naip}},\ \sigma^c_{\text{s2naip}} \tag{4.1}$$

for each channel c.
The transformation applied to each pixel/channel is:

$$y^c = \frac{x^c - \mu^c_{\text{flair}}}{\sigma^c_{\text{flair}}} \, \sigma^c_{\text{s2naip}} + \mu^c_{\text{s2naip}} \tag{4.2}$$

Equivalently, in compact vector form:

$$\mathbf{y} = \mathbf{D}_{\text{s2naip}} \, \mathbf{D}^{-1}_{\text{flair}} \left(\mathbf{x} - \boldsymbol{\mu}_{\text{flair}} \right) + \boldsymbol{\mu}_{\text{s2naip}} \tag{4.3}$$

where

$$\mathbf{D}_{\text{flair}} = \text{diag}(\sigma^1_{\text{flair}}, \ldots, \sigma^c_{\text{flair}}), \quad \mathbf{D}_{\text{s2naip}} = \text{diag}(\sigma^1_{\text{s2naip}}, \ldots, \sigma^c_{\text{s2naip}}) \tag{4.4}$$

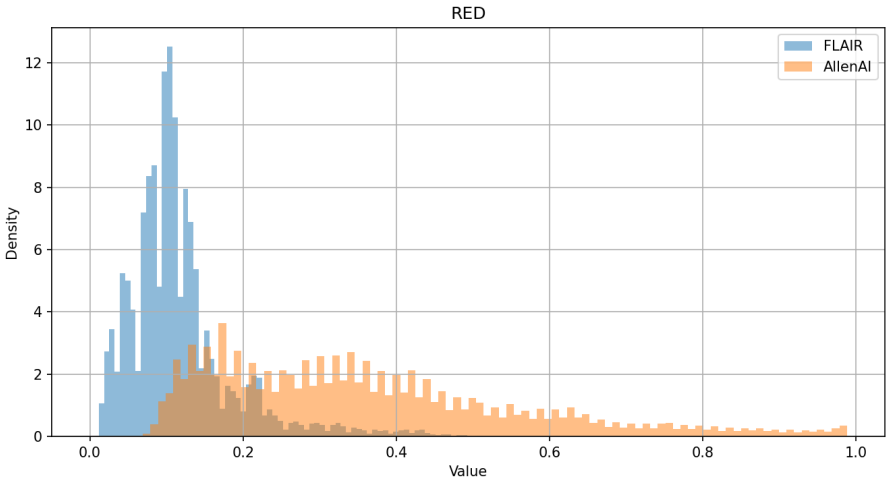

Figure 4.2. Histograms showing the distribution of the red band.

Downsampling: Given the very high resolution of the labels and the ground truth of the images (0.2 m/pixel), some tough downsampling was needed. In spite of this, the visual quality of the ground truth is still high, and shapes and colours are quite well defined, as shown in Figure 4.6. However, some information may be lost on behalf of labels, especially in terms of finer details (e.g., roads).

The target LC map was downsampled from 512×512 to 10×10 for experiments starting from LR images and to 40×40 for MTL and HR segmentation, in order to match output size (Figure 4.7). This was achieved by applying an interpolation method based on majority voting: each pixel in the resulting downsampled map is the mode of the corresponding area in the original map.

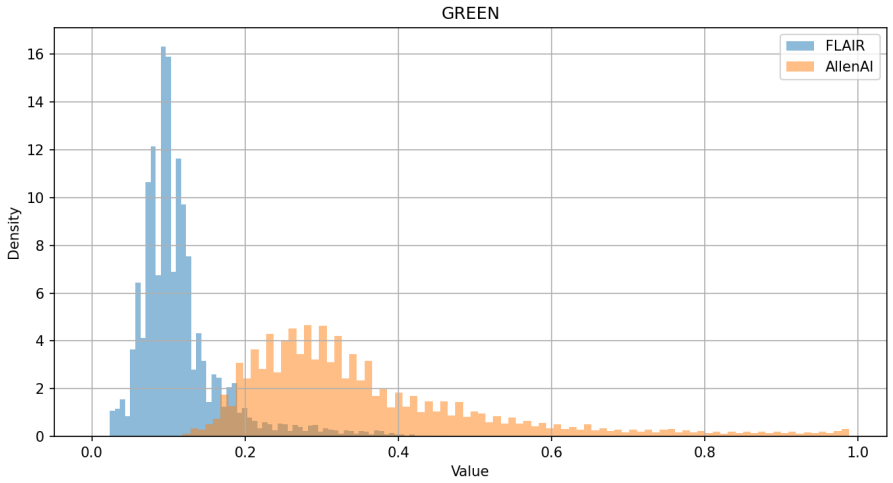

Figure 4.3. Histograms showing the distribution of the green band.

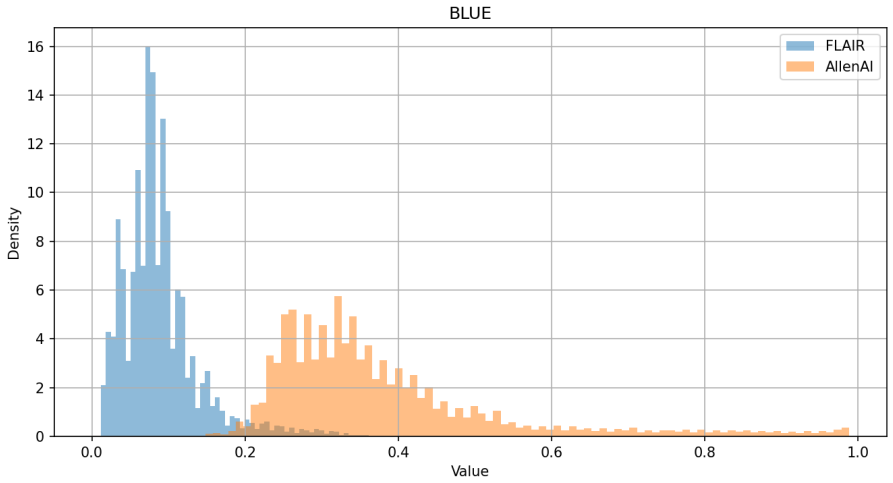

Figure 4.4. Histograms showing the distribution of the blue band.

4.3 Neural networks

After carefully reviewing the existing literature and state-of-the-art architectures, four models were selected for both super-resolution and, therefore, multi-task learning: SRCNN, ESRGAN, RCAN and SwinIR. They were all evaluated with RGB, RGB + NIR and, finally, all available bands as input channels, including SWIR bands resampled to 10 m/pixel; revisits were treated as channels (hence, using $C_{\text{img}} \times N_{\text{rev}}$ as the input dimensionality,

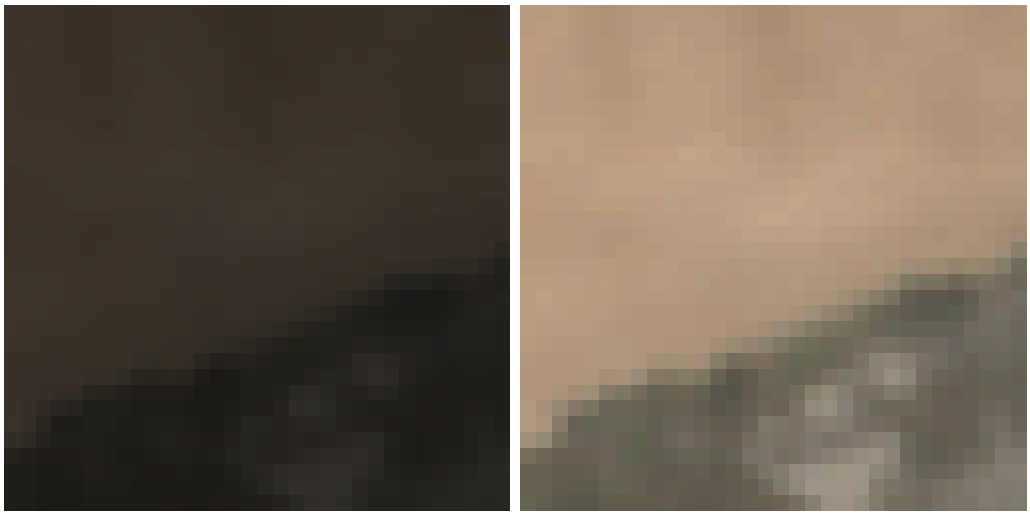

Figure 4.5. A sample from FLAIR-2 before the transformation (on the left) and after the transformation (on the right). The shift in brightness is particularly evident, while the colours are adapted to the target dataset distribution.

Figure 4.6. On the left, the downsampled image (2.5 m/pixel). On the right, the original VHR image (0.2 m/pixel).

where C_{img} denotes the number of image channels and N_{rev} the number of revisits), as well. In MTL, segmentation head receives upsampled features as input.

For segmentation, a simple U-Net was used for both low-resolution and high-resolution images, whereas UTAE was evaluated to account for revisits. These experiments were crucial in order to obtain the upper and lower bounds

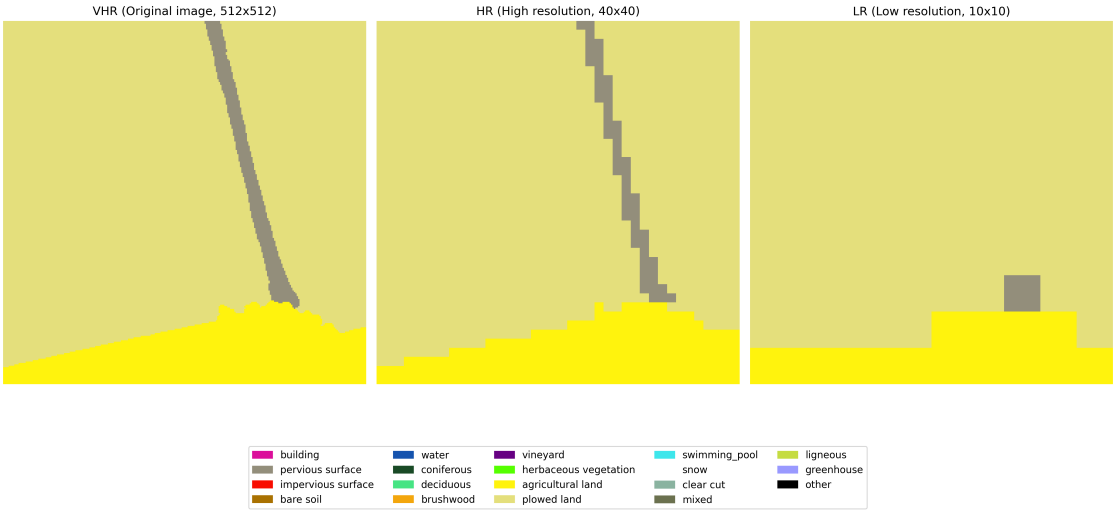

Figure 4.7. A visual example of the labels obtained by downsampling the VHR (512×512) target (on the left), to 40×40 (in the middle), and 10×10 (on the right). As expected, finer details, such as roads (coloured in grey), could not be fully preserved when reducing resolution by about 50 times.

of the baseline: HR image (2.5 m/pixel) and LR time series (10 m/pixel).

4.3.1 SRCNN

As the simplest architecture, SRCNN was already explored in the context of S2NAIP. Taking as input low-resolution images and, optionally, a reference frame (computed as the pixel-wise median across revisits), it consists of a shared double convolution encoder, followed by optional mask encoding. Encodings are stacked across channels and passed through a fusion module, which includes convolutional and residual layers. This effectively aggregates temporal information.

The super-resolution module uses a PixelShuffle-based upsampler to increase spatial resolution, followed by resizing to a fixed output size.

This model, represented in Figure 4.8, offers some clear advantages in terms of simplicity, training and inference speed and size, but performance are considered far from being state-of-the-art.

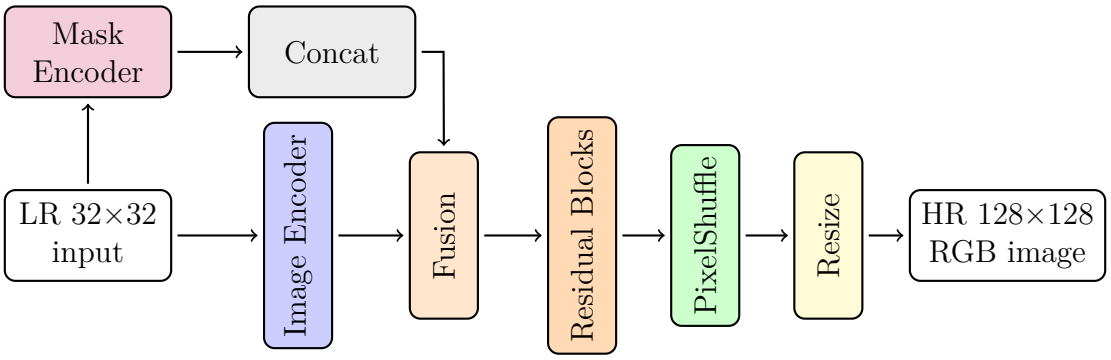

Figure 4.8. Representation of chosen SRCNN architecture, including image encoder, mask encoder, fusion, residual blocks, and PixelShuffle super-resolution head.

4.3.2 ESRGAN

The generator of the selected ESRGAN model, SSR-RRDBNet (Figure 4.10), is an enhanced super-resolution architecture based on the Residual-in-Residual Dense Block (RRDB, Figure 4.9). Firstly, the model optionally applies pixel unshuffle, a technique that rearranges spatial resolution into the channel dimension, and an initial convolution to extract features, followed by a trunk composed of multiple RRDB blocks.

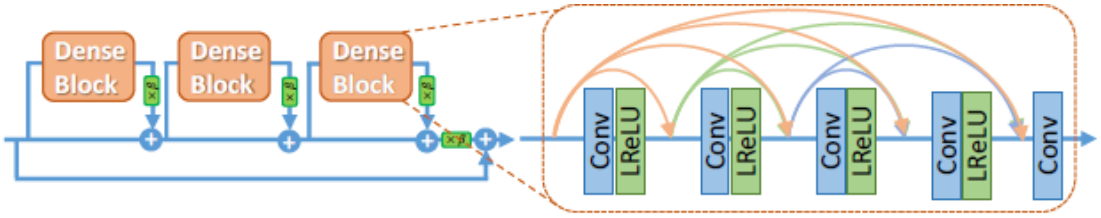

Figure 4.9. Residual in Residual Dense Block used in ESRGAN generator.

Each RRDB consists of three Residual Dense Blocks (RDBs), where dense connections and residual scaling improve feature reuse and training stability. The upsampling path performs successive interpolation-based scaling operations, followed by convolutional refinement. Depending on the desired output resolution (e.g., $\times 8$, or $\times 16$), extra upsampling layers are added.

On the other hand, the discriminator consists of a U-Net–style architecture (Figure 4.11) with spectral normalization, used in Real-ESRGAN [15].

It combines downsampling and upsampling paths with optional skip connections, allowing it to capture both global structures and fine details.

Benefitting from adversarial training and considerably larger size, ESRGAN offers higher visual quality, at the cost of long training and inference time.

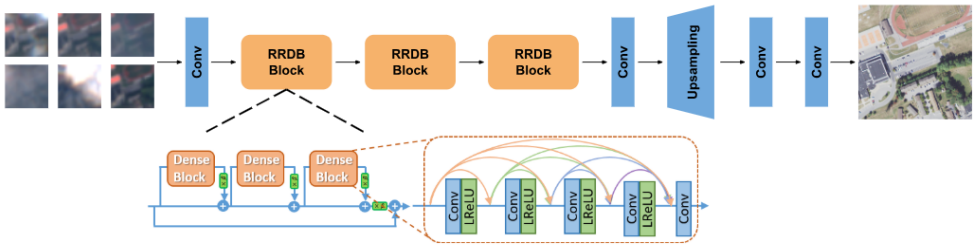

Figure 4.10. Architecture of RRDB-based generator of ESRGAN.

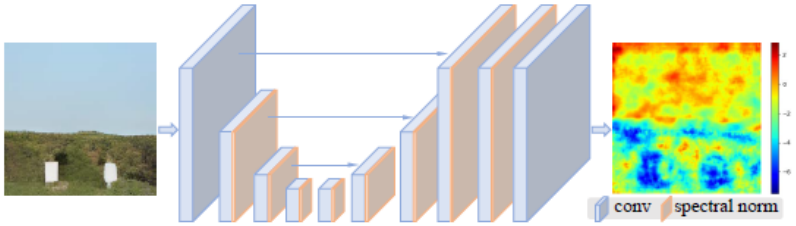

Figure 4.11. Architecture of U-Net based discriminator.

4.3.3 RCAN

In order to evaluate state-of-the-art models in the context of remote sensing applications, another convolutional model was selected: RCAN [11]. Feature extraction begins with a 3×3 convolutional layer that expands the channel dimension to 64 feature maps.

The core of the architecture consists of multiple Residual Groups, each composed of several Residual Channel Attention Blocks (RCABs). Each RCAB (Figure 4.13) contains two 3×3 convolutions followed by a Channel Attention mechanism (Figure 4.12), implemented by global average pooling and two 1×1 convolutions. This attention allows the network to dynamically weigh each channel based on its contextual relevance.

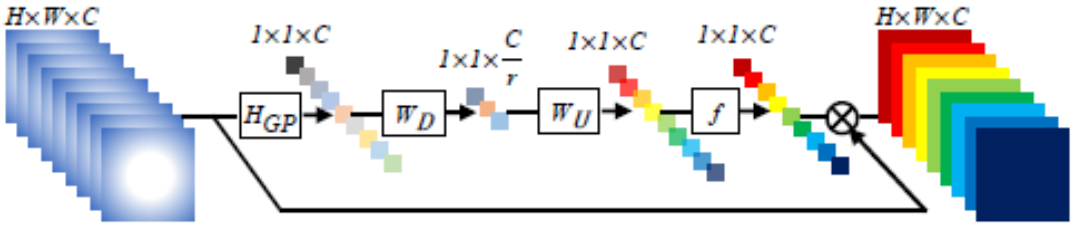

Figure 4.12. Channel attention mechanism

The extracted features are combined with the initial features through a global residual connection, which facilitates optimization even in very deep networks.

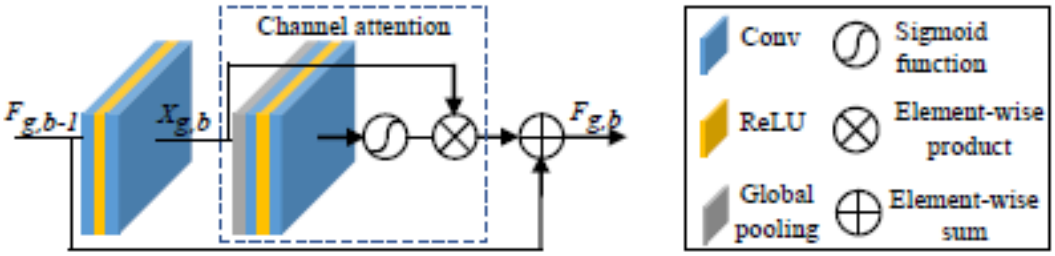

Figure 4.13. Residual Channel Attention Blocks

For the super-resolution task, the network employs an upsampling module followed by a final convolution to reconstruct a high-resolution (HR) 128×128 RGB image.

Among purely convolutional networks, this model, shown in Figure 4.14, represents a state-of-the-art solution, thanks to RCAB, although it does not achieve the visual quality provided by GANs.

4.3.4 SwinIR

In SwinIR (Figure 4.15), the input image is first normalized using predefined channel means, if available, and then scaled according to a specified range (e.g., 1 or 255). The network begins with a shallow feature extraction block (`conv_first`) that maps the input channels to a high-dimensional embedding space using a 3×3 convolution.

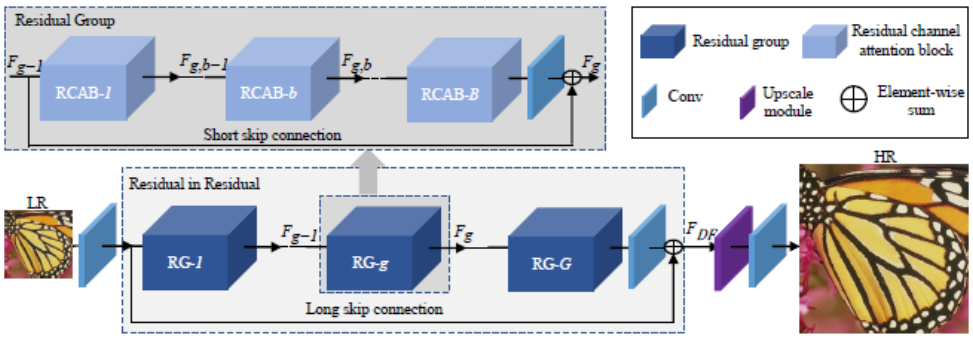

Figure 4.14. RCAN architecture

The core of the architecture consists of multiple Residual Swin Transformer Blocks (RSTBs), arranged in stages defined by the `depths` and `num_heads` hyperparameters. Each block processes non-overlapping image patches using shifted window-based self-attention (`window_size`), enabling both local and global context modeling. After feature extraction, the representation is reassembled in image form using the `patch_unembed` module.

The stage of high-resolution image reconstruction varies depending on the chosen upsampling method (options are `pixelshuffle` for classical SR, `nearest+conv` for real-world SR, `pixelshuffledirect` to save parameters, or none, for other tasks, such as image denoising).

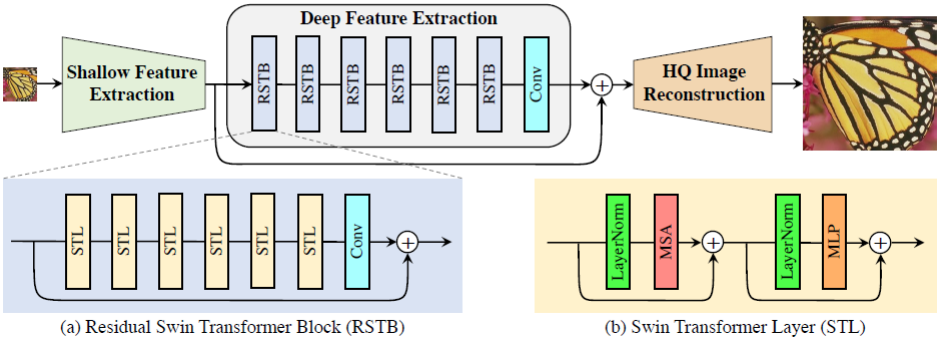

Figure 4.15. SwinIR architecture

4.3.5 Segmentation head for SR networks

As mentioned, in order to adapt the previously described SR networks to perform MTL and output a LC map, a segmentation head was appended (Figure 4.16). This was achieved by adding a convolutional layer followed by an activation function and a final 1×1 convolutional layer. The input of the segmentation head consists of upsampled features extracted before the super resolution head.

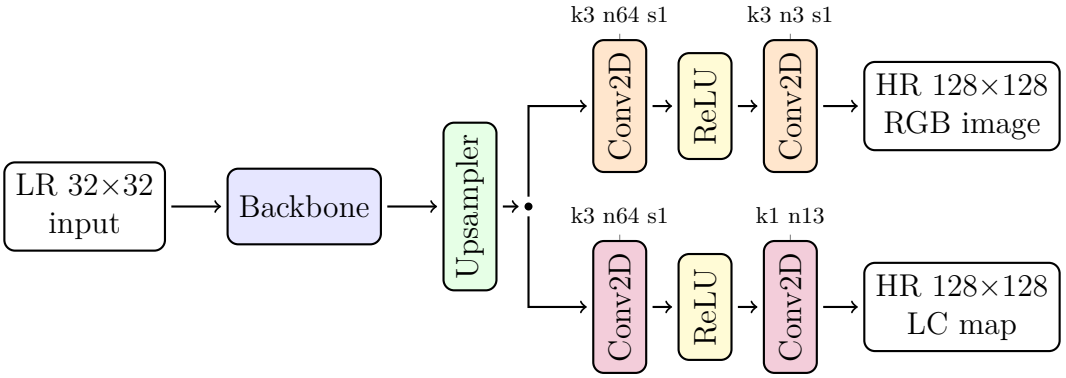

Figure 4.16. A representation of MTL architecture, with super resolution and segmentation head. The network is able to produce a super-resolved RGB image and a super-resolved LC map by taking LR images as input.

4.3.6 U-Net

The `UNet` class implements a semantic segmentation model based on the U-Net architecture [30] with a ResNet encoder backbone, utilizing the SMP (`segmentation_models_pytorch`) library. The model is designed to support two segmentation tasks: low-resolution satellite image segmentation (`LRSegmentation`) and high-resolution aerial image segmentation (`HRSegmentation`), adapting the number of input channels accordingly.

For these experiments, the encoder is a ResNet18 backbone truncated at depth 3, providing intermediate feature maps. The decoder is customized with channel dimensions (128, 64, 32) to progressively upsample and reconstruct the segmentation mask. The final segmentation output has a number of classes specified by the configuration.

The input passes through the encoder, producing a hierarchy of feature maps that are then decoded. Finally, the segmentation head produces the

class probability map. For LR segmentation, logits are cropped to 10×10, in order to match the target before the loss computation.

The model is flexible to easily incorporate metadata via additional MLP layers (however, these were not employed for the sake of the present work) and supports switching between different input channels and tasks by configuration, making it suitable for multi-source remote sensing segmentation tasks.

4.3.7 UTAE

The `UTAE` architecture (Figure 4.17) includes an encoder, which extracts multi-scale spatio-temporal features via a combination of convolutional layers and temporal self-attention modules. This design allows the network to adaptively attend across different times in the image sequence, enabling better modeling of temporal patterns.

The attention mechanism helps in weighing the contribution of different timestamps depending on how informative they are for semantic level in the data.

This model proved to be very effective when applied to satellite multi-temporal inputs, especially for applications related to agriculture, such as crop monitoring. This is due to the capability to carry critical information that cannot be fully exploited when processing single images individually.

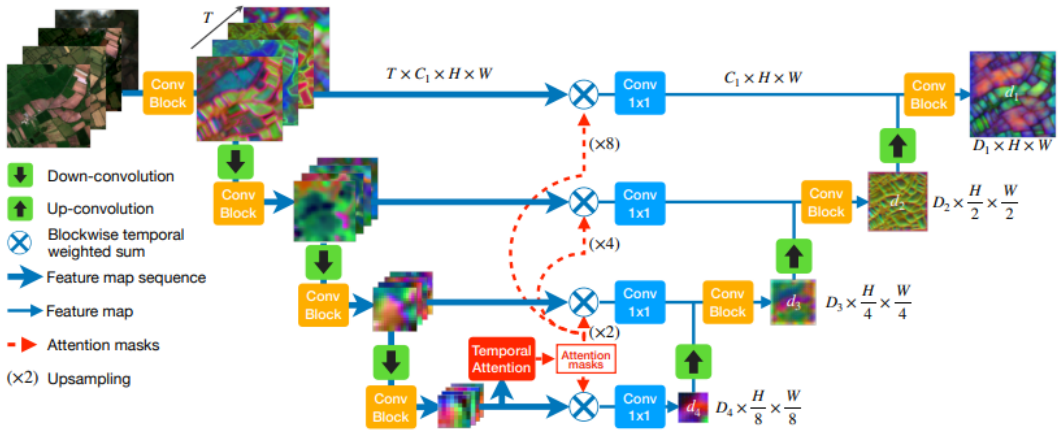

Figure 4.17. UTAE architecture

4.3.8 Time Texture Flair

The previously cited paper by Garioud et al. [36] presented an interesting architecture that employs both LR revisits (for the UTAE temporal branch) and VHR images (for the U-Net texture branch), using a combined loss. However, because of the input it takes, this model, depicted in Figure 4.18, is not suited for our purposes.

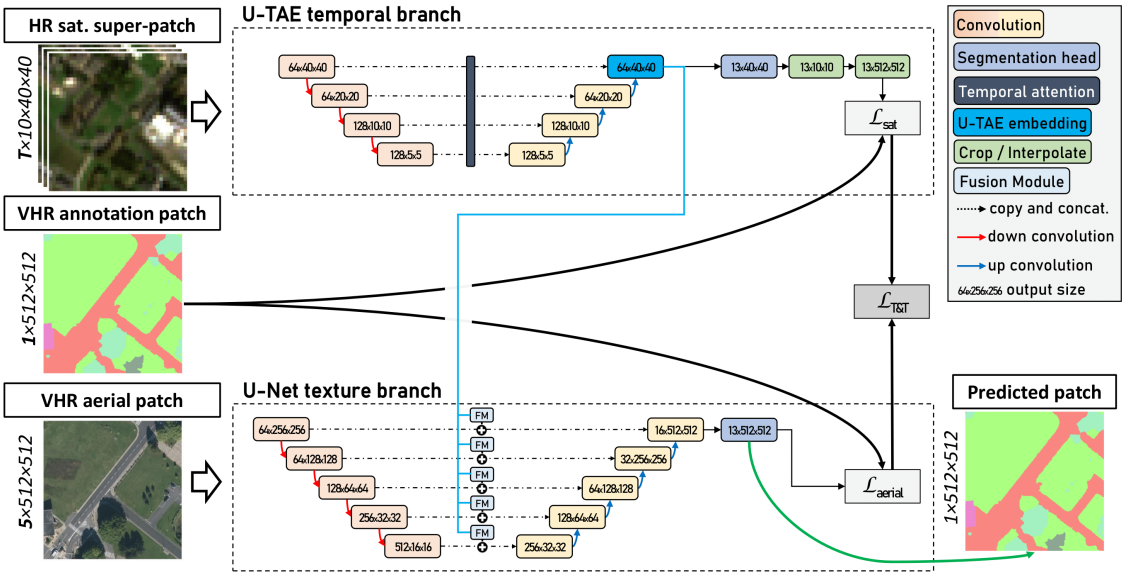

Figure 4.18. Time Texture Flair architecture, as presented in the paper.

4.4 Losses

In order to efficiently train and compare models, different losses were explored and evaluated, both individually and in weighted combinations. Loss selection was tailored to the specific task: super-resolution or segmentation.

4.4.1 Super-resolution

Pixel-wise loss: The most basic and widely used loss for image restoration is the pixel-wise loss, typically implemented as L1 (Mean Absolute Error, MAE) or L2 (Mean Squared Error, MSE). This loss directly penalizes the differences between the predicted and GT pixel values, encouraging accurate reconstruction at a low level.

$$\mathcal{L}_{L1} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{4.5}$$

$$\mathcal{L}_{L2} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{4.6}$$

SSIM loss: Structural Similarity Index (SSIM) loss focuses on the perceptual quality of the image by evaluating structural information, contrast, and luminance. Using SSIM as a loss encourages the network to preserve textures and edges, aligning with human visual perception.

$$\mathcal{L}_{SSIM}(x, y) = 1 - \text{SSIM}(x, y) \tag{4.7}$$

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{4.8}$$

Perceptual loss: Also known as feature loss, it compares high-level activations extracted from a pre-trained network (e.g., VGG19). By minimizing the difference in feature space rather than pixel space, it promotes semantic fidelity and perceptually convincing output.

$$\mathcal{L}_{perc} = \sum_{l}\frac{1}{C_l H_l W_l}\left\|\phi_l(y) - \phi_l(\hat{y})\right\|_2^2 \tag{4.9}$$

where $\phi_l(\cdot)$ are features extracted by layer l of a pre-trained network (e.g. VGG or AlexNet).

Adversarial loss: Used in GAN-based architectures, this loss introduces a discriminator network to distinguish real from generated images. The generator is trained to fool the discriminator, leading to more realistic and natural results. However, adversarial loss may lead to instability and requires careful balance with content loss.

For the generator:

$$\mathcal{L}_{adv}^{G} = -\mathbb{E}_{\hat{y}}\left[\log D(\hat{y})\right] \tag{4.10}$$

For the discriminator:

$$\mathcal{L}_{adv}^{D} = -\mathbb{E}_{y}\left[\log D(y)\right] - \mathbb{E}_{\hat{y}}\left[\log(1 - D(\hat{y}))\right] \tag{4.11}$$

CLIP loss: CLIP loss leverages the multi-modal feature space of OpenAI CLIP model [37] to guide generation using textual or image embeddings. This loss promotes alignment between image content and higher-level semantic descriptions, helping enforce meaningful structures in a zero-shot or loosely supervised setting.

$$\mathcal{L}_{CLIP} = 1 - \frac{f_{\text{img}}(\hat{y}) \cdot f_{\text{text}}(t)}{\|f_{\text{img}}(\hat{y})\| \, \|f_{\text{text}}(t)\|} \tag{4.12}$$

4.4.2 Segmentation

Dice loss: Dice loss is derived from the Dice Similarity Coefficient and is particularly effective in handling class imbalance. It measures the overlap between predicted and ground-truth masks, favoring the correct segmentation of small or underrepresented structures.

$$\mathcal{L}_{Dice} = 1 - \frac{2\sum_{i=1}^{N} p_i g_i + \epsilon}{\sum_{i=1}^{N} p_i + \sum_{i=1}^{N} g_i + \epsilon}, \tag{4.13}$$

where $p_i \in [0,1]$ is the predicted probability, $g_i \in \{0,1\}$ is the label and ϵ is used for numerical stabilization.

Cross-entropy loss: Widely used in classification tasks, cross-entropy loss penalizes the divergence between predicted probability distributions and one-hot encoded ground truths. For segmentation, it is applied pixel-wise and is effective when class distributions are relatively balanced. On the other hand, its weighted variant may be remarkably useful when the class distribution is imbalanced.

$$\mathcal{L}_{CE} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} y_{i,c} \log \hat{y}_{i,c} \tag{4.14}$$

where $y_{i,c}$ is one-hot ground-truth, $\hat{y}_{i,c}$ is predicted probability for class c.

The weighted version is:

$$\mathcal{L}_{WCE} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} w_c \, y_{i,c} \log \hat{y}_{i,c} \tag{4.15}$$

with w_c weight for each class.

Focal loss: Designed to address extreme class imbalance, Focal loss down-weights easy examples and focuses training on hard, misclassified pixels. It introduces a modulating factor to the cross-entropy formulation, allowing the model to learn from rare classes more effectively.

$$\mathcal{L}_{Focal} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C}\alpha_c(1-\hat{y}_{i,c})^{\gamma}\, y_{i,c}\,\log\hat{y}_{i,c} \tag{4.16}$$

where $\gamma > 0$ is the focusing parameter that reduces the relative loss for well-classified examples, and α_c is a weighting factor for class balance.

4.5 Metrics

In terms of metrics, possibilities were wide and related to different goals. For super-resolution, we aimed to evaluate pixel reconstruction accuracy, structural preservation, image quality, and photorealism. For segmentation, we mainly assessed Intersection over Union (IoU), yet we should cite F1-score, precision, and recall to measure mask quality and class-wise performance.

4.5.1 Super-resolution

PSNR: Peak Signal-to-Noise Ratio is a traditional metric used to assess the fidelity of image reconstruction. It quantifies the difference between the pixel values of the generated and ground-truth images on a logarithmic scale. Higher PSNR indicates better reconstruction quality.

$$\text{PSNR} = 10\cdot\log_{10}\left(\frac{MAX_I^2}{\text{MSE}}\right), \quad \text{where MSE} = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(I(i,j)-K(i,j))^2 \tag{4.17}$$

cPSNR: Color PSNR (cPSNR) is a variant of PSNR that compensates for global brightness and color shifts between the prediction and the ground truth, providing a more reliable assessment of super-resolution quality.

$$\text{cPSNR} = 10\log_{10}\left(\frac{255^2}{\text{cMSE}}\right), \tag{4.18}$$

$$\text{cMSE} = \min_b \frac{1}{N}\left\|I_{\text{SR}} - \left(I_{\text{HR}} + b\right)\right\|_2^2, \tag{4.19}$$

where b is a scalar bias optimized to minimize the mean squared error, I_{SR} is the super-resolved image, I_{HR} is the high-resolution ground truth, and N is the number of pixels.

SSIM: The Structural Similarity Index evaluates images based on structural information, contrast, and luminance. Unlike PSNR, SSIM aligns better with perceptual quality and is more robust to small geometric transformations.

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{4.20}$$

MS-SSIM: Multi-Scale SSIM extends SSIM by computing the index at multiple image resolutions, capturing both global and local structural information. MS-SSIM is more consistent with human perception, especially when fine and coarse details coexist.

$$\text{MS-SSIM}(x, y) = \prod_{j=1}^{M} \left[l_j(x, y)^{\alpha_j} \cdot c_j(x, y)^{\beta_j} \cdot s_j(x, y)^{\gamma_j} \right] \tag{4.21}$$

LPIPS: The Learned Perceptual Image Patch Similarity is a deep learning-based metric that compares features extracted from deep networks such as VGG. LPIPS correlates better with human judgment than traditional pixel-based metrics.

$$\text{LPIPS}(x, y) = \sum_{l} \frac{1}{H_l W_l} \sum_{h=1}^{H_l} \sum_{w=1}^{W_l} \| w_l \odot (\phi_l(x)_{hw} - \phi_l(y)_{hw}) \|_2^2 \tag{4.22}$$

NIQE: Natural Image Quality Evaluator is a no-reference metric that estimates the quality of an image without ground truth, using statistical regularities of natural images. Lower NIQE scores generally indicate higher perceptual quality.

$$\text{NIQE}(I) = \sqrt{(\mu_I - \mu_n)^\top (\Sigma_I + \Sigma_n)^{-1} (\mu_I - \mu_n)} \tag{4.23}$$

CLIP score: CLIP score [38] measures similarity in a joint text-image embedding space provided by CLIP. It can be used to evaluate how well the generated images align with semantic content or textual descriptions, providing a zero-shot perceptual quality signal.

$$\text{CLIP}(I, T) = \frac{f_{\text{img}}(I) \cdot f_{\text{text}}(T)}{\|f_{\text{img}}(I)\| \, \|f_{\text{text}}(T)\|} \tag{4.24}$$

4.5.2 Segmentation

IoU: Intersection over Union (IoU), also known as the Jaccard Index, measures the overlap between predicted and ground-truth masks. It is calculated per class and averaged (mIoU) to provide a robust segmentation quality measure.

$$\text{IoU} = \frac{|P \cap G|}{|P \cup G|} \tag{4.25}$$

Precision and recall: Precision measures how many of the predicted positive pixels are correct, while recall quantifies how many of the true positives are recovered. Analyzing both gives insight into the types of error the model makes (over- vs under-segmentation).

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4.26}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4.27}$$

F1-score: The F1-score combines precision and recall into a single metric. It is particularly informative when dealing with imbalanced datasets, as it balances false positives and false negatives.

$$F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.28}$$

Chapter 5

Experiments

All the following experiments were conducted on the FLAIR-2 dataset. When training from scratch on LR images, input combinations of 3, 4 and 10 channels and 1, 2, 4 and 8 revisits were evaluated. As stated previously, revisits were mostly treated as channels, whereas, for segmentation, using UTAE allowed for effective temporal aggregation.

5.1 Super-resolution

Two of the chosen architectures, SRCNN and ESRGAN, were built identically to those used for S2NAIP. Some pre-trained generators and discriminators for ESRGAN were employed as well.

After that, RCAN and SwinIR were implemented and explored in order to test modern architectures with remote sensing real-world images.

As an optimizer, AdamW was empirically selected, as it outperformed Adam and SGD: learning rate was set to 0.0001, weight decay to 0.001 and early stopping was used to prevent overfitting. In addition, combined loss was evaluated as a way to optimize certain aspects of image quality and possibly increase human judgement, as one of the goals of this task is to evaluate it together with quantitative metrics (which are reported in Tables 5.1–5.12). Some examples of super-resolved images are shown in Figure 5.2.

For pre-trained ESRGAN, metrics show better image quality (low LPIPS), but worse PSNR and SSIM. These models are optimal to reconstruct realistic details, but may be lacking in pixel-wise metrics, especially when not fine-tuned (Figure 5.1). Furthermore, they sometimes generate artifacts and

suffer instability: this is a well-known problem with GANs, and with this network, as well, but it does not make these models less valuable and does not have a remarkable impact for the goals of the present work.

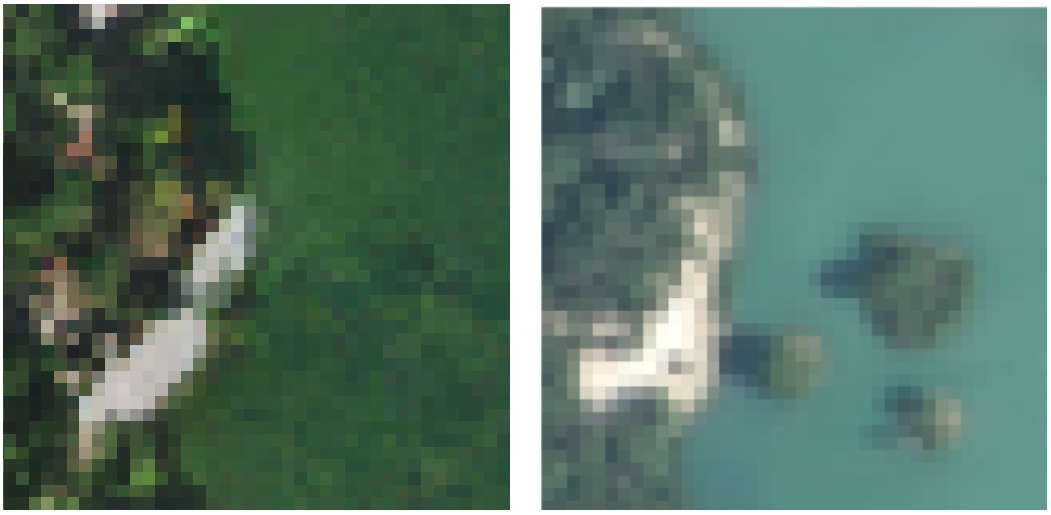

Figure 5.1. Example of the output produced by the pre-trained ESRGAN model, with the super-resolved image on the left and the HR target on the right. The model tends to overestimate the presence of green areas and vegetation, while excelling at reconstructing fine details and textures (e.g., trees).

Table 5.1. Results for super-resolution using $C_{\text{img}} = 3$ (RGB) with a single revisit ($N_{\text{rev}} = 1$). PSNR and cPSNR are reported in dB.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	cPSNR ↑
Bilinear	13.795	0.313	0.309	22.814
SRCNN	18.234	0.399	0.278	23.094
RCAN	18.319	0.393	0.269	22.891
SwinIR	**18.366**	**0.402**	0.256	**23.162**
ESRGAN (pre-trained)	15.974	0.330	**0.114**	21.298
ESRGAN (adversarial)	18.284	0.374	0.131	22.884

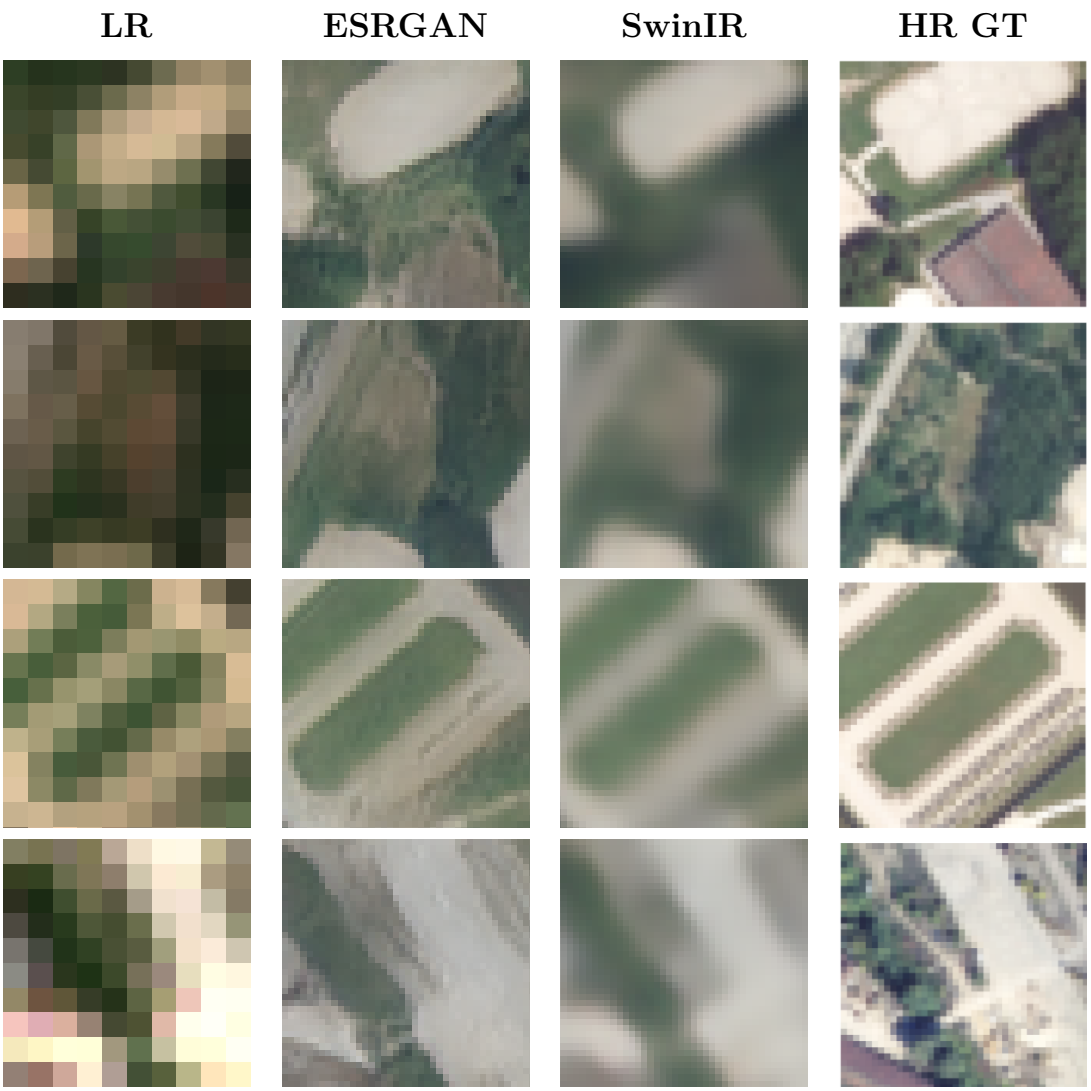

Figure 5.2. Some examples of super-resolution outputs. From left to right, LR image, output by ESRGAN and SwinIR and HR target (which is the result of downsampling from the original VHR target). The target is far richer in details, being able to preserve them after downsampling, whereas the starting LR image lost most of them. Therefore, the networks are not able to learn and reconstruct some finer objects, such as cars, which are way smaller than input resolution (10 m/pixel).

5.2 Land cover segmentation

For HR experiments and for UTAE, the FLAIR-2 provided implementations were used, while a slightly smaller U-Net was empirically chosen for LR

Table 5.2. Results for super-resolution using $C_{\text{img}} = 3$ (RGB) with $N_{\text{rev}} = 2$. PSNR and cPSNR are reported in dB.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	cPSNR ↑
SRCNN	18.660	0.406	0.273	23.164
RCAN	**18.845**	**0.413**	0.271	23.252
SwinIR	18.814	**0.413**	0.262	**23.268**
ESRGAN (pre-trained)	16.236	0.323	**0.098**	21.370
ESRGAN (adversarial)	18.556	0.381	0.118	22.972

Table 5.3. Results for super-resolution using $C_{\text{img}} = 3$ (RGB) with $N_{\text{rev}} = 4$. PSNR and cPSNR are reported in dB.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	cPSNR ↑
SRCNN	18.817	0.410	0.275	23.253
RCAN	**18.987**	0.415	0.268	23.320
SwinIR	18.935	**0.416**	0.260	**23.414**
ESRGAN (pre-trained)	15.722	0.342	**0.100**	21.685
ESRGAN (adversarial)	18.719	0.390	0.136	23.143

Table 5.4. Results for super-resolution using $C_{\text{img}} = 3$ (RGB) with $N_{\text{rev}} = 8$. PSNR and cPSNR are reported in dB.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	cPSNR ↑
SRCNN	18.939	0.415	0.271	23.402
RCAN	**19.060**	0.416	0.267	23.398
SwinIR	18.930	**0.419**	0.256	**23.461**
ESRGAN (pre-trained)	14.481	0.336	**0.105**	21.752
ESRGAN (adversarial)	18.884	0.394	0.125	23.291

Table 5.5. Results for super-resolution using $C_{\text{img}} = 4$ (RGB + NIR) with $N_{\text{rev}} = 1$. PSNR and cPSNR are reported in dB.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	cPSNR ↑
SRCNN	18.385	0.402	0.277	**23.229**
RCAN	**18.521**	**0.404**	0.272	23.218
SwinIR	18.238	0.402	0.263	23.191
ESRGAN	18.340	0.383	**0.145**	23.093

Table 5.6. Results for super-resolution using $C_{\text{img}} = 4$ (RGB + NIR) with $N_{\text{rev}} = 2$. PSNR and cPSNR are reported in dB.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	cPSNR ↑
SRCNN	18.637	0.406	0.275	23.171
RCAN	18.899	0.413	0.270	23.296
SwinIR	**18.960**	**0.415**	0.262	**23.313**
ESRGAN	18.664	0.387	**0.133**	23.105

Table 5.7. Results for super-resolution using $C_{\text{img}} = 4$ (RGB + NIR) with $N_{\text{rev}} = 4$. PSNR and cPSNR are reported in dB.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	cPSNR ↑
SRCNN	18.803	0.410	0.276	23.273
RCAN	18.981	0.414	0.268	23.321
SwinIR	**19.004**	**0.416**	0.262	**23.380**
ESRGAN	18.707	0.398	**0.156**	23.241

segmentation, mainly because of the smaller input size and to make it comparable with other SR models size.

All of these experiments were carried out with SGD optimizer and cross-entropy loss: weighted versions were evaluated, but they did not perform better than the standard one. The chosen learning rate was 0.001. The entire set of hyperparameters matches the ones used by the original paper.

Table 5.8. Results for super-resolution using $C_{\text{img}} = 4$ (RGB + NIR) with $N_{\text{rev}} = 8$. PSNR and cPSNR are reported in dB.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	cPSNR ↑
SRCNN	18.943	0.415	0.276	23.409
RCAN	**19.178**	**0.419**	0.265	**23.489**
SwinIR	18.932	0.417	0.268	23.402
ESRGAN	18.868	0.393	**0.130**	23.282

Table 5.9. Results for super-resolution using $C_{\text{img}} = 10$ (all available bands) with $N_{\text{rev}} = 1$. PSNR and cPSNR are reported in dB.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	cPSNR ↑
SRCNN	18.465	0.401	0.283	23.097
RCAN	**18.649**	0.404	0.272	**23.260**
SwinIR	18.500	**0.405**	0.260	23.227
ESRGAN	18.525	0.381	**0.136**	23.060

Table 5.10. Results for super-resolution using $C_{\text{img}} = 10$ (all available bands) with $N_{\text{rev}} = 2$. PSNR and cPSNR are reported in dB.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	cPSNR ↑
SRCNN	18.855	0.408	0.277	23.201
RCAN	**18.917**	**0.413**	0.271	**23.294**
SwinIR	18.902	**0.413**	0.261	23.286
ESRGAN	18.611	0.378	**0.137**	22.993

As a reference, we will also include results for bilinear interpolation applied to LR image (Table 5.15) and results for segmentation performed on super-resolved images (Table 5.16).

VHR and HR: The original result was obtained by employing the TXT-Flair model. However, purely comparing to VHR segmentation (having a

Table 5.11. Results for super-resolution using $C_{\text{img}} = 10$ (all available bands) with $N_{\text{rev}} = 4$. PSNR and cPSNR are reported in dB.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	cPSNR ↑
SRCNN	18.648	0.408	0.279	23.257
RCAN	18.950	0.412	0.269	23.262
SwinIR	**18.979**	**0.414**	0.263	**23.330**
ESRGAN	18.679	0.380	**0.129**	23.063

Table 5.12. Results for super-resolution using $C_{\text{img}} = 10$ (all available bands) with $N_{\text{rev}} = 8$. PSNR and cPSNR are reported in dB.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	cPSNR ↑
SRCNN	18.771	0.414	0.276	23.407
RCAN	**19.099**	0.416	0.263	**23.441**
SwinIR	19.051	**0.418**	0.256	**23.441**
ESRGAN	18.862	0.380	**0.119**	23.103

resolution equal to 0.2 m/pixel) any results obtained from LR images would be misleading and useless, so those results should be read as a reference and not as a target for the present study. Indeed, using a simple U-Net on HR single images is more fair, therefore this will be the upper bound. These results are reported in Table 5.13.

Table 5.13. Results for HR (2.5 m/pixel) segmentation mIoU. Experiment for VHR (0.2 m/pixel), which is not included in the table, produced mIoU = 0.5506.

	RGB	RGB + NIR	RGB + NIR + elevation
U-Net	0.3568	0.4061	0.4639

LR: As the lower bound of the baseline, the two models were compared, in order to account for both single-image and multi-image, leveraging revisits.

Both were fed LR images and produced LR segmentation maps in different settings in terms of input channels (3, 4, 10) and revisits (1, 2, 4, 8), as shown in Table 5.14.

A better reference to establish a comparison with the final multi-task version consists of "squeezing" decoded features before the segmentation head, in order to match them with a 40×40 map. Therefore, in this case, the net outputs "super-resolved" logits thanks to bilinear interpolation: this is the result that this work aims to improve, in terms of both quality and metrics, and it is reported in Table 5.15.

Table 5.14. Comparing mIoU for LR U-Net e UTAE by varying C_{img} and N_{rev}.

	UNet (N_{rev})				**UTAE** (N_{rev})			
C_{img}	1	2	4	8	1	2	4	8
3	0.2467	0.2636	0.2938	0.3006	0.2487	0.2729	0.3444	0.3704
4	0.2727	0.2876	0.2984	0.3157	0.2496	0.3206	0.3595	0.3807
10	0.3146	0.3306	0.3307	0.3424	0.3546	0.3641	0.3789	**0.3994**

Moreover, applying bilinear interpolation to features before the segmentation head, i.e. generating HR output from LR input, often scores worse results. However, the benefit that UTAE derives from temporal attention and aggregation is remarkable, and it can be easily noticed for higher number of revisits.

Table 5.15. Comparing mIoU for U-Net e UTAE by varying C_{img} and N_{rev} with bilinear interpolation applied to features.

	UNet (N_{rev})				**UTAE** (N_{rev})			
C_{img}	1	2	4	8	1	2	4	8
3	0.2428	0.266	0.2638	0.2982	0.2501	0.2763	0.3184	0.345
4	0.26	0.294	0.3011	0.3199	0.2804	0.3046	0.3355	0.3646
10	0.3054	0.3212	0.3152	0.3489	0.3219	0.3564	0.3697	**0.4024**

Of course, HR segmentation leads to much higher mIoU than LR segmentation, and even additional channels and revisits are not easily bridging the gap.

SR: With the goal of investigating the contribution of SR to downstream tasks, super-resolved images were used as input for LC segmentation. In particular, this step required new full datasets, consisting of the output of the best performing SwinIR (in terms of metrics) and both pre-trained and best performing ESRGAN (in terms of visual quality). U-Net was then trained and tested on the resulting dataset (Table 5.16).

Table 5.16. Comparing mIoU for U-Net trained on datasets made of super-resolved images (output of selected networks), with different numbers of channels and revisits. Results prove to be vastly inferior to those achieved by multi-task learning

Network	Channels	Revisits	mIoU ↑
SwinIR	3	1	0.2023
SwinIR	10	8	**0.2778**
ESRGAN (pre-trained)	3	1	0.2035
ESRGAN (adversarial)	10	8	0.2669

5.3 Multi-task learning

After obtaining a baseline, the next step consisted in applying SR models to output a super-resolved LC map. This goal was achieved in different ways.

5.3.1 Training from scratch

First of all, SRCNN, RCAN and SwinIR were trained from scratch with the addition of the segmentation head. Optimizing both tasks at the same time was not straightforward, compared to having two separate, specialized models, therefore different configurations of loss weights were evaluated.

5.3.2 Pre-trained models

Because of the good visual performance provided by the ESRGAN architecture, their generators were used as a starting point for MTL. All layers were frozen, except for the first convolution (to adapt to the colours and brightness differences) and the segmentation head: as expected, since the

net parameters adapt to segmentation, photorealism is spoiled and therefore the advantage of having a GAN is canceled.

After this first experiment, the entire generator was frozen, and only the segmentation head was trained. Unsurprisingly, extracted features were not optimal as well, so further solutions had to be found in order to provide the segmentation head with more information without adding too much weight and complexity to the architecture.

Specifically, a promising strategy was extracting multi-scale features from different layers of the frozen generator and concatenating them before feeding the segmentation head. Two methods were used: at first, a 3×3 convolution followed by a ReLU was inserted to extract information from the first layer, from the body and from the first upsampler. These features were interpolated to match the size of those generated by the second upsampler, and then concatenated before feeding the segmentation head, as shown in Figure 5.3 and Figure 5.4.

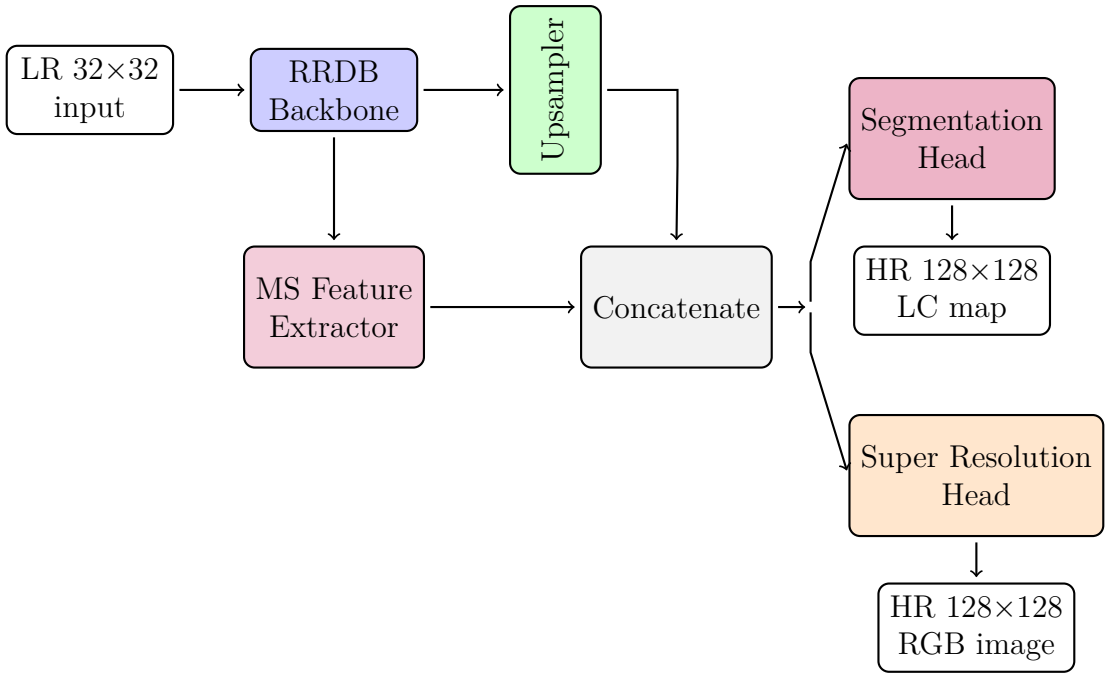

Figure 5.3. An image of customized RRDB generator with the addition of a multiscale feature extractor followed by concatenation.

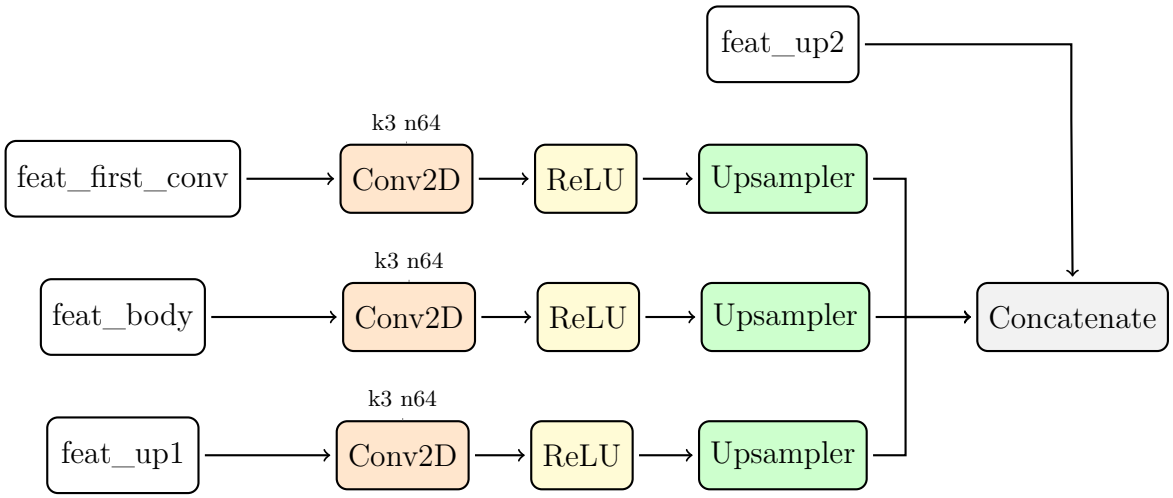

Figure 5.4. The internal structure of multiscale feature extractor, followed by concatenate block.

For the second method, an interesting idea came from Low-Rank Adaptation (LoRA) [39], which is state-of-the-art for Parameter-Efficient Fine-Tuning PEFT in Large Language Models (LLM). LoRA introduces a simple yet powerful mechanism to adapt a pre-trained model by injecting trainable low-rank matrices into existing weight layers, while keeping the original weights frozen (Figure 5.5): the main intuition is to constrain the update of large weight matrices to a low-dimensional subspace, allowing the model to learn task-specific knowledge without modifying the full set of parameters.

In terms of size, none of these methods represented more than 2% of the original RRDB parameters, making them very lightweight solutions.

5.3.3 Adversarial multi-task learning

In the end, ESRGAN was trained from scratch using the discriminator, having to balance the role of the adversarial loss and the contribution of the cross entropy loss. This approach proved successful, allowing to generate good quality images while extracting useful features for segmentation.

5.3.4 Results for multi-task learning

Similarly to super-resolution, also in multi-task learning the impact of considering all the channels is noticeable, but not as remarkable as in segmentation:

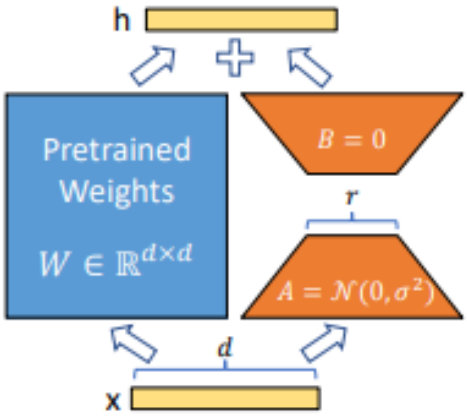

Figure 5.5. LoRA reparametrization: only A and B are trained, while pre-trained ESRGAN weights are not.

this can be explained by the stronger relationship that exists between some Sentinel bands and some types of LC classes.

For instance, in semantic segmentation tasks, the NIR band is crucial in identifying vegetation, as plants exhibit high reflectance in this spectral region, distinguishing them from soil or buildings. Analogously, SWIR bands are effective in detecting water, since it strongly absorbs in this range. In contrast, in SR tasks the goal is to reconstruct fine spatial details rather than to separate semantic categories, so while additional channels may provide useful information that slightly improve reconstruction quality, their contribution is limited.

Furthermore, the absence of a specifically designed temporal attention and temporal aggregation module makes the contribution of revisits inferior compared to UTAE, which leverages them greatly.

Metrics for multi-task learning are reported in Tables 5.17–5.28, which also include results for the ESRGAN model trained with different strategies (some evaluated only on RGB inputs, i.e. Tables 5.17, 5.18, 5.19, and 5.20). The tested approaches comprise a frozen generator with an added segmentation head, multiscale feature extraction through convolutional layers, LoRA, and finally full adversarial training with a discriminator.

In general, MTL is capable of outperforming standard LR in almost all the considered cases, provided that the SR model is large and efficient enough. Among the compared ones, SRCNN sometimes provides smaller or no advantage over UTAE, but largely wins against U-Net. Moreover, in some of

Table 5.17. Results for multi-task learning (compared to segmentation) with $C_{\text{img}} = 3$ (RGB) using $N_{\text{rev}} = 1$. PSNR and cPSNR are reported in dB.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	cPSNR ↑	mIoU ↑
U-Net	—	—	—	—	0.2428
UTAE	—	—	—	—	0.2501
SRCNN	18.273	0.375	0.246	22.685	0.2372
RCAN	18.360	0.397	0.265	22.953	0.2666
SwinIR	**18.561**	**0.401**	0.262	**23.039**	0.2674
ESRGAN (pre-trained)	15.974	0.330	**0.114**	21.298	0.1830
ESRGAN (multiscale feat.)	15.974	0.330	**0.114**	21.298	0.2572
ESRGAN (LoRA)	18.142	0.332	0.139	22.345	0.2631
ESRGAN (adversarial)	18.436	0.361	0.124	22.800	**0.3068**

Table 5.18. Results for multi-task learning (compared to segmentation) with $C_{\text{img}} = 3$ (RGB) using $N_{\text{rev}} = 2$. PSNR and cPSNR are reported in dB.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	cPSNR ↑	mIoU ↑
U-Net	—	—	—	—	0.2660
UTAE	—	—	—	—	0.2766
SRCNN	18.650	0.378	0.247	22.697	0.2894
RCAN	**18.886**	**0.405**	0.268	22.978	**0.3043**
SwinIR	18.767	**0.405**	0.258	**23.041**	0.3041
ESRGAN (pre-trained)	16.236	0.323	**0.098**	21.370	0.1957
ESRGAN (multiscale feat.)	16.236	0.323	**0.098**	21.370	0.2659
ESRGAN (LoRA)	18.356	0.344	0.159	22.431	0.225
ESRGAN (adversarial)	18.610	0.366	0.127	22.756	0.2763

the settings, not only MTL improves segmentation, but it also helps regularize super-resolution, achieving moderately superior results to the single task scenario. The best performing model overall is SwinIR, that, considering

Table 5.19. Results for multi-task learning (compared to segmentation) with $C_{\text{img}} = 3$ (RGB) using $N_{\text{rev}} = 4$. PSNR and cPSNR are reported in dB.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	cPSNR ↑	mIoU ↑
U-Net	—	—	—	—	0.2638
UTAE	—	—	—	—	0.3184
SRCNN	18.794	0.384	0.257	22.791	0.3054
RCAN	**18.947**	**0.408**	0.270	23.077	**0.3439**
SwinIR	18.888	0.407	0.260	**23.093**	0.3261
ESRGAN (pre-trained)	15.722	0.342	**0.100**	21.685	0.2205
ESRGAN (multiscale feat.)	15.722	0.342	**0.100**	21.685	0.2762
ESRGAN (LoRA)	18.505	0.366	0.156	22.712	0.2501
ESRGAN (adversarial)	18.655	0.367	0.127	22.828	0.3401

Table 5.20. Results for multi-task learning (compared to segmentation) with $C_{\text{img}} = 3$ (RGB) using $N_{\text{rev}} = 8$. PSNR and cPSNR are reported in dB.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	cPSNR ↑	mIoU ↑
U-Net	—	—	—	—	0.2982
UTAE	—	—	—	—	0.3450
SRCNN	19.084	0.390	0.250	22.967	0.3290
RCAN	**19.173**	**0.413**	0.266	23.217	**0.3693**
SwinIR	19.025	0.412	0.263	**23.235**	0.3574
ESRGAN (pre-trained)	14.481	0.336	**0.105**	21.752	0.2296
ESRGAN (multiscale feat.)	14.481	0.336	**0.105**	21.752	0.2725
ESRGAN (LoRA)	18.617	0.343	0.150	22.420	0.2975
ESRGAN (adversarial)	18.747	0.373	0.131	22.896	0.3614

mIoU, surpasses UTAE by up to 3.6 points and U-Net by up to 8.2 points, but RCAN still scores better in some settings.

When frozen and equipped with the segmentation head only, ESRGAN

Table 5.21. Results for multi-task learning (compared to segmentation) with $C_{\text{img}} = 4$ (RGB + NIR) using $N_{\text{rev}} = 1$. PSNR and cPSNR are reported in dB.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	cPSNR ↑	mIoU ↑
U-Net	—	—	—	—	0.2601
UTAE	—	—	—	—	0.2804
SRCNN	18.523	0.380	0.248	22.751	0.2799
RCAN	18.456	0.396	0.266	22.878	0.2999
SwinIR	**18.609**	**0.401**	0.264	**23.081**	**0.3009**
ESRGAN (adversarial)	18.182	0.360	**0.132**	22.763	0.2983

Table 5.22. Results for multi-task learning (compared to segmentation) with $C_{\text{img}} = 4$ (RGB + NIR) using $N_{\text{rev}} = 2$. PSNR and cPSNR are reported in dB.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	cPSNR ↑	mIoU ↑
U-Net	—	—	—	—	0.2940
UTAE	—	—	—	—	0.3046
SRCNN	18.687	0.378	0.252	22.693	0.3254
RCAN	**18.829**	0.404	0.268	22.955	0.3361
SwinIR	18.742	**0.406**	0.263	**23.016**	0.3305
ESRGAN (adversarial)	18.712	0.371	**0.144**	22.780	**0.3405**

performs poorly in terms of mIoU. On the other hand, some fine-tuning is needed for the net to adapt to the new task, but it is not always enough to be competitive with the baseline: LoRA has a slightly negative impact on visual quality, but it improves the other metrics, whereas the method based on multi-scale feature extraction preserves photorealism, since it leaves the generator untouched. As mentioned before, the most promising approach for GANs consists in multi-task adversarial training, thanks to the discriminator: the downsides of this latter technique are indeed its instability and the need

Table 5.23. Results for multi-task learning (compared to segmentation) with $C_{\text{img}} = 4$ (RGB + NIR) using $N_{\text{rev}} = 4$. PSNR and cPSNR are reported in dB.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	cPSNR ↑	mIoU ↑
U-Net	—	—	—	—	0.3011
UTAE	—	—	—	—	0.3355
SRCNN	18.822	0.382	0.251	22.810	0.3494
RCAN	**18.967**	0.408	0.270	23.092	**0.3459**
SwinIR	18.944	**0.409**	0.265	**23.131**	0.3443
ESRGAN (adversarial)	18.709	0.371	**0.127**	22.879	0.3419

Table 5.24. Results for multi-task learning (compared to segmentation) with $C_{\text{img}} = 4$ (RGB + NIR) using $N_{\text{rev}} = 8$. PSNR and cPSNR are reported in dB.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	cPSNR ↑	mIoU ↑
U-Net	—	—	—	—	0.3199
UTAE	—	—	—	—	0.3646
SRCNN	19.040	0.391	0.252	22.977	0.3309
RCAN	18.995	0.408	0.277	23.092	0.3310
SwinIR	**19.045**	**0.413**	0.261	**23.206**	0.3752
ESRGAN (adversarial)	18.797	0.378	**0.134**	22.985	**0.3943**

for longer training time, but it has potential to produce better visual quality and segmentation output.

In the end, the benefit of using SR models is not only remarkable when looking at quantitative metrics, but also evident when observing qualitative results (Figure 5.6), which show better defined shapes and enhanced reconstruction for smaller details.

Table 5.25. Results for multi-task learning (compared to segmentation) with $C_{\text{img}} = 10$ (all available bands) using $N_{\text{rev}} = 1$. PSNR and cPSNR are reported in dB.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	cPSNR ↑	mIoU ↑
U-Net	—	—	—	—	0.3054
UTAE	—	—	—	—	0.3219
SRCNN	18.494	0.377	0.252	22.730	0.3362
RCAN	18.426	0.394	0.266	22.887	0.3369
SwinIR	**18.794**	**0.400**	0.260	**23.018**	0.3580
ESRGAN (adversarial)	18.257	0.359	**0.122**	22.799	**0.3586**

Table 5.26. Results for multi-task learning (compared to segmentation) with $C_{\text{img}} = 10$ (all available bands) using $N_{\text{rev}} = 2$. PSNR and cPSNR are reported in dB.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	cPSNR ↑	mIoU ↑
U-Net	—	—	—	—	0.3212
UTAE	—	—	—	—	0.3564
SRCNN	**18.809**	0.382	0.250	22.788	0.3572
RCAN	18.799	0.402	0.271	22.928	0.3660
SwinIR	18.770	**0.406**	0.264	**23.124**	**0.3855**
ESRGAN (adversarial)	18.743	0.368	**0.124**	22.819	0.3712

Table 5.27. Results for multi-task learning (compared to segmentation) with $C_{\text{img}} = 10$ (all available bands) using $N_{\text{rev}} = 4$. PSNR and cPSNR are reported in dB.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	cPSNR ↑	mIoU ↑
U-Net	—	—	—	—	0.3152
UTAE	—	—	—	—	0.3697
SRCNN	18.807	0.384	0.257	22.836	0.3646
RCAN	**19.005**	0.407	0.268	23.112	**0.3990**
SwinIR	18.948	**0.410**	0.263	**23.160**	0.3979
ESRGAN (adversarial)	18.702	0.379	**0.146**	22.926	0.3919

Table 5.28. Results for multi-task learning (compared to segmentation) with $C_{\text{img}} = 10$ (all available bands) using $N_{\text{rev}} = 8$. PSNR and cPSNR are reported in dB.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	cPSNR ↑	mIoU ↑
U-Net	—	—	—	—	0.3489
UTAE	—	—	—	—	0.4024
SRCNN	18.675	0.398	0.266	23.068	0.3908
RCAN	18.917	0.410	0.273	**23.213**	0.4032
SwinIR	**19.064**	**0.411**	0.261	**23.184**	**0.4154**
ESRGAN (adversarial)	18.833	0.381	**0.133**	22.974	0.3949

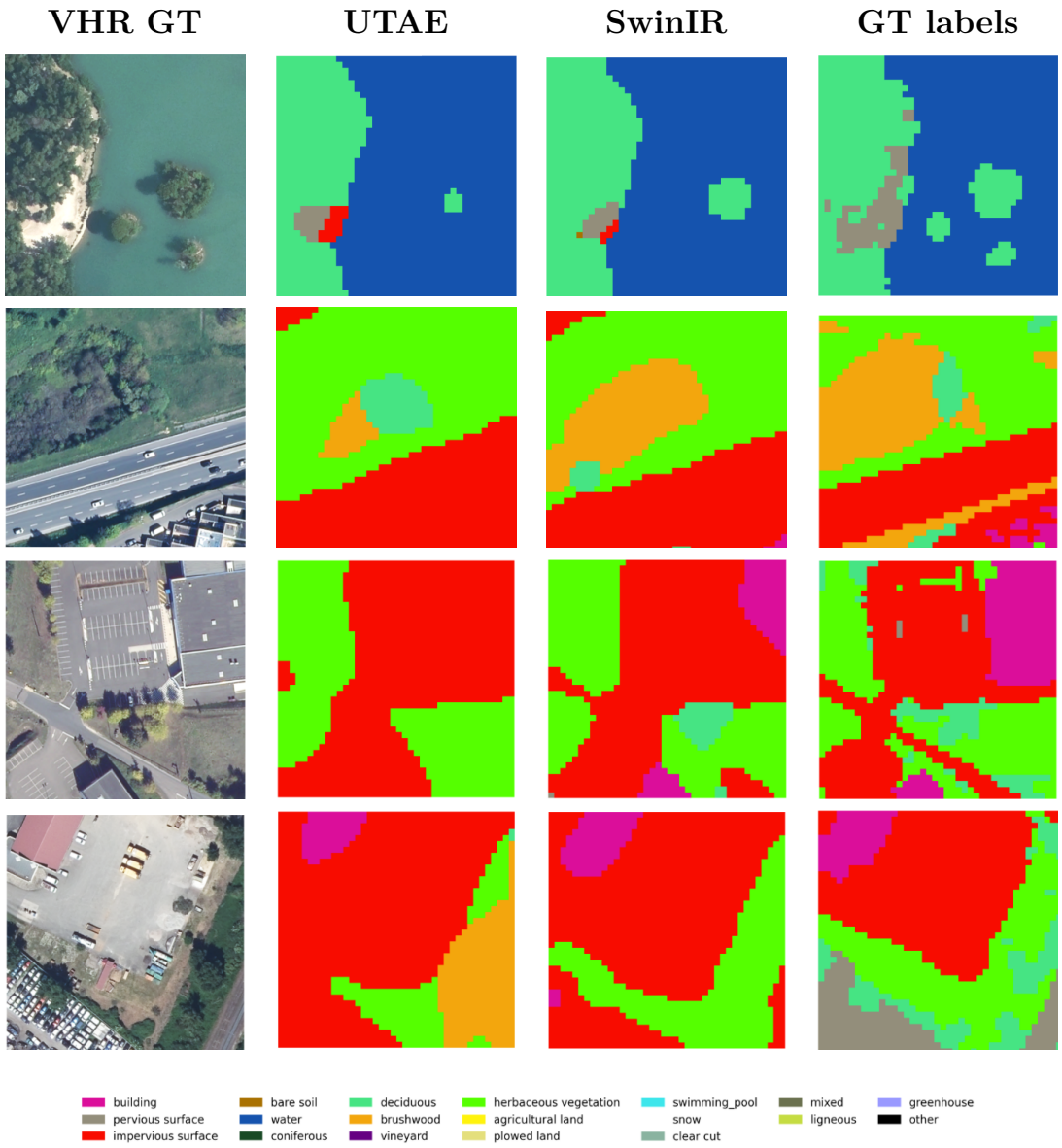

Figure 5.6. Some examples of land cover maps. From the left to the right, VHR ground truth (0.2 m/pixel), results obtained by UTAE, results obtained by SwinIR (both using 10 channels and 8 revisits) and GT labels (2.5 m/pixel). Beyond quantitative metrics, the SR model is also remarkably better at reconstructing shapes and details.

Chapter 6

Conclusion

In summary, the present work compares different categories of SR architectures and applies them to remote sensing, analyzing trade-offs, and enhancing their potential to improve land cover segmentation.

The main idea that the thesis aims to investigate and validate is whether and to what extent super-resolved features could be exploited by segmentation task, without the need for two different networks or decoders.

Overall, the best performing model for the SR task was SwinIR, which is based on Vision Transformer architecture, but ESRGAN obtained remarkable results in terms of visual quality and photorealism, yet scored the lowest in pixel-wise metrics and produced more artifacts. This demonstrates the different goals of standard and adversarial networks in image generation and restoration and the lack of correspondence between qualitative and quantitative results.

Moreover, SR networks are effective in improving mIoU when equipped with a segmentation head: the latter task benefits the most from architectures that optimize traditional metrics, such as PSNR and SSIM, whereas it needs some adaptation when extracting information from GANs.

6.1 Contributions

The key contributions of this thesis include:

1. **A baseline for LR (10 m/pixel) and HR (2.5 m/pixel) land cover segmentation on FLAIR-2 dataset**, in order to obtain lower and upper bounds for the experiments.

2. **A thorough comparison of different types of SR networks**, analyzing strengths, trade-offs, losses and metrics when fed with multi-spectral and multi-temporal data.

3. **Multi-task learning for SRCNN, RCAN and SwinIR**, obtained by **training the full SR network from scratch** with the addition of a segmentation head: this showed to be effective in improving segmentation results with respect to the LR baseline.

4. **Multi-task learning for pre-trained GANs**, performed by adding a segmentation head to a frozen RRDB generator, and then including lightweight **multiscale features extractors**. This allowed the net to preserve photorealism while benefiting segmentation, despite having a tiny number of trainable parameters, thanks to innovative techniques such as **LoRA**.

5. **MTL with adversarial training**, performed by training ESRGAN from scratch with both discriminator and segmentation head, overcoming challenges related to instability.

6.2 Future works

Although numerous experiments and scenarios were considered, there is much more left to be explored in how super-resolution can benefit other tasks. In particular, interesting research topics may include:

- **Using a bigger dataset specifically aimed at multi-task learning**, since FLAIR-2 was originally built for VHR land cover segmentation. Furthermore, better-matched LR-HR pairs, avoiding downsampling, interpolation, and cropping, could simplify the task and lead to better results.

- **Comparing modern architectures that are specifically designed for remote sensing**, although results scored by general-purpose models look promising. This might include diffusion models (e.g. EDiffSR), which were not considered in this work and pose their own challenges, in terms of both results and computational cost.

- **Multi-modal architectures**: leveraging different sources such as SAR, InSAR or metadata is an interesting option to add useful information.

- **Experimenting additional tasks** (e.g., object detection) may further regularize training, improve convergence and enhance results, helping generate richer and versatile upscaled features.
- **Additional losses and metrics**, such as CLIP loss, which have been shown to be effective in speeding up training and achieving better results.

In conclusion, super-resolution demonstrates significant potential to enhance related tasks such as segmentation and object detection, serving as a valuable performance booster. Although it currently occupies a more niche position compared to well-established fields like classification or segmentation, the findings of this work highlight that there remains considerable scope for further exploration and application. This opens promising perspectives for both academic research and industrial practice, where the integration of super-resolution could provide tangible added value.

Acknowledgements

I would like to sincerely thank my supervisor, Professor Paolo Garza, for his guidance and constant feedback throughout the various experimental phases, the writing process, and the administrative procedures. I am also grateful to Edoardo, my co-supervisor, Marco and Luca, for their continuous support, patience, help, and advice from the very first to the last day of this experience, which were essential for the successful completion of this project. Furthermore, I would like to thank LINKS Foundation for providing the necessary tools and resources, without which this work would not have been possible.

My deepest gratitude goes to those who have supported, encouraged, and helped me on a personal and emotional level throughout this challenging journey, which reaches its conclusion with this work. They never stopped believing in me and always pushed me to do the same. First and foremost, to my parents and grandparents, to my uncles, aunts and cousins, to my girlfriend and her wonderful family, who have always stood by me and made me feel at home. Words cannot express how grateful and fortunate I feel to have you all in my life.

Finally, I would like to thank my friends, who have given me not only moments of lightheartedness but also their help in times of need.

Bibliography

[1] "Sentiwiki - copernicus programme." https://sentiwiki.copernicus.eu/web/copernicus-programme.

[2] Copernicus Programme, "Sentinel-1 mission." https://sentinels.copernicus.eu/copernicus/sentinel-1.

[3] Copernicus Programme, "Sentinel-2 mission." https://sentinels.copernicus.eu/copernicus/sentinel-2.

[4] W. Yang, X. Zhang, Y. Tian, W. Wang, J.-H. Xue, and Q. Liao, "Deep learning for single image super-resolution: A brief review," *IEEE Transactions on Multimedia*, vol. 21, no. 12, pp. 3106–3121, 2019.

[5] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[6] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *European conference on computer vision*, pp. 184–199, Springer, 2014.

[7] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *European conference on computer vision*, pp. 391–407, Springer, 2016.

[8] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1874–1883, 2016.

[9] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 136–144, 2017.

[10] N. Ahn, B. Kang, and K.-A. Sohn, "Fast, accurate, and lightweight super-resolution with cascading residual network," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 252–268, 2018.

[11] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 286–301, 2018.

[12] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[13] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4681–4690, 2017.

[14] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy, "Esrgan: Enhanced super-resolution generative adversarial networks," in *Proceedings of the European conference on computer vision (ECCV) workshops*, pp. 0–0, 2018.

[15] X. Wang, L. Xie, C. Dong, and Y. Shan, "Real-esrgan: Training real-world blind super-resolution with pure synthetic data," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1905–1914, 2021.

[16] Z. Wei, Y. Huang, Y. Chen, C. Zheng, and J. Gao, "A-esrgan: Training real-world blind super-resolution with attention u-net discriminators," in *Pacific Rim International Conference on Artificial Intelligence*, pp. 16–27, Springer, 2023.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[18] A. Dosovitskiy, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[19] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, "Swinir: Image restoration using swin transformer," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1833–1844, 2021.

[20] D. Zhang, F. Huang, S. Liu, X. Wang, and Z. Jin, "Swinfir: Revisiting the swinir with fast fourier convolution and improved training for image super-resolution," *arXiv preprint arXiv:2208.11247*, 2022.

[21] J. Yoo, T. Kim, S. Lee, S. H. Kim, H. Lee, and T. H. Kim, "Enriched cnn-transformer feature aggregation networks for super-resolution," in

Proceedings of the IEEE/CVF winter conference on applications of computer vision, pp. 4956–4965, 2023.

[22] Z. Chen, Y. Zhang, J. Gu, L. Kong, and X. Yang, "Recursive generalization transformer for image super-resolution," *arXiv preprint arXiv:2303.06373*, 2023.

[23] X. Chen, X. Wang, J. Zhou, Y. Qiao, and C. Dong, "Activating more pixels in image super-resolution transformer," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22367–22377, 2023.

[24] C.-C. Hsu, C.-M. Lee, and Y.-S. Chou, "Drct: Saving image super-resolution away from information bottleneck," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6133–6142, 2024.

[25] Y. Zhou, Z. Li, C.-L. Guo, S. Bai, M.-M. Cheng, and Q. Hou, "Srformer: Permuted self-attention for single image super-resolution," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 12780–12791, 2023.

[26] C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet, and M. Norouzi, "Image super-resolution via iterative refinement," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 4, pp. 4713–4726, 2022.

[27] C. Lanaras, J. Bioucas-Dias, S. Galliani, E. Baltsavias, and K. Schindler, "Super-resolution of sentinel-2 images: Learning a globally applicable deep neural network," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 146, pp. 305–319, 2018.

[28] Y. Gong, P. Liao, X. Zhang, L. Zhang, G. Chen, K. Zhu, X. Tan, and Z. Lv, "Enlighten-gan for super resolution reconstruction in mid-resolution remote sensing images," *Remote Sensing*, vol. 13, no. 6, p. 1104, 2021.

[29] T. Tarasiewicz, J. Nalepa, R. A. Farrugia, G. Valentino, M. Chen, J. A. Briffa, and M. Kawulok, "Multitemporal and multispectral data fusion for super-resolution of sentinel-2 images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–19, 2023.

[30] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[31] H. Cao, Y. Wang, J. Chen, D. Jiang, X. Zhang, Q. Tian, and M. Wang,

"Swin-unet: Unet-like pure transformer for medical image segmentation," in *European conference on computer vision*, pp. 205–218, Springer, 2022.

[32] V. S. F. Garnot and L. Landrieu, "Panoptic segmentation of satellite image time series with convolutional temporal attention networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4872–4881, 2021.

[33] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv preprint arXiv:1706.05098*, 2017.

[34] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7482–7491, 2018.

[35] P. Wolters, F. Bastani, and A. Kembhavi, "Zooming out on zooming in: Advancing super-resolution for remote sensing," *arXiv preprint arXiv:2311.18082*, 2023.

[36] A. Garioud, N. Gonthier, L. Landrieu, A. De Wit, M. Valette, M. Poupée, S. Giordano, *et al.*, "Flair: a country-scale land cover semantic segmentation dataset from multi-source optical imagery," *Advances in Neural Information Processing Systems*, vol. 36, pp. 16456–16482, 2023.

[37] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, pp. 8748–8763, PmLR, 2021.

[38] J. Hessel, A. Holtzman, M. Forbes, R. Le Bras, and Y. Choi, "Clipscore: A reference-free evaluation metric for image captioning," in *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 7514–7528, 2021.

[39] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, *et al.*, "Lora: Low-rank adaptation of large language models.," *ICLR*, vol. 1, no. 2, p. 3, 2022.