Sentinel-2 Satellite-Image Super-Resolution using SwinIR and MambaIR

by

Adam Kuca (220006635)

BSc (Hons) Computer Science

April 4, 2025



University of
St Andrews

Declaration

I hereby certify that this dissertation, which is approximately 10122 words in length, has been composed by me, that it is the record of work carried out by me and that it has not been submitted in any previous application for a degree. This project was conducted by me at the University of St Andrews from 09/2024 to 04/2025 towards fulfilment of the requirements of the University of St Andrews for the degree of BSc (Hons) Computer Science under the supervision of Dr Oggie Arandelovic.

In submitting this project report to the University of St Andrews, I give permission for it to be made available for use in accordance with the regulations of the University Library. I also give permission for the title and abstract to be published and for copies of the report to be made and supplied at cost to any bona fide library or research worker, and to be made available on the World Wide Web. I retain the copyright in this work.

Abstract

High-resolution satellite imagery plays a crucial role in many scientific and non-commercial applications, such as environmental monitoring or agriculture. However, access to such data is limited due to high costs and proprietary restrictions. Satellites such as Sentinel or Landsat have enabled access to free satellite images, but these are often limited in spatial resolution, making them insufficient for many tasks. This work investigates whether recent advances in deep learning, particularly transformer-based and state-space-based architectures, can be used to enhance freely available satellite imagery through single-image super-resolution (SISR).

Two state-of-the-art models, SwinIR and MambaIR, were implemented and trained on the S2-NAIP dataset, which pairs low-resolution Sentinel-2 images (32×32) with high-resolution NAIP images (128×128) for $4\times$ super-resolution. Both models were evaluated against standard baselines, including the CNN-based SRResNet and the GAN-based ESRGAN, using quantitative metrics such as PSNR, SSIM, and CLIPScore, as well as visual comparison. Results show that both SwinIR and MambaIR outperform the CNN-based model, with MambaIR achieving the best overall performance, marginally better than SwinIR.

An ablation study was conducted to examine the impact of CLIPScore loss and model weight initialisation. Findings suggest that perceptual quality can be improved at the cost of a slight decrease in pixel-wise accuracy, while $2\times$ pre-training does not lead to any improvement. Overall, this work shows the potential of advanced deep-learning architectures, like SwinIR and MambaIR, to improve access to affordable high-resolution satellite imagery.

Contents

Declaration i

Abstract iii

1 Introduction 1

1.1 Background and Motivation 1

1.2 Super-Resolution Approaches 2

1.3 Contributions 3

2 Literature Review 5

2.1 Traditional Super-Resolution 5

2.2 Deep Learning-Based Super-Resolution 6

2.2.1 Convolutional Neural Networks (CNNs) 6

2.2.2 Generative Adversarial Networks (GANs) 7

2.2.3 Attention-Based Super-Resolution 7

2.2.4 Mamba-Based Super-Resolution 8

3 Requirements Specification 9

3.1 Project Requirements based on Primary Objectives 9

3.2 Project Requirements based on Secondary Objectives 10

3.3 Project Requirements based on Tertiary Objectives 10

4 Software Engineering Process 13

4.1 Development Approach 13

4.2 Python 14

4.3 PyTorch . . . 14
4.4 BasicSR . . . 14
4.5 Hugging Face and MambaSSM . . . 14

5 Ethiscs **15**

6 Design **17**

6.1 SwinIR . . . 17
6.1.1 Shallow Feature Extraction . . . 18
6.1.2 Deep Feature extraction . . . 18
6.1.3 Image Reconstruction . . . 19
6.1.4 Loss Function . . . 20
6.1.5 Training Design . . . 20
6.2 MambaIR . . . 20
6.2.1 Shallow Feature Extraction . . . 21
6.2.2 Deep Feature Extraction . . . 21
6.2.3 Image Reconstruction . . . 23
6.2.4 Loss Function . . . 23
6.2.5 Training Design . . . 23
6.3 Problem Formulation . . . 23
6.4 Models and Comparison Criteria . . . 24
6.5 Dataset . . . 24
6.6 Training Setup . . . 25
6.7 Evaluation Metrics . . . 26
6.8 Experiment Design . . . 26
6.9 Ablation Study Design . . . 26

7 Implementation **29**

7.1 Dataset . . . 29
7.2 BaseSuperResolutionModel . . . 30
7.3 SwinIR . . . 31
7.4 MambaIR . . . 32

7.5 ESRGAN . . . 33
7.6 SRResNet . . . 33
7.7 Bicubic Interpolation (Baseline) . . . 34
7.8 Training and Testing . . . 34

8 Experimental Results **35**
8.1 Quantitative Results . . . 35
8.2 Qualitative Analysis . . . 36
8.3 Ablation Study . . . 37
8.3.1 Effect of CLIPScore Loss on Model Performance . . . 37
8.3.2 Effect of 2× Model Weight Initialisation . . . 38

9 Discussion **41**
9.1 Comparative Analysis of Model Architectures . . . 41
9.2 MambaIR vs. SwinIR . . . 42
9.3 Perceptual Quality of ESRGAN . . . 43
9.4 Bicubic's SSIM Performance . . . 44
9.5 Impact of CLIPScore Loss . . . 44
9.6 Effect of 2× Pre-training . . . 45
9.7 Dataset and Domain Challenges . . . 46
9.8 Key Findings . . . 48

10 Evaluation and Critical Appraisal **49**
10.1 Primary Objectives . . . 49
10.2 Secondary Objectives . . . 50
10.3 Tertiary Objectives . . . 50
10.4 Related Work . . . 50

11 Conclusion **51**

References **53**

A Testing Summary **57**

B User Manual **59**

B.1 Repository Structure 59

B.2 Installation and Setup 60

B.3 Running the Project 60

B.4 Training a Model 60

B.5 Evaluating a Model 61

B.6 Visualising Model Outputs 61

B.7 Dataset 61

B.8 Modifying Configuration Files 62

Chapter 1

Introduction

1.1 Background and Motivation

High-resolution (HR) satellite imagery is becoming increasingly important for a wide range of scientific and non-commercial applications. These include environmental monitoring, which aims to use HR images to track deforestation or monitor the impacts of climate change; urban planning, which often uses satellite imagery to support decision-making making; and agriculture, where HR satellite images are relevant to, among others, precision farming, crop health assessment, and yield prediction.

Despite its importance, access to HR satellite imagery remains expensive. Most high-quality satellite images are proprietary and, therefore, not suitable for academic or scientific use. Although some freely available satellite imagery exists, such as Sentinel and Landsat imagery, it is almost always imagery from small satellites that is limited in spatial resolution, typically around 10 to 20 meters per pixel. Small satellites cannot produce high-resolution satellite images because they are constrained by their physical size and the size of their equipment.

For many applications, this resolution is insufficient. As a consequence, super-resolution (SR) techniques have emerged as an essential approach to enhancing freely available low-resolution satellite imagery(Illarionova et al., 2023). SR aims to reconstruct a high-resolution image from a low-resolution input with minimal loss in detail. More formally, satellite-image super-resolution attempts to infer a high-resolution image I_{HR} from a corresponding

low-resolution input I_{LR} by an upscaling factor s, such that $I_{HR} \approx \mathcal{F}(I_{LR}; s)$.

Recent advances in deep learning have significantly improved the performance of SR methods. In particular, transformer-based architectures have outperformed traditional convolutional neural networks (CNNs) in a wide variety of computer vision tasks. Their success is largely due to their ability to model long-range dependencies via the attention mechanism, allowing them to capture global context much more effectively than CNNs, constrained by the convolution operation's localised nature.

However, applying attention globally, as in the Vision Transformer architecture, is computationally expensive, with quadratic complexity in image size. Two promising approaches have been developed to improve computational efficiency while retaining global context encoding.

The Swin Transformer(Liu et al., 2021) introduces shifted windowed attention that applies attention locally within each non-overlapping window, then shifts these windows in following layers to also encode cross-window dependencies.

An alternative, Mamba, abandons attention altogether in favour of state-space models. The Mamba model uses an efficient selective scan mechanism to process long sequences, offering linear complexity while retaining the ability to encode global dependencies effectively.

Both architectures have already demonstrated success in natural image super-resolution with models like SwinIR and MambaIR(Liang et al., 2021; Guo et al., 2024). However, their application to satellite image super-resolution, particularly in real-world settings using publicly available data, remains unexplored.

This work aims to address this by investigating the effectiveness of SwinIR and MambaIR models for super-resolution of low-resolution Sentinel-2 satellite images to high-resolution ground truth NAIP satellite images, as visualised in Figure 1.1. The goal is to explore whether these architectures can help improve access to affordable, high-resolution satellite imagery for scientific and non-commercial applications.

1.2 Super-Resolution Approaches

Super-resolution techniques can be broadly split into two approaches: multi-image super-resolution (MISR), which combines several low-resolution images of the same scene to reconstruct a high-resolution image, and single-image super-resolution (SISR), which enhances

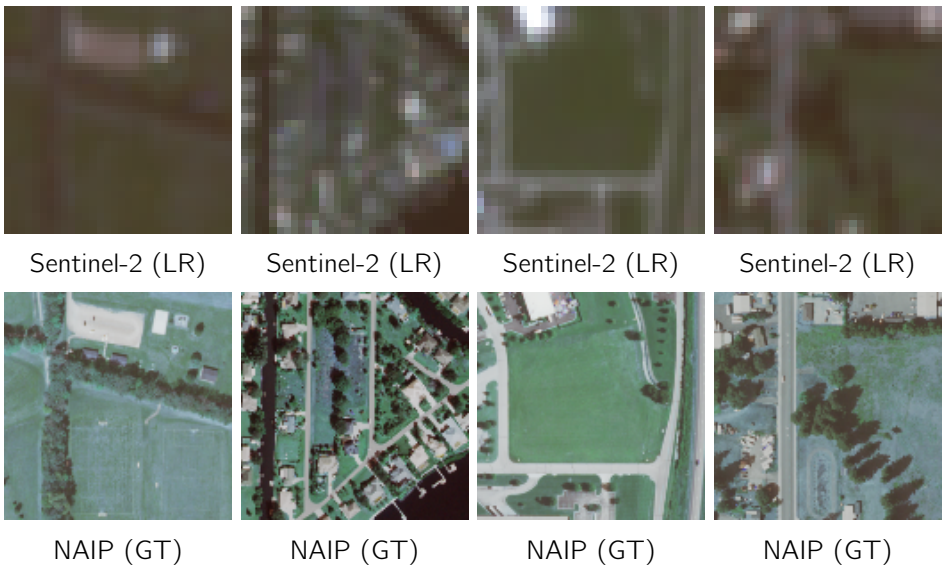


Figure 1.1: Illustration of Sentinel-2 vs NAIP images

the resolution of a single low-resolution image. There are three main methods for SISR: interpolation-based, reconstruction-based and learning-based. Interpolation-based techniques are the simplest, but they generally produce overly smooth results without new details. The most commonly used interpolation-based technique is bicubic interpolation. Reconstruction-based methods use prior knowledge about image structure to better reconstruct high-frequency details. They often use optimisation algorithms. Finally, learning-based methods are the newest and most popular methods. Learning-based methods use machine learning, mainly deep neural networks, to learn a mapping from low-resolution to high-resolution examples.

This research focuses exclusively on single-image super-resolution using learning-based methods, which are considered the state-of-the-art in satellite-image super-resolution.

1.3 Contributions

The key contributions of this work are as follows:

- Implementation and adaptation of SwinIR and MambaIR models for single satellite image super-resolution using the BasicSR framework.
- Empirical evaluation of SwinIR and MambaIR against SRResNet and ESRGAN on real-world satellite image data from the S2-NAIP dataset.

- Analysis and interpretation of the results, including visual comparisons, quantitative evaluation, and discussion of the trade-off between the evaluated models.
- Investigation into the impact of main hyperparameter design choices, including the inclusion of CLIPScore loss and the effect of initialising weights from a pre-trained $2\times$ model.

Chapter 2

Literature Review

2.1 Traditional Super-Resolution

Traditional super-resolution methods can be broadly classified into interpolation-based and reconstruction-based approaches. These were the earliest techniques developed for super-resolution and still remain used in some real-time and resource-constraint environments.

Interpolation-based techniques such as nearest-neighbour, bilinear, and bicubic interpolation estimate missing pixel values based on neighbouring pixel intensities. These methods are simple to implement, computationally efficient, and widely used in real-world applications. However, their efficiency comes at the cost of non-adaptivity, meaning they cannot adapt to different types of images. Additionally, they generally produce overly smooth outputs, with no new details compared to the low-resolution images. Their drawbacks, therefore, make them less effective for non-real-time high-quality applications.

Reconstruction-based methods combine prior assumptions about image characteristics, such as smoothness or continuity, with optimisation algorithms to reconstruct high-resolution images. While they are more flexible than interpolation-based techniques and can produce sharper, more detailed images, they are computationally expensive and typically degrade rapidly with increasing scale factor(Yang et al., 2019).

2.2 Deep Learning-Based Super-Resolution

Deep learning-based super-resolution (SR) methods are the most recent and innovative single-image super-resolution (SISR) approach. These methods learn a mapping from low-resolution (LR) to high-resolution (HR) images using a large number of examples. Compared to traditional methods, they provide significantly better reconstruction quality and can be optimised to specific image types. The most prominent approaches include CNN-based methods and GAN-based methods. Additionally, two novel approaches are investigated: attention-based methods and Mamba-based methods.

2.2.1 Convolutional Neural Networks (CNNs)

Convolutional neural networks were the first successful deep learning architectures applied to SISR. Proposed by Dong et al. (2016), SRCNN was the first CNN-based model successfully applied to super-resolution. SRCNN showed significant performance improvements over traditional methods by learning end-to-end mappings between low- and high-resolution images with a CNN. Building upon SRCNN, Kim et al. (2016) introduced the Very Deep Super-Resolution (VDSR) model, a much deeper CNN architecture inspired by VGG(Simonyan and Zisserman, 2015). Incorporating residual connections proposed by He et al. (2016), VDSR was able to become much deeper while avoiding the vanishing gradient problem. Another crucial improvement came with Efficient Sub-Pixel Convolutional Neural Network (ESPCN)(Shi et al., 2016), which avoided the computational inefficiency of performing super-resolution on an upscaled low-resolution (LR) image by instead performing super-resolution in the LR space and upscaling at the end. This was made possible using an efficient subpixel convolution layer known as the Pixel Shuffle layer. Later Ledig et al. (2017) proposed SRResNet, a deep residual CNN that integrates perceptual loss computed from a pre-trained VGG network. By including the perceptual losses Ledig et al. (2017) produced more realistic textures, which are difficult to produce with pixel-wise loss only.

Despite their early success, CNN-based methods are limited in capturing long-range dependencies due to the inherently local nature of convolutional operations.

2.2.2 Generative Adversarial Networks (GANs)

GAN-based methods introduced by Goodfellow et al. (2014) have also been successful in many computer vision tasks. GAN-based super-resolution methods use a generator-discriminator structure, which makes them particularly effective in producing realistic super-resolved images.

Ledig et al. (2017) introduced the first GAN-based SISR model, cobming adverarial training with a perceptual loss(Johnson et al., 2016), SRGAN was able to produce high-quality, realistic-looking images. SRGAN was improved upon by Wang et al. (2018), who introduced several improvements in the Enhanced Super-Resolution Generative Adversarial Network (ESRGAN). Among the improvements were a new Residual-in-Residual Dense Block (RRDB) design, which further helps model training by avoiding the vanishing gradient problem, an improved relativistic discriminator, and a perceptual loss computed that constrained features before activation. Together, these implements led to both better qualitative and quantitative results. Later, in 2020, Pineda et al. (2020) successfully applied a GAN model, based on the SRGAN architecture, to Sentinel-2 satellite imagery, showing the potential of GAN models in satellite image super-resolution.

While GAN-based super-resolution methods excel at producing high-quality, photo-realistic images, they are computationally expensive and often suffer from training instability. Furthermore, their outputs occasionally contain hallucinated artefacts not present in the ground truth images.

2.2.3 Attention-Based Super-Resolution

The attention mechanism represents an alternative to CNNs, which excels at modelling global dependencies in text and images.

Proposed by Dai et al. (2019), a Second-order Attention Network (SAN) introduces a novel second-order channel attention mechanism that effectively captures long-range contextual dependencies and feature correlations, outperforming previous SISR methods in quantitative metrics and visual quality. Niu et al. (2020) enhanced the channel attention mechanisms by modelling holistic interdependencies among network layers, further improving super-resolution performance with the proposed Holistic Attention Network (HAN). The UFormer architecture, proposed by Wang, Cun, Bao, Zhou, Liu and Li (2022), reduces the computational complexity

associated with attention while maintaining good performance. To do so, UFormer uses a U-shaped encoder-decoder architecture with a window-based mechanism based on a similar idea to the shifted window attention mechanism in the Swin Transformer. Building upon the Swin Transformer architecture, Liang et al. (2021) developed SwinIR, which incorporates Residual Swin Transformer Blocks for deep feature extraction, achieving state-of-the-art performance on super-resolution and image restoration tasks.

Despite its good performance, the global attention mechanism is computationally expensive for most super-resolution applications due to its quadratic complexity. Windowed or shifted windowed attention, as seen in SwinIR and UFormer, avoids this cost but compromises on the quality of modelling of global dependencies as they apply attention locally rather than globally.

2.2.4 Mamba-Based Super-Resolution

The Mamba architecture represents a novel approach to representing global context. It is based on state-space models (SSMs) and offers a promising balance between the efficiency of CNNs and the global modelling capacity of Transformers.

The Mamba architecture, introduced by Gu and Dao (2023), proposes an efficient selective scan algorithm that allows SSMs to focus selectively on different parts of the input while maintaining linear scaling with sequence length. This property makes the Mamba architecture very attractive for long-range dependency modelling. Liu et al. (2024) applied the Mamba architecture to vision tasks such as ImageNet classification, proposing bidirectional Mamba blocks that outperform Vision Transformers in both efficiency and accuracy. Guo et al. (2024) extended Vision Mamba to super-resolution, proposing MamaIR as a benchmark model that achieves state-of-the-art results in both PSNR and perceptual metrics, surpassing attention-based models like SwinIR. To further optimise computational efficiency in high-resolution super-resolution, Qiao et al. (2024) introduced Hierarchical Mamba, which reduces the number of directional scans needed for accurate global context representation through block hierarchies and directional alternation.

While still relatively new, Mamba-based approaches are a promising alternative for SR tasks due to their ability to efficiently model long-range spatial dependencies. However, their applicability to remote sensing super-resolution remains largely unexplored.

Chapter 3

Requirements Specification

A number of project requirements can be derived from the original objectives defined at the beginning of the project. These requirements help guide the research and development process. The project requirements are structured and assigned priority according to their corresponding objectives, with primary objectives considered crucial for demonstrating the project's success and secondary and tertiary objectives providing valuable extensions and improvements.

3.1 Project Requirements based on Primary Objectives

Requirements derived from the primary objectives are assigned the highest level of priority. These objectives are core to the project. They include understanding model architectures, data acquisition, model development, training, and comparative evaluation. The following primary objectives are established:

1. An exploration of the transformer and Mamba architectures to build knowledge required for implementation.

2. A collected and preprocessed dataset of low-resolution Sentinel-2 and corresponding high-resolution NAIP image pairs for 4× super-resolution.

3. An implemented deep learning model for single image super-resolution based on a transformer architecture (SwinIR).

4. An implemented deep learning model for single image super-resolution based on a Mamba architecture (MambaIR).

5. A training pipeline that trains the SwinIR and MambaIR models for 4× super-resolution on the collected data.

6. An evaluation of the trained models using relevant quantitative metrics (e.g., PSNR, SSIM, CLIP) and visual inspection.

7. A performance comparison of the trained models against existing super-resolution methods, particularly bicubic interpolation, SRResNet (CNN-based), and ESRGAN (GAN-based).

3.2 Project Requirements based on Secondary Objectives

The secondary objectives mainly concern refining the performance and usability of the models. While not as important for the project's success, they are important for improving the effectiveness and usability of the implemented models. Requirements derived from these objectives are, therefore, given medium priority.

1. An ablation study exploring the impact of main model design decisions, such as the integration of CLIPScore loss and the effect of pre-training from a 2× super-resolution model.

2. A user-facing interface or API for demonstrating the functionality of the super-resolution models.

3.3 Project Requirements based on Tertiary Objectives

Tertiary objectives explore long-term considerations, such as deployment and generalisation. These objectives are not core to the project's success but are valuable avenues for future development. Thus, they are given the lowest priority.

1. A small-sized and optimised version of the trained models for deployment on edge devices with limited computational resources.

2. An investigation of the transferability of the model to other satellite image types and resolutions.

These requirements should be completed in order of priority. The high-priority requirements should be achieved to demonstrate the project's success, while the medium- and low-priority ones are useful for further refinement and practical use.

Chapter 4

Software Engineering Process

A core part of the project was the development of SwinIR and MambaIR, deep learning models for Sentinel-2 satellite image super-resolution. During the development, good software engineering practices were followed, focusing on reproducibility, modularity and clarity. A number of tools and libraries were used in the implementation and to manage the software.

4.1 Development Approach

The project followed the waterfall development approach. This was influenced by the nature of the project, with clearly defined distinct stages that build sequentially upon one another (data acquisition, model implementation, training, and evaluation).

The development cycle consisted of four main stages: planning, design, implementation and testing. The planning and design phases mainly consisted of making high-level architectural decisions and designing the training and evaluation process. These decisions were influenced mainly by the choice to implement the SwinIR and MambaIR models following their proposed design and by using the BasicSR framework(Wang, Xie, Yu, Chan, Loy and Dong, 2022). The Implementation involved translating the MambaIR and SwinIR into PyTorch code. Finally, the testing phase involved debugging and ensuring that implemented models produced stable and meaningful results.

Version control was done through a private GitHub repository, which was selected as it provided a simple way to record changes and experiment with features on separate branches. For readability, code was documented throughout the development process.

A rigorous scientific approach was followed during training and evaluation. Model hyperparameters and design decisions were tested systematically through configuration files. Each experiment and evaluation was logged, and observations were written down for later discussion.

4.2 Python

Python was the primary programming language used in the project. It was selected for its wide range of machine learning and image processing libraries. Additionally, its syntax and ecosystem allowed for rapid prototyping.

4.3 PyTorch

The PyTorch deep learning framework was selected as the primary framework for model development. It was chosen because it provides a clean and intuitive way to define and train models. Additionally, it offers abstraction over low-level details and strong community support. All models were implemented using PyTorch.

4.4 BasicSR

BasicSR is a popular open-source framework for image and video restoration(Wang, Xie, Yu, Chan, Loy and Dong, 2022). It was used to structure model manipulation, training and evaluation in a unified, robust way. By registering the implemented models with BasicSR, the framework's scripts were used to configure, train and consistently evaluate models.

4.5 Hugging Face and MambaSSM

For the shifted window attention mechanism used by SwinIR, the Hugging Face 'PyTorch Image Models' repository implementation was adapted. Additionally, for the MambaIR implementation, the `mamba-ssm` package was used as it offers an efficient implementation of the selective scan algorithm.

Chapter 5

Ethiscs

There were no ethical considerations raised by the project. The preliminary self-assessment form was completed, indicating that there were no ethical issues raised. So, ethical application was not required.

Chapter 6

Design

In satellite image super-resolution and many other computer vision applications, a clear trade-off between global context modelling and computational efficiency emerges(Guo et al., 2024). Traditional CNN-based architectures, such as SRResNet, are effective at capturing local features and are computationally efficient due to the parallelisable nature of convolutions. However, their limited receptive field makes it difficult to model long-range dependencies, which are often important in satellite image super-resolution.

Recent transformer-based models offer improved global context modelling via self-attention but at a significant computational cost. SwinIR avoids this by using the Swin Transformer architecture, which applies attention within local windows and uses a shifted window mechanism to capture cross-window dependencies.

In contrast, MambaIR takes a different approach and uses state-space models (SSMs) to model long-range dependencies without attention. Its efficient selective scan algorithm allows it to capture global context with linear-time complexity.

This project investigates the relative strengths of SwinIR and MambaIR and compares them to established methods such as the CNN-based SRResNet and the GAN-based ESRGAN.

6.1 SwinIR

The first investigated architecture is SwinIR. It was selected for its state-of-the-art performance in image restoration tasks(Liang et al., 2021). The architecture builds on the Swin Transformer backbone and balances local and global feature modelling. This work implements and evaluates

the SwinIR architecture for the super-resolution of single satellite images.

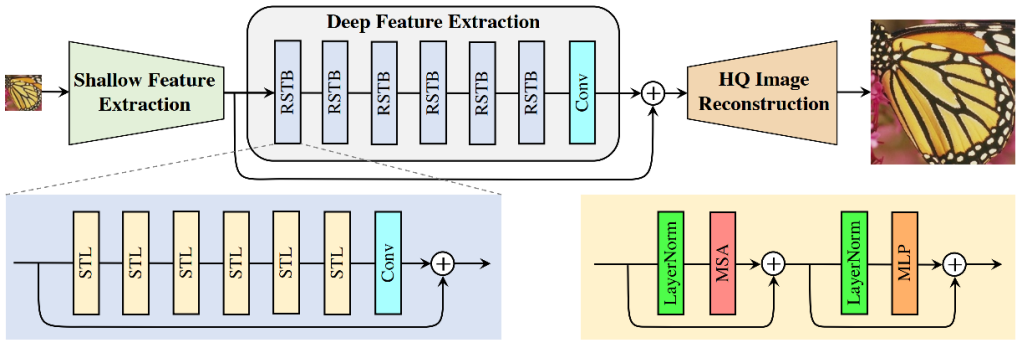


Figure 6.1: SwinIR architecture overview. Image from Liang et al. (2021).

As illustrated in Figure 6.1, SwinIR is composed of three modules: shallow feature extraction, deep feature exaction and image reconstruction. Each was implemented according to the original design to try to replicate its success on satellite image super-resolution.

6.1.1 Shallow Feature Extraction

The SwinIR architecture first applies a 3×3 convolutional layer that projects the low-resolution input image $I_{LQ} \in \mathbb{R}^{H \times W \times C_{in}}$ to a high-dimensional representation with C channels ($F_0 \in \mathbb{R}^{H \times W \times C}$):

$$F_0 = H_{SF}(I_{LQ}) \tag{6.1}$$

where $H_{SF}(\cdot)$ represents the shallow feature extraction function and C is the number of intermediate channels. This layer allows the model to extract basic structural information and allows more stable training(Liang et al., 2021).

6.1.2 Deep Feature extraction

The extracted shallow features F_0 are then processed by a series of K Residual Swin Transformer Blocks (RSTBs), each containing multiple Swin Transformer Layers (STLs) and a convolutional layer. The transformation of an RSTB can be represented as:

$$F_i = H_{RSTB_i}(F_{i-1}), \quad i = 1, 2, \ldots, K, \tag{6.2}$$

where $H_{RSTB_i}(\cdot)$ represents the i-th RSTB. The final deep features are obtained by:

$$F_{DF} = H_{CONV}(F_K) + F_K, \tag{6.3}$$

where $H_{CONV}(\cdot)$ is a final convolutional layer that helps improve the aggregation of features in the reconstruction module. SwinIR also includes a residual connection ($+F_K$), which helps preserve some low-frequency details and improve the flow of gradient in the network(Liang et al., 2021; He et al., 2016).

Each Swin Transformer Layer adapts the standard multi-head self-attention of the original Transformer layer(Vaswani et al., 2017) into a local window-based self-attention mechanism, which divides an input into non-overlapping local windows of size $M \times M$ and applies self-attention locally within each window. For a features $X \in \mathbb{R}^{M^2 \times C}$ within a single window, the self-attention is defined as:

$$\text{Attention}(Q, K, V) = \text{SoftMax}\left(\frac{QK^T}{\sqrt{d}} + B\right)V \tag{6.4}$$

where $Q = XP_Q$, $K = XP_K$, $V = XP_V$ are the query, key and value projections of the learned shared weight matrix, d is the attention head dimension, and B is the relative positional encoding representing the positions of individual patches. To apply attention between windows, SwinIR uses the same approach as the Swin Transformer(Liu et al., 2021), alternating between regular and shifted window partitioning, where shifted window partitioning means shifting by $\left(\lfloor \frac{M}{2} \rfloor, \lfloor \frac{M}{2} \rfloor\right)$.

6.1.3 Image Reconstruction

Finally, the high-resolution image is reconstructed by combining the extracted shallow and deep features:

$$I_{HQ} = H_{REC}(F_0 + F_{DF}), \tag{6.5}$$

where $H_{REC}(\cdot)$ is the reconstruction function. SwinIR uses subpixel convolution, also known as the Pixel Shuffle layer, introduced by Shi et al. (2016) to efficiently increase the spatial resolution of the feature maps(see Section 2.2.1). Additionally, another residual

connection is used to help stabilise the training and improve the quality of the reconstructed image(Liang et al., 2021).

6.1.4 Loss Function

SwinIR is optimised with the L1 loss function, which measures the absolute difference between the reconstructed image I_{HQ} and the ground truth image I_{GT}:

$$\mathcal{L} = \|I_{HQ} - I_{GT}\|_1. \tag{6.6}$$

L1 loss was shown to produce sharper images compared to the L2 loss, hence it's use by SwinIR(Ledig et al., 2017).

6.1.5 Training Design

With the goal to reproduce the success of SwinIR on super-resolution, initially, the hyperparameters proposed by Liang et al. (2021) and detailed in the supplementary material of the SwinIR paper were used. These were then iteratively changed to establish hyperparameter settings that yield a strong single-image super-resolution performance.

Following the design of SwinIR, the model was configured with 6 RSTBs, each containing 6 STLs, with a window size of 8×8. The feature dimension C is set to 180, and each attention layer has 6 heads. The model was trained using a batch size of 4, compared to 32 proposed by SwinIR, because of less GPU memory. The learning rate was initialised at 1×10^{-4} since this configuration allowed the model to converge better within the $300,000$ training iterations. The learning rate was halved at milestones: [125K, 200K, 225K, 237.5K] iterations. Finally, following the design of SwinIR, Adam optimiser was used with parameters $\beta_1 = 0.9$ and $\beta_2 = 0.99$.

6.2 MambaIR

The second architecture explored in this study is MambaIR, which was selected for its unique approach to capturing global context (SSMs instead of attention). MambaIR uses state-space models (SSMs) to effectively model long-range dependencies with linear complexity.

The MambaIR architecture is again implemented and evaluated for single satellite image super-resolution.

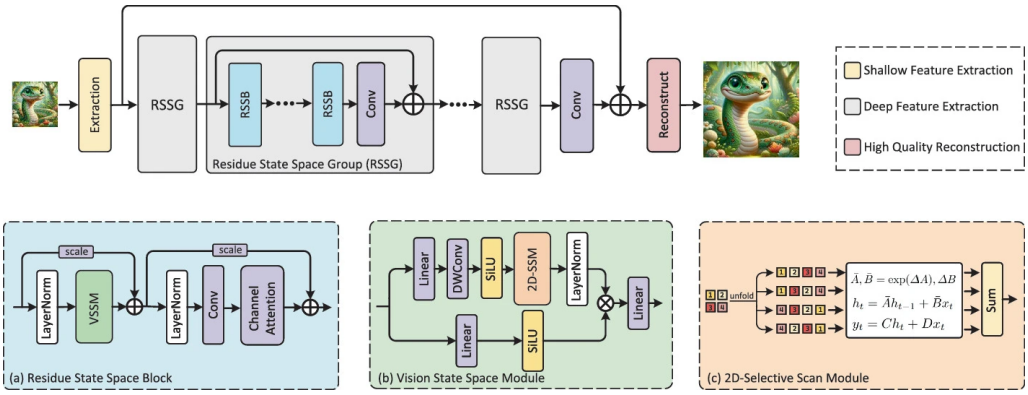


Figure 6.2: MambaIR architecture overview. Image from Guo et al. (2024).

As displayed in Figure 6.2, MambaIR is again made of three primary modules: shallow feature extraction, deep feature extraction, and image reconstruction. While this high-level structure is equivalent to that of SwinIR, MambaIR differs by the selection of components used within these modules, particularly in the deep feature extraction module.

6.2.1 Shallow Feature Extraction

Like SwinIR, MambaIR uses a 3×3 convolutional layer for shallow feature extraction, mapping the low-resolution input image $I_{LQ} \in \mathbb{R}^{H \times W \times C_{in}}$ to a high-dimensional representation $F_0 \in \mathbb{R}^{H \times W \times C}$.

6.2.2 Deep Feature Extraction

While SwinIR relies on Residual Swin Transformer Blocks (RSTBs), MambaIR uses Residual State-Space Groups (RSSGs) to capture both local and global feature dependencies. Each RSSG contains multiple Residual State-Space Blocks (RSSBs). At the end of each RSSG, an additional 3×3 convolutional layer is applied to refine the features before passing them on.

To make use of the long-range modelling capabilities of the Mamba-style state-space models, in each RSSB, MambaIR implements a custom-designed structure and thus departs from the transformer-style Attention $\rightarrow$ MLP structure.

Given an input feature F_D^l at layer l, the first block in RSSB applies a layer normalisation

followed by the Vision State-Space Module (VSSM):

$$Z^l = \text{VSSM}(\text{LN}(F_D^l)) + s \cdot F_D^l \tag{6.7}$$

where $s \in \mathbb{R}^C$ is a learnable scale factor applied to the residual connection. The VSSM itself allows for two parallel data flows. One branch applies a linear projection, followed by depth-wise convolution, an activation function, a 2D-Selective Scan Module, and a layer normalisation. The second branch performs another linear projection and applies an activation function (SiLU). The outputs of the two branches are combined with an element-wise (Hadaramrd) product:

$$X_1 = \text{LN}(\text{2D-SSM}(\text{SiLU}(\text{DWConv}(\text{Linear}(X))))) \tag{6.8}$$

$$X_2 = \text{SiLU}(\text{Linear}(X)) \tag{6.9}$$

$$X_{\text{out}} = \text{Linear}(X_1 \odot X_2) \tag{6.10}$$

Here, DWConv is a depth-wise convolution, and $\odot$ is the element-wise product. The 2D-Selective Scan Module (2D-SSM) aims to capture the 2D spatial information of an image by applying the state-space model on an image flattened in four directions: top-left to bottom-right, bottom-right to top-left, top-right to bottom-left, and bottom-left to top-right. This way, the state-space model captures long-range dependencies along all four image directions. Finally, 2D-SSM sums and reshapes all four sequences to recover the 2D structure (Guo et al., 2024).

SSM inherently operate in a single dimension, so a convolutional block is added following the VSSM to effectively capture local pixel interactions. To further help with this, MambaIR also added a Channel Attention (CA) module after the mentioned convolution. This helps the model focus on channels with valuable information while suppressing redundant ones(Guo et al., 2024). The convolutional block and the channel attention can be grouped into a single block, represented as:

$$F_D^{l+1} = \text{CA}(\text{Conv}(\text{LN}(Z^l))) + s' \cdot Z^l \tag{6.11}$$

where Z^l is the output of the first block in RSSB containing the VSSM, and s' is another learnable scale factor.

6.2.3 Image Reconstruction

Just as in the case of SwinIR, the high-resolution image is reconstructed by combining the extracted shallow and deep features:

$$I_{HQ} = H_{REC}(F_0 + F_{DF}), \tag{6.12}$$

where $H_{REC}(\cdot)$ is the reconstruction function. The image reconstruction in MambaIR is equivalent to that in SwinIR, applying a subpixel convolution (pixel shuffle) to upsample and a residual connection to improve training stability and reconstruction accuracy.

6.2.4 Loss Function

MambaIR, like SwinIR, is trained using the L1 loss function:

$$\mathcal{L} = \|I_{HQ} - I_{GT}\|_1 \tag{6.13}$$

6.2.5 Training Design

To allow the SwinIR and MambaIR models to be easily compared, MambaIR was trained using the same procedure and with hyperparameter values that were as similar as possible. The main differences include using a patch size of 1 instead of 8, an MLP ratio of 2 instead of 4, and introducing a hidden state dimension $d_{state} = 16$, all specific to the MambaIR architecture. MamaIR was trained the same way as SwinIR with the same training hyperparameters(see section 6.1.5).

6.3 Problem Formulation

Single-image super-resolution (SISR) can be formally defined as learning a mapping $\mathcal{F}_\theta : \mathbb{R}^{H\times W\times C} \to \mathbb{R}^{sH\times sW\times C}$, where s is the scale factor (here $s = 4$), and θ are the model

parameters. Given a low-resolution image I_{LR}, the model predicts a corresponding high-resolution super-resolved image $\hat{I}_{HR}$ that approximates the ground truth I_{HR}.

According to Hadamard's definition, SISR is an ill-posed inverse problem because multiple high-resolution solutions may correspond to a single low-resolution input(Yan et al., 2024). However, this underdetermination makes it a problem suitable for a deep learning-based approach.

6.4 Models and Comparison Criteria

In this work, SwinIR and MambaIR are adapted for the task of 4× single satellite image super-resolution and are compared against established models, a CNN-based SRResNet model, and a GAN-based state-of-the-art ESRGAN model, as well as a bicubic interpolation baseline. We evaluate these models on the following criteria:

- **Quantitative metrics:** PSNR, SSIM, and CLIPScore.
- **Qualitative performance:** Visual quality and spatial coherence.
- **Ablation study:** The effect of CLIPScore loss and 2× pre-training.

6.5 Dataset

A major challenge in supervised deep learning applied to satellite-image super-resolution is the lack of ground truth high-resolution images. While some previous works have synthetically degraded HR images to artificially create LR-HR pairs(Pineda et al., 2020; Fu et al., 2018), this setup, however, tends to overestimate model performance due to unrealistic assumptions.

Two alternatives exist in the literature: in-band super-resolution (e.g., between 10m and 20m Sentinel-2 bands (Wagner et al., 2019)) and cross-sensor super-resolution using temporally aligned high-resolution imagery from another sensor (Zabalza and Bernardini, 2022; Salgueiro Romero et al., 2020). Since Sentinel-2 does not offer spectral bands with a 4× resolution difference, the cross-sensor approach is used in this work with the S2-NAIP dataset(Wolters et al., 2023).

The S2-NAIP dataset consists of Sentinel-2 and NAIP image pairs from the continental United States. NAIP images are downsampled to 25% of their original resolution for 4× super-resolution. Only RGB bands are used for both inputs and the ground truth since NAIP only operates in the RGB bands. Because the full dataset mostly contains monotonous landscapes, the `train_urban_set` subset is used instead. It includes ≈ 1.1 million Sentinel-2/NAIP image pairs within a 5km radius of U.S. cities with a population over 50,000.

6.6 Training Setup

All models are trained for 300,000 iterations, which empirical evaluation found to be sufficient for convergence. Cross-validation performance typically peaked before 150,000 iterations, with no improvements past 200,000. Additionally, because of the large number of image pairs in the S2-NAIP dataset, only a subset of 50,000 randomly selected images is used for training.

The primary loss used for SwinIR, MambaIR, and SRResNet is the pixel-wise L_1 loss:

$$\mathcal{L}_{L_1} = \| I_{HQ} - I_{GT} \|_1. \tag{6.14}$$

Inspired by Wolters et al. (2023), a secondary CLIPScore loss is used in the ablation study. CLIPScore is based on the cosine similarity between the CLIP embeddings of the predicted and ground truth images(Hessel et al., 2021):

$$\mathcal{L}_{CLIP} = \max\left(\cos\left(\phi(I_{HQ}), \phi(I_{GT})\right), 0\right) \tag{6.15}$$

where $\phi(\cdot)$ denotes the ViT-B-16-SigLIP-256 CLIP encoder (cross-validation) or the siglip-ViT-SO400M-14 CLIP encoder (testing). The final loss is a weighted sum:

$$\mathcal{L}_{total} = \mathcal{L}_{L_1} + \mathcal{L}_{CLIP}. \tag{6.16}$$

ESRGAN, in addition to these two losses, also optimises a GAN loss.

6.7 Evaluation Metrics

Model performance is evaluated using PSNR, SSIM, and CLIPScore. PSNR and SSIM are standard pixel-wise metrics, while CLIPScore measures perceptual similarity in the CLIP embedding space. Although PSNR and SSIM are widely used, previous works report they are poorly correlated with human perception, particularly for super-resolution(Aybar et al., 2024; Ledig et al., 2017). Therefore, CLIPScore is used to obtain a more holistic assessment that better aligns with human preference, $\approx 80\%$ agreement compared to $\approx 50\%$ for PSNR and SSIM(Wolters et al., 2023).

6.8 Experiment Design

The evaluation consists of three parts: quantitative comparison against baselines, qualitative assessment, and an ablation study. For training, $50,000$ random patch pairs are selected from the `train_urban_set`. A cross-validation set (`small_val_set`) of 256 urban image pairs is used for in-training evaluation every $1,000$ iterations. The final evaluation is performed on the full `val_set`, which consists of $8,192$ image pairs.

Training was performed using a batch size of 4 on a single NVIDIA RTX 3060 GPU. The learning rate for SwinIR and MambaIR is initialised to 1×10^{-4} and halved at [125,000, 200,000, 225,000, 237,500] iterations. For ESRGAN, the learning was also initialised to 1×10^{-4}, but no learning rate halving was performed. For SRResNet, a learning rate initialised to 1×10^{-4} was also used, combined with a cosine annealing scheduler with restart at $250,000$ iterations was used. These hyperparameters were selected because they were either proposed in their corresponding works or provided better empirical performance on the S2-NAIP dataset.

6.9 Ablation Study Design

To investigate the contribution of specific hyperparameter settings, an ablation study on two main hyperparameters was performed.

CLIPScore Losss. The effect of incorporating CLIPScore as a secondary loss was investigated. This allowed for the assessment of whether a more semantic/perceptual-oriented

loss would help nudge the models toward better predictions. In this study, models trained on 300,000 iterations with 50,000 random patch pairs selected from the `train_urban_set` and their standard loss were compared to models trained for 300,000 iterations on the same data with their standard loss plus CLIPScore loss.

2× **Pre-training.** The second ablation study explores the benefit of initialising models with weights trained on an easier 2× super-resolution task. A model is trained for 200,000 iterations on downsampled NAIP ground truth images (from 2.5m to 5m), then fine-tuned for 100,000 iterations on the 4× super-resolution task. A subset of 30,000 2× path pairs is used for pre-training, and 10,000 4× patch pairs are used for fine-tuning. Additionally, the learning rate of each model is doubled for the 2× pre-training. This setup allows for testing whether pre-training on a simpler task improves convergence and performance.

Chapter 7

Implementation

This section describes the implementation of the models and the dataset used in the experiment.

Building on the architectural designs presented in Chapter 6, the implementation of SwinIR, MambaIR, and the models they are compared with are described. All models were implemented using the PyTroch framework. Additionally, all models were integrated into the BasicSR framework(Wang, Xie, Yu, Chan, Loy and Dong, 2022) for unified, consistent training and evaluation.

7.1 Dataset

The S2-NAIP dataset was used for training and evaluation. A dataset loader script was adopted and modified from the official S2-NAIP repository(Wolters et al., 2023). Two main changes were added. First, the ground truth NAIP image degrading was implemented to support 2× super-resolution. If the scale factor was set to 2×, ground truth images were degraded from 128×128 to 64×64. Additionally, the normalisation process for input images was moved from the model implementations to the dataset pipeline, simplifying the model logic.

7.2 BaseSuperResolutionModel

A shared superclass `BaseSuperResolutionModel` was created in `archs/utils/basic_sr.py` to avoid code duplication. This class encapsulates the common components of SwinIR and MambaIR, including shallow feature extraction and image reconstruction. The model was implemented using PyTorch's `nn.Module`. Following good PyTorch software engineering practices, layers were defined in the `__init__` method and the flow of data was defined in the `forward` method. As suggested by the SwinIR and MambaIR architectures, image reconstruction and the upsampling within are performed using 3×3 convolutional layers and a Pixel Shuffle layer.

```
# normalise the input image
self.mean = self.mean.type_as(x)
x = (x - self.mean) * self.img_range

# shallow feature extraction
x = self.shallow_conv(x)
# deep feature extraction
x = self.deep_conv(self.forward_features(x)) + x
# final upsample and convolution layer
x = self.conv_before_upsample(x)
x = self.conv_last(self.upsample(x))

# denormalize the output image
x = x / self.img_range + self.mean

return x[:, :, :H * self.upscale, :W * self.upscale]
```

The deep feature extraction is implemented via an abstract method `build_residual_groups`, overridden in each child model. This modular design allows SwinIR and MambaIR to define their own residual group structures while sharing a common super-resolution pipeline.

7.3 SwinIR

The SwinIR model was implemented based on the design proposed by Liang et al. (2021), described in Chapter 6. It is defined in `archs/SwinIR_arch.py` and is called `SwinIRS2` to avoid name conflict with BasicSR's models. The model extends `BaseSuperResolutionModel` and implements the `build_residual_groups` method with Residual Swin Transformer Blocks (RSTBs).

Each `ResidualSwinTransformerBlock` consists of alternating `SwinTransfromerLayers`, with every other layer using shifted windows. The `ResidualSwinTransformerBlock` also performs patch embedding and unembedding, which is implemented in `archs/utils/layers.py` to allow it to be used by both SwinIR and MambaIR. The patch embedding converts the input feature map from (B, C, H, W) to (B, C, N) using reshape operations to allow attention operations over flattened windows.

The `SwinTransformerLayers` apply window-based multi-head self-attention locally in non-overlapping windows with windows shifted in every other layer to allow for cross-window context representation. They follow the structure proposed by Liang et al. (2021):

```
if self.shift_size > 0:
    shifted_x = shift(x)
else:
    shifted_x = x

x = x + self.attn(self.norm1(shifted_x))

if self.shift_size > 0:
    x = unshift(x)
else:
    x = shifted_x

x = x + self.mlp(self.norm2(x))
return x
```

The shifted window attention mechanism was adapted from HuggingFace's `timm` library(Wightman, 2019).

7.4 MambaIR

MambaIR was again implemented according to the architecture described by Guo et al. (2024). It is defined in `archs/MambaIR_arch.py` and also extends `BaseSuperResolutionModel`. Its deep feature extraction is performed by Residual State Space Groups (RSSGs), each with multiple Residual State Space Blocks (RSSBs).

Each RSSB has two states. First, the input F_i is processed by the Vision State Space Module with a residual connection:

$$Z = \text{VSSM}(\text{LN}(F_i)) + s \cdot F_i, \tag{7.1}$$

implemented as:

```
x = self.drop_path(self.vssm(self.ln1(x)))
    + self.scale1 * x
```

Then, the result Z is further refined by a bottleneck convolution and channel attention:

$$F_{i+1} = \text{CA}(\text{Conv}(\text{LN}(Z))) + s' \cdot Z \tag{7.2}$$

which is essentially implemented as:

```
x = self.ca(self.local_conv(self.ln2(x)))
    + self.scale2 * x
```

The VSSM is implemented using two branches. The first branch applies a linear projection (`nn.Linear`) followed by a depth-wise convolution (`nn.Conv2d`), SiLU activation (`nn.SiLU`), the 2D-Selective Scan Module (2D-SSM), and layer normalisation (`nn.LayerNorm`). The second branch consists of another linear expansion (`nn.Linear`) followed by a second SiLU activation (`nn.SiLU`). Each branch independently transforms the input, and results are combined via element-wise multiplication (Hadamard product).

The bottleneck convolution consists of two `Conv2d` layers with GeLU activation in between. The Channel Attention block is implemented using average pooling (`nn.AdaptiveAvgPool2d(1)`) followed by:

```
nn.Sequential(
    nn.Conv2d(...),
    nn.ReLU(),
    nn.Conv2d(...),
    nn.Sigmoid()
)
```

The 2D-SSM layer flattens the feature map in four diagonal directions and applies a state-space model recurrence equation from Equation (7.3), using weights A, B, C, D and Δ. Each direction is processed by a helper method `process_scan_direction`, and the outputs are summed to restore the original structure. `process_scan_direction` also applies the selective scan algorithm, which is imported from the original Mamba implementation(Gu and Dao, 2023).

$$\begin{aligned} \bar{A}, \bar{B} &= \exp(\Delta A), \Delta B \\ h_t &= \bar{A} h_{t-1} + \bar{B} x_t \\ y_t &= C h_t + D x_t \end{aligned} \tag{7.3}$$

7.5 ESRGAN

The ESRGAN model was adopted to include a perceptual-based GAN model for comparison. Specifically, the implementation of ESRGAN for the S2-NAIP dataset by Wolters et al. (2023) was used. Only minor changes were made to the model to make it compatible with the BasicSR configuration used.

7.6 SRResNet

The SRResNet model from the BasicSR framework was used to include a CNN-based model. Again, minor changes were made to support the CLIPScore loss and the selective saving of

in-training validation images.

7.7 Bicubic Interpolation (Baseline)

A bicubic interpolation baseline was also included. It was implemented using PyTorch's built-in interpolation function:

```
F.interpolate(self.lq, scale_factor=self.scale_factor,
    mode='bicubic', align_corners=False)
```

Like other models, it was registered with BasicSR to be accessed via the same interface for consistency.

7.8 Training and Testing

The training and evaluation processes were built on top of BasicSR's scripts, which already provide implementation of robust data loading, performance logging, model saving, and more. The following changes were made: First, `tqdm` was used to better visualise the training progress. Additionally, custom validation functionality was added to allow the setting of image saving frequency and the saving of a subset of validation images. The testing script provided by BasicSR was used without any changes.

Chapter 8

Experimental Results

8.1 Quantitative Results

Table 8.1 summarises the performance of all investigated models trained *without* the CLIPScore loss. Each model was optimised with its standard loss function: L_1 loss for SRResNet, SwinIR, and MambaIR, and $L1$ loss plus adversarial loss for ESRGAN. Bicubic interpolation is included as a classical baseline.

Model	PSNR (dB)	SSIM	CLIPScore
Bicubic	14.7	0.409	0.677
SRResNet	18.8	0.444	0.714
ESRGAN	18.0	0.346	**0.821**
SwinIR	**20.5**	0.480	0.706
MambaIR	**20.5**	**0.484**	0.714

Table 8.1: Super-resolution performance on the S2-NAIP validation set for models trained *without* CLIPScore loss. The highest values for each metric are bolded.

From these results, MamabIR achieves the highest PSNR (20.5 dB) and SSIM (0.484), marginally outperforming SwinIR and clearly outperforming SRResNet. This indicates that models with global dependencies representation, like SwinIR and MambaIR, can recover pixel-wise details more accurately than CNN-based models under the same loss.

Interestingly, despite its relatively low pixel-based scores, ESRGAN achieves the highest CLIPScore (0.821). This indicates it can produce super-resolved images most semantically similar to the ground truth. This is consistent with the trade-off between perceptual quality

and pixel-level accuracy observed in previous works(Wang et al., 2018).

8.2 Qualitative Analysis

Qualitative analysis was performed by visually inspecting the super-resoled outputs produced by each model against the ground truth NAIP images. Figure 8.1 presents example super-resolved images produced by the models.

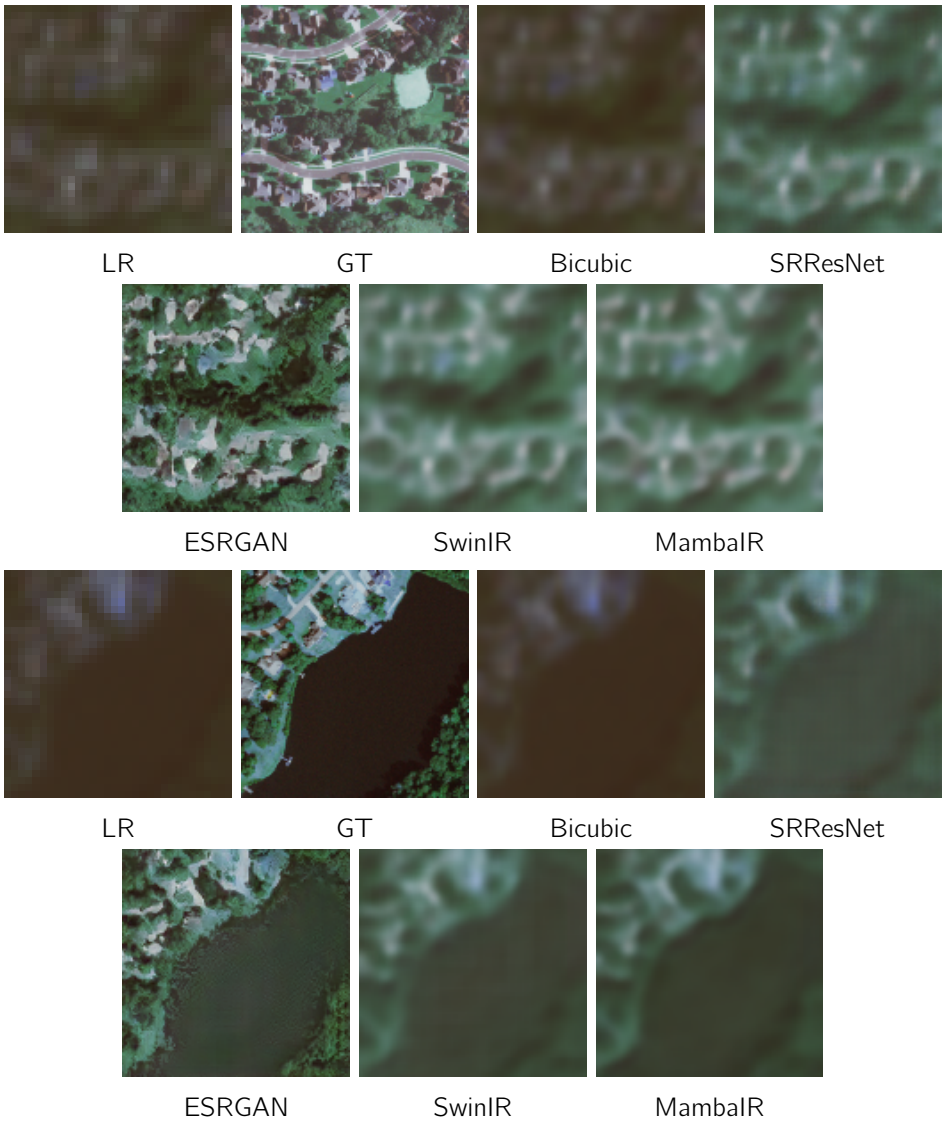


Figure 8.1: Qualitative comparison of 4× super-resolution outputs from the investigated models (trained without CLIPScore loss) on sample Sentinel-2 `small_val_set` images

SRResNet generates structurally accurate reconstructions but with significant blurring and overly smooth textures. This is typical for models optimised with a pixel-wise loss. SwinIR

and MambaIR produce slightly sharper outputs with some finer details. Importantly, all three models reconstruct prominent structures in the image, although they might miss some finer details. ESRGAN clearly produces the most visually sharp and detailed predictions. The image is not blurred, making it possible to make out finer details. However, it occasionally hallucinates high-frequency details and textures not present in the ground truth. Finally, as expected, bicubic interpolation produces a smooth and blurred image. Major features of the image are recognisable and preserved, but details cannot be resolved.

Overall, the qualitative observations correlate well with the quantitative metric findings.

8.3 Ablation Study

8.3.1 Effect of CLIPScore Loss on Model Performance

Table 8.2 shows results for models trained for 300,000 iterations *with* the CLIPScore loss added to their objective. This is compared to results for models trained for 300,000 iterations on the same data, *without* the CLIPScore loss in table 8.1.

Including this semantic perceptual loss generally led to higher CLIPScores across the board and improvements in quality and amount of details in produced images as observed in Figure 8.2. However, these improvements came at the cost of a small decrease in PSNR and SSIM scores for all models.

Model	PSNR (dB)	SSIM	CLIPScore
SRResNet + CLIP	18.6	0.390	0.795
ESRGAN + CLIP	17.6	0.322	**0.851**
SwinIR + CLIP	**20.1**	**0.411**	0.808
MambaIR + CLIP	20.0	**0.411**	0.785

Table 8.2: Super-resolution performance on the S2-NAIP validation set for models trained *with* CLIPScore loss. The highest values for each metric are bolded.

ESRGAN improved the least from the added CLIPScore loss, likely due to its adversarial training that already attempts to maximise perceptual resemblance. For the other models, the CLIPScore and image quality improvements were more obvious.

These results suggest that semantic losses like CLIPScore can be used to complement pixel-wise objectives. This is especially useful in satellite image super-resolution, where image

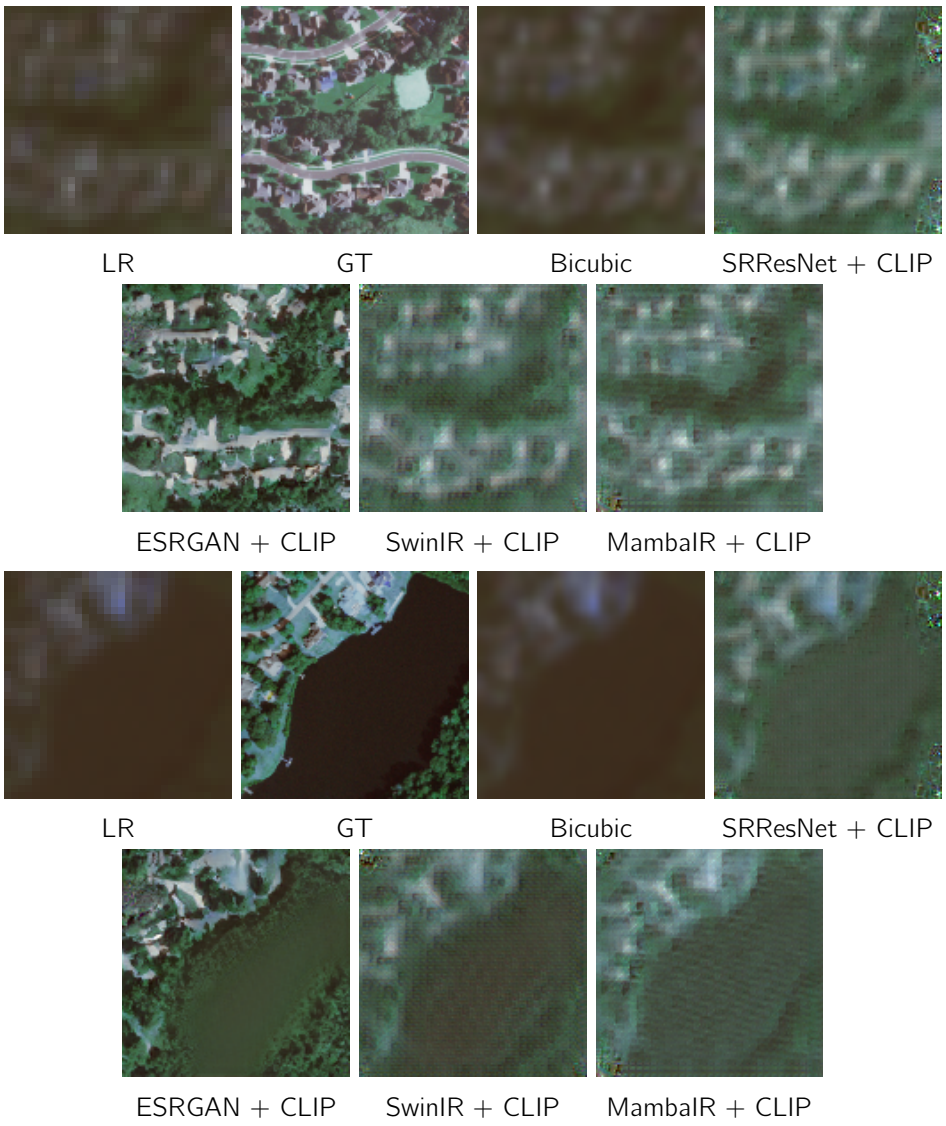


Figure 8.2: Qualitative comparison of 4× SR results of the investigated methods (trained with CLIPScore loss) on sample Sentinel-2 `small_val_set` images

quality and details are often quite important. Including a semantic loss like CLIPScore appears to help pixel-wise models produce more recognisable images with finer details. However, the GAN approach still achieves the best perceptual quality.

8.3.2 Effect of 2× Model Weight Initialisation

To assess if pre-training on an easier task improves performance on more difficult super-resolution, SwinIR and MambaIR were initialised with weights from models trained on 2× SR before fine-tuning for 4× SR. The 2× models were trained for 200,000 iterations, followed

by 100,000 iterations of fine-tuning on the 4× super-resolution. A randomly selected subset of 30,000 patch pairs from `train_urban_set` was used for pre-training, and 10,000 patch pairs were used for fine-tuning.

Model	PSNR (dB)	SSIM	CLIPScore
SwinIR (random init)	**20.5**	0.480	0.709
SwinIR (2× init)	20.2	0.476	0.706
MambaIR (random init) weights	**20.5**	**0.484**	**0.714**
MambaIR (2× init)	20.4	0.483	**0.714**

Table 8.3: Comparison of model performance with and without 2× pre-trained weight initialisation. The highest values for each metric are bolded.

Interestingly, pre-training on 2× super-resolution does not improve PSNR or SSIM for either model. In fact, the randomly initialised models slightly outperform the ones initialised with 2× weights. For the semantic CLIPScore metric, 2× pre-training does not appear to help either. SwinIR achieves better performance without 2× pre-training, and MambaIR achieves the same performance with and without 2× pre-training.

The qualitative assessment, illustrated in Figure 9.5, seems to support these findings with no apparent differences between the images produced by the model with and without 2× pre-training.

Given the extra complexity of such pre-training and no observable benefits, 2× pre-training for SwinIR and MambaIR does not seem to be worth it.

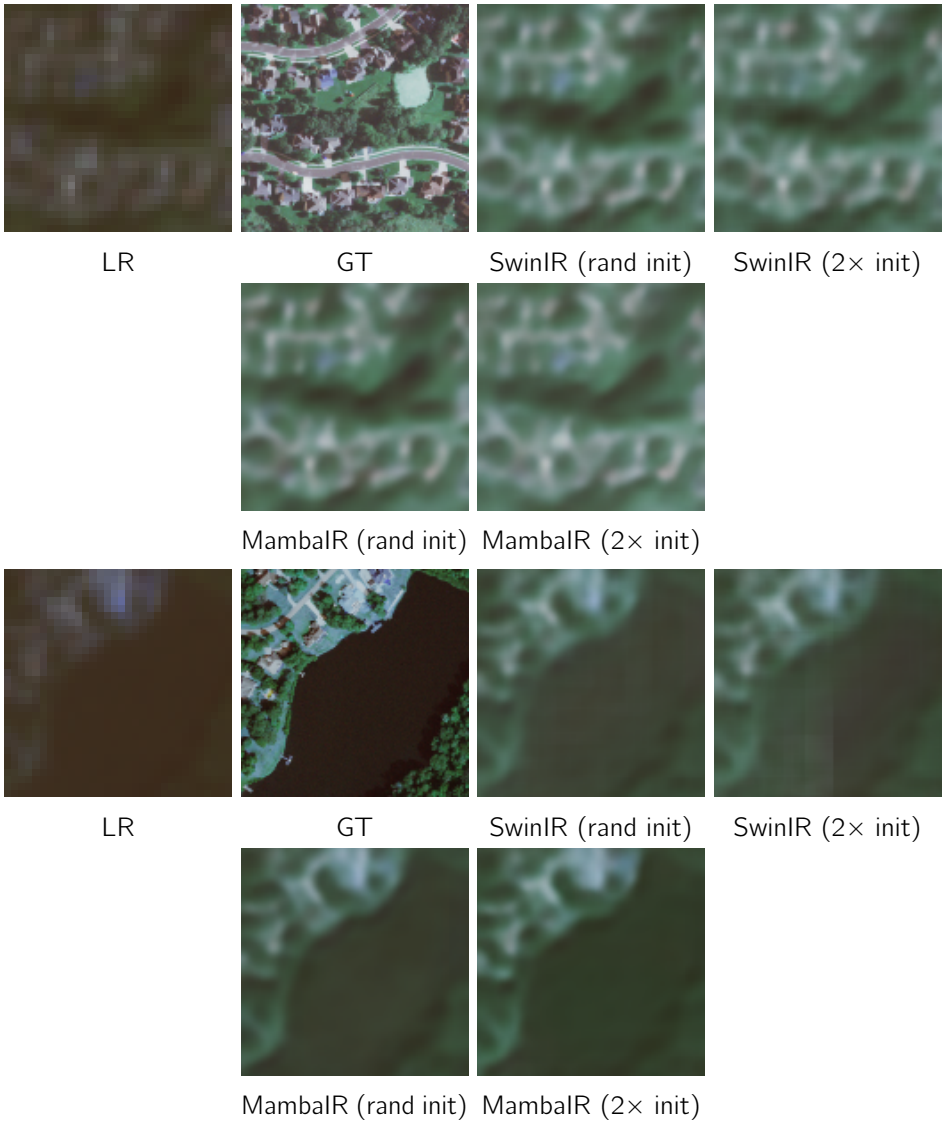


Figure 8.3: Qualitative comparison of 4× SR results of SwinIR and MambaIR with and without 2× pre-training on sample Sentinel-2 `small_val_set` images

Chapter 9

Discussion

9.1 Comparative Analysis of Model Architectures

The experimental results demonstrate that both SwinIR and MambaIR outperform the CNN-based SRResNet on pixel-wise metrics such as PSNR and SSIM. This is significant because all three models were trained on the same L_1 pixel-wise loss function, indicating that any performance differences can be attributed to their model architectures.

As discussed in Chapter 2, convolutional neural networks process images with sequential convolutional layers with inherently local receptive fields. This limits their ability to capture long-range dependencies, which are particularly important in satellite images and images in general. In contrast, SwinIR relies on the sliding window attention mechanism, and MambaIR relies on state-space models (SSMs) and the selective scan algorithm. Both approaches allow the models to effectively represent long-range global dependencies(Liu et al., 2021; Gu and Dao, 2023) and better preserve high-frequency details, typically missing in pixel-wise trained CNNs (Ledig et al., 2017).

The visual outputs, such as those in Figure 9.1, confirm these observations. SRResNet consistently produces over-smoothed reconstructions without sharp detail. While neither SwinIR nor MambaIR produces sharp images, they capture more detail and appear slightly less blurry than SRResNet. This aligns with previous works, which suggest that CNN-based methods optimised for pixel metrics often generate over-smoothed results(Ledig et al., 2017).

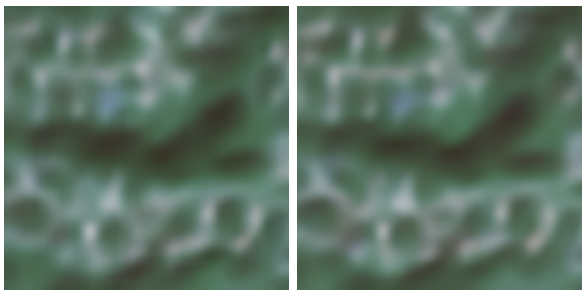


SRResNet SwinIR MambaIR

Figure 9.1: Qualitative comparison of 4× super-resolution outputs from SRResNet, SwinIR, and MambaIR (trained without CLIPScore loss) on sample Sentinel-2 `small_val_set` image

9.2 MambaIR vs. SwinIR

Another interesting finding is the very close performance of MambaIR and SwinIR. While MambaIR slightly outperforms SwinIR, the difference is only marginal, with the same PSNR and SSIM within 0.004, and visually almost imperceivable(see Figure 9.2). This suggests that MambaIR's residual state-space layers allow for at least as efficient long-range dependency modelling as SwinIR's shifted window attention. This observation is supported by Guo et al. (2024)s' observation that MambaIR achieve a larger global receptive field compared to SwinIR, allowing it to better model long-range dependencies. Therefore, while both architectures are certainly effective at capturing spatial dependencies, much more so that SRResNet and MambaIR may be slightly more promising. However, to properly establish this, additional factors need to be considered, such as memory and computational efficiency, as well as super-resolution performance with a larger scale factor.



SwinIR MambaIR

Figure 9.2: Qualitative comparison of 4× super-resolution outputs from SwinIR and MambaIR (trained without CLIPScore loss) on sample Sentinel-2 `small_val_set` image

9.3 Perceptual Quality of ESRGAN

ESRGAN, a GAN-based model, consistently achieves the highest CLIPScore among all evaluated models, reflecting its best visual quality observed. This observation seems to align with ESRGAN's architecture, as it focuses on perceptual realism over pixel accuracy(Wang et al., 2018). ESRGAN is trained with a generator learning to fool a discriminator by producing outputs that look indistinguishable from the ground truth. This adversarial training likely encourages ESRGAN to create sharp, detailed images.

In qualitative comparisons, ESRGAN's outputs have the most detailed textures and the highest quality. However, some of the high-frequency details it creates are hallucinated; they do not seem to be based on the low-quality image and do not appear in the ground truth image but look realistic(see Figure 9.3). ESRGAN's hallucinations also explain its low PSNR and SSIM scores, as these metrics penalise any pixel-level discrepancies.

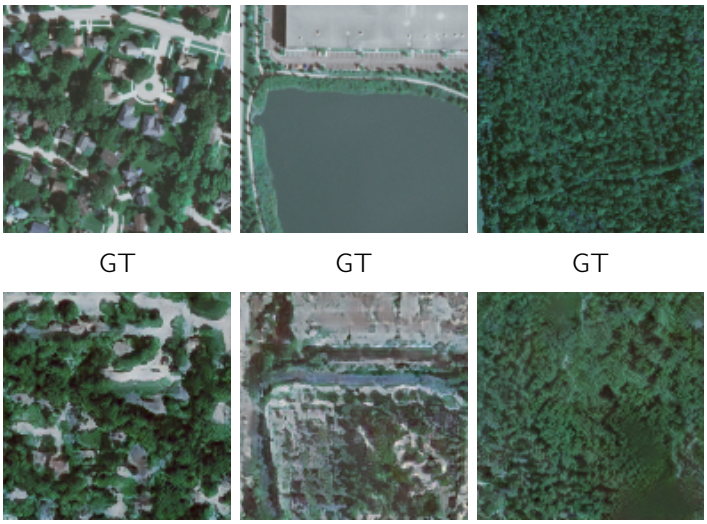


Figure 9.3: Qualitative comparison of 4× super-resolution outputs from ESRGAN (trained without CLIPScore loss) with ground truth on sample Sentinel-2 `small_val_set` image

ESRGAN's consistently high CLIPScore performance also seems to be related to the perceptual quality of its images. The CLIPScore, which is based on embeddings produced by a Vision Transformer model, appears to reward realistic-looking images with sharp details. This is supported by CLIPScore's high correlation with human preference since humans tend to prefer clear, sharp images over blurry ones(Wolters et al., 2023).

Therefore, ESRGAN could be highly effective for applications that require humans to interact with super-resolved images. However, its hallucinations make it less suitable for remote sensing use cases that require accurate image reconstruction and the preservation of existing structures.

9.4 Bicubic's SSIM Performance

A surprising observation was that bicubic interpolation achieves a higher SSIM score than ESRGAN. This observation likely stems from the nature of the SSIM, which rewards consistency in luminance, contrast and structural information but does not significantly penalise blur.

Since bicubic interpolation preserves spatial structure without introducing fake details, it achieves a decent SSIM score, even though the output is visually blurred. ESRGAN, on the other hand, adds textures and details that appear real but get penalised heavily by SSIM because they are often not consistent with the existing structure in the low-resolution image. The same trend, however, is not observed with PSNR, which evaluates pixel differences. As bicubic interpolation does not introduce any new details and instead only performs interpolation, it is likely to produce high pixel errors.

Overall, this illustrates the importance of judging model performance based on a diverse set of metrics, with ones correlating well with human preference included.

9.5 Impact of CLIPScore Loss

The ablation study found that incorporating the CLIPScore loss generally improves the perceptual quality of models trained with pixel-based losses. When trained with this additional semantic loss function, SwinIR, MambaIR, and SRResNet all produce sharper, more detailed outputs with higher CLIPScores. The improvement is especially significant for SRResNet, which otherwise struggles to produce images with much detail.

The CLIPScore loss encourages models to generate features that the pre-trained Vision Transformer (CLIP's image encoder), whose embeddings are used in the CLIPScore, perceives as similar. In practice, this means that the models are encouraged to reconstruct objects so that a large vision model (like the Vision Transformer) would recognise them as the same

object or scene. As a consequence, models trained with the CLIPScore loss tend to preserve semantic details better, resulting in sharper details and textures, and overall improved clarity of the image, as shown by Figure 9.4.

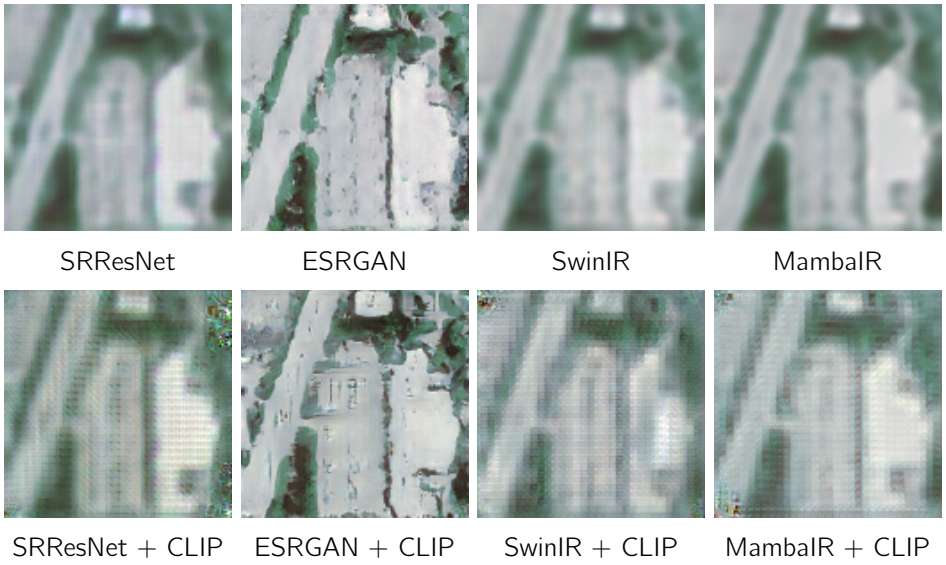


Figure 9.4: Qualitative comparison of 4× super-resolution outputs from all models with and without CLIPScore loss on sample Sentinel-2 `small_val_set` image

However, including CLIPScore comes at the cost of minor drops in PSNR and SSIM. This is, however, expected as the CLIPScore loss does not directly minimise pixel error. In fact, it might encourage some deviations from the ground truth as long as the overall semantic is improved. However, because the performance drops are small (< 0.4 dB PSNR), adding CLIPScore is probably worth it, especially when realism is important.

The only model that gains little from the addition of CLIPScore is ESRGAN. Its architecture and adversarial training already minimise perceptual loss, so the CLIPScore is likely redundant and, in some cases, might even be conflicting. These are both qualitative and quantitative results, which indicate no significant improvements.

9.6 Effect of 2× Pre-training

Another approach evaluated for improved model performance is model initialisation with weights from a 2× super-resolution model. The assumption was that a model trained on a simpler task might generalise better to a more difficult problem.

Surprisingly, this approach did not result in any improvements for either model on any of the metrics or quantitative results. This suggests that while pre-training can help encode some high-level features, it may also constrain the model from effectively learning to produce finer details required in 4× super-resolution. Further experiments with more iterations, different loss functions or weight initialisation strategies might achieve better performance, but in the conducted experiment, no benefit to 2× weight initialisation was observed.

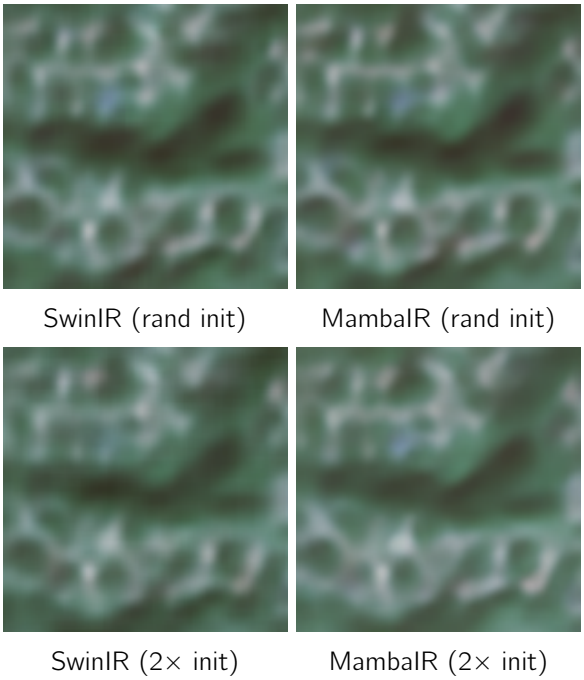


Figure 9.5: Qualitative comparison of 4× SR results of SwinIR and MambaIR with and without 2× pre-training on sample Sentinel-2 `small_val_set` images

9.7 Dataset and Domain Challenges

An important aspect that should be considered is the significant drop in performance of the evaluated SR methods compared to results of previous studies(Fu et al., 2018; Pineda et al., 2020). Unlike those studies that use low-resolution images synthetically derived by downsampling high-resolution imagery, this work uses the S2-NAIP dataset, a cross-sensor dataset combining real Sentinel-2 and NAIP images. It is, therefore, important to put into perspective the measured model performances with the dataset and domain they were applied to.

The cross-sensory nature of the used dataset presents several challenges that do not apply to synthetically generated datasets. First, the images are not captured simultaneously by the same sensor, resulting in differences in lighting, atmospheric conditions and minor misalignments, see Figure 9.6. Second, NAIP provides RGB-only images, restricting the use of Sentinel-2's multispectral bands. These differences introduce a fundamental, irreducible error between inputs and the ground truth, lowering the possible PSNR/SSIM and introducing noise into the training. This is likely part of the reason for the lower PSNR values observed in this work (around 20dB) compared to synthetic benchmarks, where values over 30 dB are not uncommon. On the other hand, this setup is more realistic and provides a better indication of the generalised performance of the models. Crucially, all models were evaluated on the same dataset with the same irreducible error, so relative comparisons remain accurate.

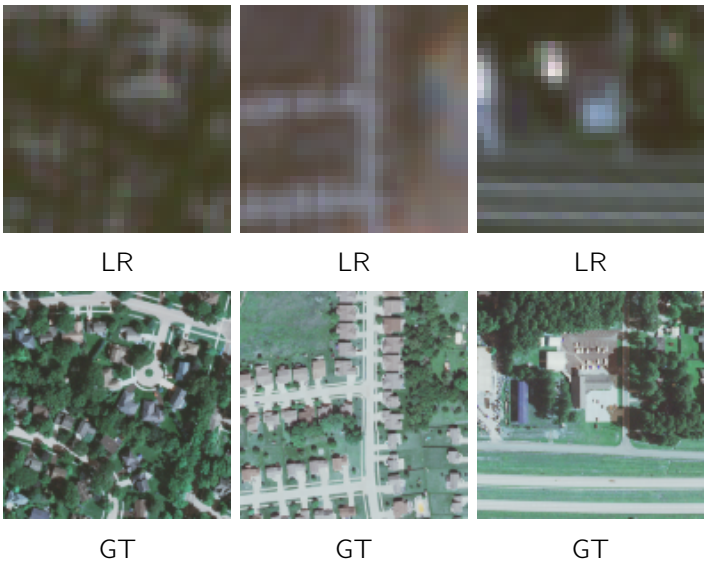


Figure 9.6: Visualisation of differences between LR and GT images in `small_val_set` including lighting, atmospheric conditions, and color scheme

To achieve better comparison with synthetic datasets, some prior works have introduced a degradation model to ensure a better match between the two sensors(Aybar et al., 2024). Therefore, future work could include using domain adaptation or harmonisation between sensors(Aybar et al., 2024) so that the task becomes closer to pure super-resolution.

9.8 Key Findings

The experiments reveal a number of interesting insights:

- The transformer-based (SwinIR) and the state-space-based (MambaIR) architectures outperform CNNs in both pixel accuracy and perceptual quality, even with the same training loss, indicating the importance of effectively representing global context.
- MambaIR achieves slightly better quantitative performance than SwinIR, suggesting state-space-based models are a viable and possibly superior alternative to attention-based models for satellite image super-resolution.
- The GAN-based model (ESRGAN) is the best at producing visually sharp, realistic images but often hallucinates high-frequency details and achieves the lowest score on pixel-based metrics.
- Bicubic interpolation performs surprisingly well on SSIM, illustrating the importance of a diverse set of evaluation metrics, including some that align well with human preference.
- Adding CLIPScore loss improves perceptual quality for all models, especially those trained with pixel-based loss. The cost of incorporating CLIPScore loss is a minor decrease in pixel-wise performance.
- Initialising a $4\times$ model with $2\times$ model weights leads to slightly lower quantitative performance, indicating no benefit to $2\times$ pre-training in its current form.
- Real-world datasets such as S2-NAIP face challenges not encountered in synthetic datasets, such as sensor gaps, imperfect ground truth, and temporal differences, leading to a relatively lower, but perhaps more realistic, performance indication.

Chapter 10

Evaluation and Critical Appraisal

10.1 Primary Objectives

The primary objectives were focused on understanding the architectures of interest, data acquisition, model implementation, training, and evaluation. All the primary objectives were completed successfully.

Although somewhat subjective, an in-depth study of the Swin Transformer and Mamba architectures was conducted, acquiring the necessary knowledge to implement them for super-resolution. The S2-NAIP dataset was also successfully acquired, consisting of low-resolution Sentine-2 images paired with high-resolution NAIP images for 4× super-resolution. Both the Swin Transformer-based SwinIR and the Mamba-based MambaIR were implemented using PyTorch and BasicSR and successfully applied to satellite image super-resolution. A robust pre-processing pipeline provided by the S2-NAIP dataset was used to pre-process data and BasicSR was used to train the implemented models. Then, to establish their performance, the implemented models were evaluated using quantitative metrics such as PSNR, SSIM, and CLIPScore. Finally, the implemented SwinIR and MambaIR were compared against bicubic interpolation, SRResNet, and ESRGAN. It was found that SwinIR and MambaIR outperform all these models on pixel-wise metrics but are surpassed by the GAN-based ESRGAN on the perceptual-oriented CLIPScore.

10.2 Secondary Objectives

The two secondary objectives focused on refining model performance and usability. The first objective was to perform an ablation study (or hyperparameter tuning) on the implemented models. The second objective was implementing a user-facing interface or API to allow easy access to the implemented model.

The first secondary objective was successfully achieved, as an ablation study of the integration of CLIPScore loss and the impact of pre-training from a $2\times$ super-resolution model was conducted, helping inform these design choices. The other secondary objective, however, was not completed within the time allocated for the project and is, therefore, an area of potential future development.

10.3 Tertiary Objectives

The tertiary objectives target long-term applicability and generalisation of the implemented models. Neither of the tertiary objectives was completed. However, both an exploration of optimisation of the models for edge devices and the exploration of the applicability of the models to other types and resolutions of satellite images represent interesting areas for future exploration.

10.4 Related Work

This project successfully implemented both a Swin Transformer-based SwinIR and Mamba-based MambaIR and confirmed their effectiveness in super-resolution satellite images. This finding generally aligns with existing literature(Liang et al., 2021; Guo et al., 2024; Qiao et al., 2024). Additionally, the performed ablation study provides details into important design decisions for these models, helping improve the understanding of MambaIR and SwinIR, particularly in the context of satellite imagery.

Chapter 11

Conclusion

This work explores whether recent advances in deep learning, particularly the transformer-based and state-space-based architectures, could improve access to affordable, high-resolution imagery through super-resolution of freely available low-resolution Sentine-2 images. Access to such imagery could benefit a wide range of scientific and non-commercial applications.

To this extent, two state-of-the-art super-resolution models, SwinIR and MambaIR, were implemented and trained on the S2-NAIP dataset, which pairs 32×32 Sentinel-2 images with 128×128 NAIP ground truth for $4\times$ super-resolution. These models were evaluated against a bicubic interpolation baseline and two established models, SRResNet and ESRGAN. Quantitative and qualitative results show that both SwinIR and MambaIR outperform the CNN-based SRResNet on all metrics and outperform the GAN-based ESRGAN on pixel-wise metrics. Additionally, it was found that MambaIR achieves marginally better performance compared to SwinIR in most cases, on most evaluation metrics.

A number of additional observations emerged from the experiments that were conducted. Including CLIPScore loss improved perceptual quality for all models, particularly those trained with a pixel-wise loss function. The cost of including CLIPScore loss appears to be a minor pixel-wise performance loss. Additionally, it was found that pre-training a $4\times$ model using weights from a $2\times$ model did not improve results, suggesting that cross-scale weight transfer does not help models learn under the experiment settings. Finally, it was found that the S2-NAIP dataset presents a more difficult challenge compared to synthetic datasets, observed by overall lower model performances; however, it possibly better simulates a real-world

super-resolution scenario.

All primary objectives set for the project were met successfully. SwinIR and MambaIR were successfully implemented, evaluated on real-world data, and compared to existing super-resolution methods. Secondary objectives were met partially. An ablation study was conducted to examine the key design choices. However, the development of the user interface was not completed in time. Tertiary objectives, such as edge deployment and evaluation of model generalisation, were not completed within the time frame of the project but are promising for future exploration.

There are several directions in which this work could be extended. One avenue could involve combining architectures such as SwinIR and MambaIR with perceptual-oriented models like the GAN-based ESRGAN in a multi-stage pipeline. Another interesting investigation could examine the impact of using temporal image sequences to improve consistency and detail. Future work could also explore domain harmonisation strategies to help mitigate cross-sensor mismatches(Aybar et al., 2024).

In conclusion, this work demonstrates the effectiveness of transformer-based and state-space-based models on real-world satellite super-resolution data. While certain limitations remain and more work needs to be done, these results suggest that super-resolving low-quality, freely available satellite images into high-resolution ones is feasible and is a promising avenue for improving access to affordable, high-quality satellite imagery.

References

Aybar, C., Montero, D., Contreras, J., Donike, S., Kalaitzis, F. and Gómez-Chova, L. (2024), 'Sen2naip: A large-scale dataset for sentinel-2 image super-resolution', *Scientific Data* **11**(1), 1389.

Dai, T., Cai, J., Zhang, Y., Xia, S.-T. and Zhang, L. (2019), Second-order attention network for single image super-resolution, *in* 'Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)'.

Dong, C., Loy, C. C., He, K. and Tang, X. (2016), 'Image super-resolution using deep convolutional networks', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **38**(2), 295–307.

Fu, J., Liu, Y. and Li, F. (2018), Single frame super resolution with convolutional neural network for remote sensing imagery, *in* 'IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium', pp. 8014–8017.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. (2014), Generative adversarial nets, *in* Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence and K. Weinberger, eds, 'Advances in Neural Information Processing Systems', Vol. 27, Curran Associates, Inc.

Gu, A. and Dao, T. (2023), 'Mamba: Linear-time sequence modeling with selective state spaces', *ArXiv* **abs/2312.00752**.

Guo, H., Li, J., Dai, T., Ouyang, Z., Ren, X. and Xia, S.-T. (2024), Mambair: A simple baseline for image restoration with state-space model, *in* 'European conference on computer vision', Springer, pp. 222–241.

He, K., Zhang, X., Ren, S. and Sun, J. (2016), Deep residual learning for image recognition, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 770–778.

Hessel, J., Holtzman, A., Forbes, M., Le Bras, R. and Choi, Y. (2021), CLIPScore: A reference-free evaluation metric for image captioning, *in* M.-F. Moens, X. Huang, L. Specia and S. W.-t. Yih, eds, 'Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing', Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, pp. 7514–7528.

Illarionova, S., Shadrin, D., Shukhratov, I., Evteeva, K., Popandopulo, G., Sotiriadi, N., Oseledets, I. and Burnaev, E. (2023), 'Benchmark for building segmentation on up-scaled sentinel-2 imagery', *Remote Sensing* **15**(9).

Johnson, J., Alahi, A. and Fei-Fei, L. (2016), Perceptual losses for real-time style transfer and super-resolution, *in* B. Leibe, J. Matas, N. Sebe and M. Welling, eds, 'Computer Vision – ECCV 2016', Springer International Publishing, Cham, pp. 694–711.

Kim, J., Lee, J. K. and Lee, K. M. (2016), Accurate image super-resolution using very deep convolutional networks, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 1646–1654.

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z. and Shi, W. (2017), Photo-realistic single image super-resolution using a generative adversarial network, *in* '2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 105–114.

Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L. and Timofte, R. (2021), Swinir: Image restoration using swin transformer, *in* 'Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops', pp. 1833–1844.

Liu, Y., Tian, Y., Zhao, Y., Yu, H., Xie, L., Wang, Y., Ye, Q., Jiao, J. and Liu, Y. (2024), 'Vmamba: Visual state space model', *Advances in neural information processing systems* **37**, 103031–103063.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S. and Guo, B. (2021), Swin transformer: Hierarchical vision transformer using shifted windows, *in* 'Proceedings of the IEEE/CVF international conference on computer vision', pp. 10012–10022.

Niu, B., Wen, W., Ren, W., Zhang, X., Yang, L., Wang, S., Zhang, K., Cao, X. and Shen, H. (2020), Single image super-resolution via a holistic attention network, *in* A. Vedaldi, H. Bischof, T. Brox and J.-M. Frahm, eds, 'Computer Vision – ECCV 2020', Springer International Publishing, Cham, pp. 191–207.

Pineda, F., Ayma, V. and Beltrán Castañón, C. (2020), 'A generative adversarial network approach for super-resolution of sentinel-2 satellite images', *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* **XLIII-B1-2020**, 9–14.

Qiao, J., Liao, J., Li, W., Zhang, Y., Guo, Y., Wen, Y., Qiu, Z., Xie, J., Hu, J. and Lin, S. (2024), 'Hi-mamba: Hierarchical mamba for efficient image super-resolution', *ArXiv* **abs/2410.10140**.

Salgueiro Romero, L., Marcello, J. and Vilaplana, V. (2020), 'Super-resolution of sentinel-2 imagery using generative adversarial networks', *Remote Sensing* **12**(15).

Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A. P., Bishop, R., Rueckert, D. and Wang, Z. (2016), 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.

Simonyan, K. and Zisserman, A. (2015), Very deep convolutional networks for large-scale image recognition, Computational and Biological Learning Society, pp. 1–14.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u. and Polosukhin, I. (2017), Attention is all you need, *in* I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan and R. Garnett, eds, 'Advances in Neural Information Processing Systems', Vol. 30, Curran Associates, Inc.

Wagner, L., Liebel, L. and Körner, M. (2019), 'Deep residual learning for single-image

super-resolution of multi-spectral satellite imagery', *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* **IV-2/W7**, 189–196.

Wang, X., Xie, L., Yu, K., Chan, K. C., Loy, C. C. and Dong, C. (2022), 'BasicSR: Open source image and video restoration toolbox', https://github.com/XPixelGroup/BasicSR.

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

Wang, Z., Cun, X., Bao, J., Zhou, W., Liu, J. and Li, H. (2022), Uformer: A general u-shaped transformer for image restoration, *in* 'Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 17683–17693.

Wightman, R. (2019), 'Pytorch image models', https://github.com/rwightman/pytorch-image-models.

Wolters, P., Bastani, F. and Kembhavi, A. (2023), 'Zooming out on zooming in: Advancing super-resolution for remote sensing'.

Yan, J., Chen, Z., Pei, Z., Lu, X. and Zheng, H. (2024), 'Mambasr: Arbitrary-scale super-resolution integrating mamba with fast fourier convolution blocks', *Mathematics* **12**(15).

Yang, W., Zhang, X., Tian, Y., Wang, W., Xue, J.-H. and Liao, Q. (2019), 'Deep learning for single image super-resolution: A brief review', *IEEE Transactions on Multimedia* **21**(12), 3106–3121.

Zabalza, M. and Bernardini, A. (2022), 'Super-resolution of sentinel-2 images using a spectral attention mechanism', *Remote Sensing* **14**(12).

Appendix A

Testing Summary

No unit or integration tests were implemented for this project. Instead, testing was performed through empirical evaluation of the super-resolution outputs produced by each model. This involved both visual inspection of reconstructed images and quantitative comparison using metrics such as PSNR, SSIM, and CLIPScore. To verify correctness and performance, these outputs were compared against baseline methods, including bicubic interpolation, SRResNet, and ESRGAN. Additional confirmation was done by comparing these outputs to results reported in related work.

Appendix B

User Manual

A brief manual for running the code developed as part of this project is provided. The implementation includes two primary super-resolution architectures, SwinIR and MambaIR, along with implementations of benchmark models (SRResNet, ESRGAN, and bicubic interpolation) for comparison. All models are implemented using PyTorch and integrated into the BasicSR framework, with custom extensions where necessary.

B.1 Repository Structure

The main components of the project are structured as follows:

- `archs/` — Contains the model architectures (SwinIR, MambaIR, SRResNet, ESRGAN).
- `models/` — Implements the BasicSR-compatible wrappers around each architecture.
- `options/` — YAML configuration files for training, testing, and evaluating each model.
- `data/` — Custom dataset loader for the S2-NAIP dataset, adapted from the Satlas repository.
- `losses/` — Custom loss functions, including CLIPScore loss.
- `metrics/` — Custom evaluation CLIPScore metric.
- `train.py` — Script for training models.
- `test.py` — Script for evaluating trained models.

- `visualise.py` — Script for generating visual comparisons between models.

B.2 Installation and Setup

To run the codebase, first install the required dependencies. These can be installed manually using `pip` with the provided `requirements.txt`, or through Docker using the setup in `DOCKER-README.md`.

Option 1: Python Virtual Environment

```
python -m venv .venv
source .venv/bin/activate
pip install -r requirements.txt
```

Option 2: Docker (Recommended)

For reproducibility and easier GPU deployment, a Docker setup is provided. Refer to `DOCKER-README.md` for detailed instructions.

B.3 Running the Project

All commands should be run from the `src/`.

B.4 Training a Model

To train a super-resolution model, run the training script with the corresponding configuration file from the `options/` directory. For example, to train the MambaIR model:

```
python train.py -opt options/mambaIR.yml
```

Training logs and model checkpoints will be saved to the directory specified in the configuration file by `experiments_root` in `path`.

B.5 Evaluating a Model

To evaluate a trained model on the test set, use the same configuration file with the `test.py` script:

```
python test.py -opt options/mambaIR.yml
```

Evaluation results will be printed to the console and optionally saved as logs.

B.6 Visualising Model Outputs

For side-by-side visual comparisons of multiple models, use the `visualise.py` script. The script takes a list of model configuration files, an output directory path, and the number of test images to evaluate:

```
python visualise.py --models  \
    options/bicubic.yml \
    options/mambaIR.yml \
    options/swinIR.yml \
    options/srresnet.yml \
    options/esrgan_s2naip.yml \
    --output ./base_model_comparisons \
    --max_images 20
```

This will visualise model performance over the specified test samples.

B.7 Dataset

The project currently supports the S2-NAIP dataset only. The dataset loader is adapted from the Satlas repository and expects a directory of Sentinel-2 (32×32) images and a directory of NAIP (128×128). Paths and preprocessing options are set in the configuration files.

B.8 Modifying Configuration Files

All training and evaluation behaviour is specified via the YAML configuration files. These files specify model type, dataset paths, batch sizes, number of iterations, loss functions, logging behaviour, etc.