Satellite Image Cloud Removal:

Learning Within and Beyond the Sample

Mikolaj Czerkawski

CIDCOM Group

Electronic and Electrical Engineering

University of Strathclyde, Glasgow

2023

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

Abstract

Earth observation technologies constitute a powerful set of tools for understanding the systems and ongoing processes that occur on Earth. The technology largely focuses on satellite imaging, which relies on observation from remote positions, away from the planet's surface. As a consequence, a large portion of the satellite imagery in the optical spectrum is hindered by the presence of clouds in the atmosphere, which obstruct a clear view of the ground. While it is difficult to prevent this issue at the acquisition stage, it may be possible to approach the problem from a data-based perspective by processing images affected by clouds. More precisely, the cloud removal technology aims to approximate the features of the ground obscured by the clouds present in a given image. Given the established power and versatility of Deep Learning methods for image synthesis problems, the recent solutions to the cloud removal problem are primarily focused on deep neural networks. Still, state-of-the-art techniques are limited in at least several aspects; they often cannot easily adapt to new signal representations, or easily ingest new types of guidance signals. Trained on limited datasets, they are generally run under the risk of overfitting. Furthermore, the evaluation of these models is often performed on non-ideal validation data, where the cloud-free ground truth is often divergent from the theoretical ground truth corresponding to a given cloudy image.

This work explores several themes related to these limitations and proposes solutions to overcome some of them. Several novel methods for performing cloud removal or satellite image inpainting are proposed, most of them operating in an internal learning setting, where no dataset-based training is performed. These methods either rely on the existing information in the inference sample or priors captured by models trained on

different tasks, such as vision-language models. The key advantage of these techniques is their flexibility and ability to adjust to diverse data scenarios, with different numbers of channels and guidance signals.

The related problem of evaluating cloud removal solutions and training on reliable data is also explored, and consequently, a novel framework of SatelliteCloudGenerator for simulating clouds and shadows in optical multi-spectral images is proposed. The key advantage of the approach is a high degree of control over the features of the generated clouds, based on a set of adjustable parameters. The quality of the simulated images is further demonstrated by applying models trained exclusively on simulated data to real images.

Finally, the question of the benefits of the proposed internal learning and language-based techniques, compared to an externally trained model, is treated by testing these approaches on a common dataset with both historical (Sentinel-2) and cross-sensor (Sentinel-1) guidance. It is found that a performance gap remains between the internal learning and language-based methods when compared to externally trained solutions, despite a promising level of performance.

Contents

List of Figures

List of Tables

Acronyms

CLIP Contrastive Language–Image Pre-training. viii, 6, 7, 10, 69–74, 78, 79, 81, 83, 99

CNN convolutional neural network. 17, 19, 33, 34, 43–45, 67

CSM channel-specific magnitude. 115

DDIM denoising diffusion implicit model. 28, 31

DDPM denoising diffusion probabilistic model. 28, 31

DIP Deep Image Prior. 34, 150, 152

GAN generative adversarial network. 19, 23, 25, 26, 28, 29, 31

HSI hyperspectral imaging. 14

MAE mean absolute error. 37, 40

MCPN multi-modal convolutional parameterisation network. 49, 50, 52, 56–65, 67, 150, 154

MSE mean square error. 37, 38, 50

MSI multi-spectral imaging. 14, 18, 33, 70, 73, 83, 91, 93, 96, 99, 151, 152

PSNR peak signal-to-noise ratio. 38, 40

RGB red green blue. 14, 15, 19, 22, 33, 48, 73, 137, 150–153, 158

RMSE root mean square error. 37, 40, 52, 53, 55, 56, 59, 65, 67

SAM spectral angle mapper. 40

SAR synthetic-aperture radar. 15, 42, 48, 55, 57, 68, 75, 77

SIFID single image Fréchet inception distance. 53

SSIM structural similarity index. 38–40, 52, 55–61, 65, 131, 151–153

SWIR short wave infrared. 130

VAE variational autoencoder. 25, 26, 28, 31

Acknowledgements

This document culminates a rather beautiful experience of my PhD journey. While it does not currently feel as blissful, at the very end of the writing process and so close to having it submitted, the process of pursuing a doctoral degree was an absolute highlight of my life thus far. The freedom to explore my own interests and directly benefit from developing ideas is right in the centre of my comfort zone.

For most of my young life, I would not consider research an environment where I belong. This would have to be changed by Dr Christos Tachtatzis, my current supervisor, who offered me project opportunities during my undergraduate years and gradually helped me steer towards a doctoral degree. Among the many things I can thank Christos for, the most fundamental one is his belief and confidence in my competence, which at the beginning of our relationship was still under development. His drive and supervision have made the completion of this degree smoother than I could anticipate.

Besides that, the most important person who made this work possible was my life partner, Gabriela Kulesza. It is difficult to express the amount of support she has given to me over the years of our relationship and importantly, during the course of this work. My motivation, my mental health, my drive, my excitement, my creativity, all of these have been nurtured largely by interacting with her. I love you, Gaga, and I want to thank you for making all of this worthwhile.

Despite the pandemic that occurred several months after starting this degree, I was not going through it in isolation. For some of my achievements, I have to thank my wider environment. Most imporantly, I owe a lot to my friends and my colleagues, Tasos, Tom, Priti, Sebas, Laura, James, Javier, Nati, Maciek, Magda, Marek, and

everyone else I have had the pleasure to know during my time at Strathclyde. My environment was not merely limited to people, but also places. I wish to pay some gratitude to Scotland and the city of Glasgow. Having moved in 2015 for my undergraduate studies, by 2023 I felt completely at home and at peace in Glasgow. The beauty of Glasgow is difficult to summarise in one sentence, so I can only point to my favourite elements of the experience, such as the freshest air I could think of, the beautiful parks for walking, the sandstone tenement buildings, the pubs, the canals, the cosy subway line, the cottonrake bakery, and the smell of rain.

Finally, to extend beyond Glasgow and Scotland, I want to reach further and thank my family, my parents, Kasia and Wojtek, my two brothers, Kuba and Maciek, and my aunt, Agata, for being the great family to me that they are.

Chapter 1

Introduction

Figure 1.1: *The Blue Marble*, one of the first non-stitched photographs of fully-illuminated Earth, demonstrates the wide-spread effect of clouds on satellite imagery.

On their way to the Moon, Apollo 17 crew had a unique opportunity to capture one of the most widely known satellite images of the Earth, *The Blue Marble* [1] (Figure 1.1). It was not the first time a human has taken a photograph of the entire Earth from its side (that would be the *Earthrise* taken from the lunar orbit during the Apollo 8 mission [2]), so there should be more reasons for the high popularity and wide recognition of the image [3]. The official name of the photograph might be a good hint. The comparison to a marble has been made because, unlike in *Earthrise*, *The Blue Marble* shows Earth

illuminated by sun with no visible shadow, allowing to observe the entire scope of one of the planet's sides. *The Blue Marble* gave an opportunity for humans to look at a true, non-stitched photograph of the Earth and perceive it in detail, as a whole, integrated object.

However, upon inspection, one can realise that the blue marble is not really blue, despite the name. There is quite a lot of dark blue ocean present, but this colour is not exclusive. There is also some green and orange, indicating land, and most importantly, quite a lot of white. This white component, widely known as clouds and ever-present in Earth's atmosphere, is the key subject in this work. No matter how much data is captured, how frequently, and how high the resolution is, the clouds are very likely to obscure a large part of all remote optical measurements of Earth.

This thesis proposes that both statements of *"Planet Earth is blue / And there's nothing I can do"*, made at least as early as 1969 ("Space Oddity" by David Bowie), are very much incorrect in their literal sense. It is argued that (i) Earth is not only blue, often due to cloud cover, and (ii) there is something that can be done about that, at least in the domain of image synthesis methods.

Coincidentally, the year of the Apollo 17 mission, 1972, links not only to the birth of *The Blue Marble*, but also to the launch of the first Landsat satellite dedicated to the sensing of environment [1], as part of the NASA program aimed at capturing satellite imagery of Earth. Landsat-1 (originally ERTS-A) was the first satellite to carry a multi-spectral scanner, which allowed to capture observations of the Earth's surface, such as the one pictured in Figure 1.2. The program continues to run at the time when this text is being written (Landsat-9 is the latest operating satellite), making it one of the most profound and long-term examples of human effort to monitor Earth from space. As later described, the amount of imagery captured this way is rapidly growing. Yet, the challenge of understanding the observations distorted by clouds remains.

This challenge can be addressed in many ways. The techniques and tools developed within the fields of machine learning and computer vision are particularly appropriate and powerful. However, their power is directly dependent on the data used for the learning process. Hence, the development of effective methods for processing or re-

Figure 1.2: An image of San Francisco Bay Area captured by Landsat 1 Satellite [4].

moving clouds from satellite imagery demands a good understanding of the value that diverse and high-quality data sources can bring.

This work aims to explore a variety of sources and learning contexts. The learning context can be i) internal (for example, the high prevalence of the dark blue colour in the *Blue Marble* sample should give a good hint of what the likely colours are in some areas covered by clouds), ii) external, where the solution is based on information extracted from other samples. Furthermore, in the external context, it is often possible to transfer knowledge from other tasks, which could be treated as a separate context of its own. In terms of the data sources used for removing clouds, a plethora of potential configurations exists with the available sources of imagery. In this work, it is recognised that the use of historical optical images and recent radar acquisitions (which can penetrate the clouds) can considerably improve the quality of the cloud removal tools.

1.1 Problem Definition

In this introductory chapter, it is crucial to precisely define the problem addressed in this work. Cloud removal refers to the task of synthesizing a satellite image without the presence of clouds based on an input image that could contain clouds. In other words, the aim is to produce an output image that precisely corresponds to the input image under cloud-free conditions.

The motivation for a cloud removal tool links to a large number of remote sensing applications, such as agricultural or disaster monitoring. Simply put, the presence of clouds can be problematic for applications where an image from a specific time must be acquired, or where a region must be observed over an extended period of time (known as satellite image time series).

1.2 Motivation: Next Generation of Cloud Removal Technology

What follows is a description of some of the desired features in a technology tasked to remove clouds from satellite imagery.

- **No Requirement for Supplied Cloud Mask.** The advancement of cloud detection algorithms is still in progress, and a good cloud removal system will not be dependent on the selection of a specific cloud mask source. Instead, the knowledge should ideally be shared implicitly between the cloud detection and cloud removal task. Furthermore, this allows the system to be readily applied by the end user.

- **Good Preservation of Cloud-Free Pixels.** In many images, there will be some areas that contain minimal or no cloud obstructions. The property of preserving this information and reconstructing it in the generated sample is not guaranteed by any algorithm and it is partially dependent on the implicit or explicit cloud detection mechanism. An ideal system will be able to identify these

regions precisely and make sure that they are faithfully reproduced in the generated sample.

- **Generalisation Across**:
  - **Time (Conditions).** The cloud removal model should be able to remove the clouds in a variety of new conditions.
  - **Space.** The model should also be capable of performing well over varying regions of the Earth.

  In both cases, it may likely be infeasible to retrain the model over all acquired data and the volume of the acquired data is still merely a portion of the entire spatial and temporal scope of potential Earth observation. Hence, the model must be able to generalise beyond the limited volume of data available during its production.

- **Adaptation to Sensing Modalities.** There is a large number of different sensing modalities affected by clouds to various extents, coming from various sources and instruments. Ideally, a cloud removal model will not need to be retrained for every new type of sensor. This reduction in cost does require solving two key challenges: i) different sensing modalities may require a model with a different number of image channels, and ii) if the image is severely dominated by cloud obstruction, the limited information about the ground properties will likely need to be supported by other sources (such as historical data). Just like in the case of different channels, it would also be beneficial to offer flexibility of selecting the supporting information at inference time, based on what is available to the user.

The features listed above require solutions to several technical challenges. Given the ease of learning directly from data owing the advances in the field of deep learning, there is a common tendency to rely on deep neural networks for this type of computer vision problems. This work is no exception, where several fundamentally different types of learning are explored, all of which can be classified as deep learning.

1.3 Summary of Contributions

This thesis explores several distinct paths for applying deep learning for removing clouds from satellite images. It proposes solutions for internal learning (within the sample) and learning from language (beyond the sample). It also introduces a new model for simulating presence of clouds in satellite images and demonstrates the utility of using it for training and evaluation of models trained on pair-based datasets. Finally, several techniques from different modes of learning are compared on a common evaluation dataset.

- Three types of internal learning approaches to convolutional parameterisation for satellite image generative tasks are proposed and tested on satellite image inpainting and super-resolution (Chapter 3)
- It is shown that a pre-trained CLIP model can be used for detecting cloud presence in a zero-shot manner and can transfer well between different data sources of Sentinel-2 and Landsat-8 (Chapter 4)
- A pipeline for StableDiffusion inpainting with ControlNet guide is introduced and used for conditional inpainting of satellite images (based on historical data) (Chapter 4)
- A method for filling additional channels of multi-spectral images based on an RGB inpainting is proposed and tested (Chapter 4)
- A new model for simulating cloud presence in satellite images is introduced (Chapter 5)
- It is demonstrated that deep learning models trained solely on simulated data can achieve good performance on real data for the task of cloud and shadow detection and cloud removal (Chapter 5)
- A new convolutional neural network architecture capable of ingesting both SAR and optical support data has been proposed and trained (Chapter 6)

- A common evaluation dataset has been designed to test the problem setting where a radar image as well as a historical optical image are available (Chapter 6)
- A comparison between different types of learning has been performed on the common evaluation dataset (Chapter 6)

1.4 Author's Publications

The following is the list of peer-reviewed publications that the Author has contributed to during the course of his PhD degree.

Published

- **M. Czerkawski** et al., *"Deep Internal Learning for Inpainting of Cloud-Affected Regions in Satellite Imagery,"* Remote Sensing, vol. 14, no. 6, p. 1342, Mar. 2022, doi: 10.3390/rs14061342.
- **M. Czerkawski**, R. Atkinson, and C. Tachtatzis, *"Detecting Cloud Presence in Satellite Images using the RGB-based CLIP Vision-Language Model,"* International Geoscience and Remote Sensing Symposium (IGARSS 2023), Pasadena, US, 2022

In Review

- **M. Czerkawski** et al., *"Neural Knitworks: Patched Neural Implicit Representation Networks"* - in review, Elsevier Pattern Recognition
- **M. Czerkawski** et al., *"Multi-Modal Convolutional Parameterisation Network for Guided Image Inverse Problems"* - in review, Springer Applied Intelligence
- **M. Czerkawski** and C. Tachtatzis, *"SatelliteCloudGenerator: Parameterized Cloud and Shadow Synthesis in Optical Satellite Imagery"* - in review, MDPI Remote Sensing

- **M. Czerkawski** and C. Tachtatzis, *"Exploring the Capability of Text-to-Image Diffusion Models with Structural Edge Guidance for Multi-Spectral Satellite Image Inpainting"* - in review, IEEE Geoscience and Remote Sensing Letters

Other

- **M. Czerkawski**, C. Clemente, C. Michie and C. Tachtatzis, *"On input formats for radar Micro-Doppler signature processing by convolutional neural networks,"* International Conference on Radar Systems (RADAR 2022), Hybrid Conference, Edinburgh, UK, 2022, pp. 383-388, doi: 10.1049/icp.2022.2348.

- **M. Czerkawski**, C. Clemente, C. Michie, I. Andonovic and C. Tachtatzis, *"Robustness of Deep Neural Networks for Micro-Doppler Radar Classification,"* 2022 23rd International Radar Symposium (IRS), Gdansk, Poland, 2022, pp. 480-485, doi: 10.23919/IRS54158.2022.9905017.

- P. Upadhyay, **M. Czerkawski** et al., *"A Flexible Multi-Temporal and Multi-Modal Framework for Sentinel-1 and Sentinel-2 Analysis Ready Data,"* Remote Sensing, vol. 14, no. 5, p. 1120, Feb. 2022, doi: 10.3390/rs14051120. [Online]. Available: http://dx.doi.org/10.3390/rs14051120

- D. Pavlovic, **M. Czerkawski** et al., *"Behavioural Classification of Cattle Using Neck-Mounted Accelerometer-Equipped Collars,"* Sensors, vol. 22, no. 6, p. 2323, Mar. 2022, doi: 10.3390/s22062323.

- **M. Czerkawski**, J. Cardona, R. Atkinson, C. Michie, I. Andonovic, C. Clemente, C. Tachtatzis, *"Neural Weight Step Video Compression,"* NeurIPS Workshop on Pre-Registration Science 2021

- **M. Czerkawski** et al., *"Non-invasive Diver Respiration Rate Monitoring in Hyperbaric Lifeboat Environments using Short-Range Radar,"* OCEANS 2021: San Diego – Porto, San Diego, CA, USA, 2021, pp. 1-5.

- **M. Czerkawski**, C. Ilioudis, C. Clemente, C. Michie, I. Andonovic and C. Tachtatzis, *"A Novel Micro-Doppler Coherence Loss for Deep Learning Radar Ap-*

plications," 2021 18th European Radar Conference (EuRAD), London, United Kingdom, 2022, pp. 305-308, doi: 10.23919/EuRAD50154.2022.9784491.

- **M. Czerkawski**, C. Ilioudis, C. Clemente, C. Michie, I. Andonovic and C. Tachtatzis, *"Interference Motion Removal for Doppler Radar Vital Sign Detection Using Variational Encoder-Decoder Neural Network,"* 2021 IEEE Radar Conference (RadarConf21), Atlanta, GA, USA, 2021, pp. 1-6, doi: 10.1109/RadarConf2147009.2021.9454986.

- **M. Czerkawski**, C. Ilioudis, C. Clemente, C. Michie, I. Andonovic and C. Tachtatzis, *"On Models and Approaches for Human Vital Signs Extraction from Short Range Radar Signals,"* 2020 14th European Conference on Antennas and Propagation (EuCAP), Copenhagen, Denmark, 2020, pp. 1-5, doi: 10.23919/EuCAP48036.2020.9135189.

1.5 Thesis Outline

This work is organised with the following structure:

- **Chapter 2 (Background)** contains an overview of the research conducted to date on the relevant topics of remote sensing, computer vision, and image generation.

- **Chapter 3 (Internal Learning)** introduces several techniques capable of satellite image inpainting task based on the input sample. The experiments test the performance achieved by each technique based on several types of supporting data (such as historical and SAR). These approaches require no large-scale training and can adapt to any number of spectral channels. It is also demonstrated how a similar approach can be used for other tasks, such as satellite image super-resolution or guided image inpainting.

- **Chapter 4 (Learning from Language)** explores the suitability of the representations learned by large language-based vision models for satellite image processing. Several methods of employing the CLIP model for classifying cloud-affected images are proposed and evaluated. Further, a technique for performing image inpainting with the open-source StableDiffusion text-to-image model is shown. The StableDiffusion model operates on RGB data, so a subsequent step of transferring the information from the RGB inpainting output to more channels is also proposed and tested.

- **Chapter 5 (Simulation)** describes the features and internal design of a satellite cloud generator tool. The tool has been designed to provide high-quality synthetic data for training and evaluating deep learning models for cloud removal and cloud detection. To demonstrate the quality of the synthesised data, networks are trained from scratch on real data and simulated data to enable a comparison. It is shown that models trained solely on synthetic data can achieve good performance on real data.

- **Chapter 6 (Comparison)** describes the creation of a common evaluation dataset using the cloud simulation tool and compares several methods using the same test images. Specifically, the internal learning method of MCPN is compared against the StableDiffusion approach, and finally, an externally trained network is proposed for the purpose of this work. The external approach is an extension of the DSen2-CR approach, developed to allow conditioning on both radar and historical images.

- **Chapter 7 (Conclusion)** contains a summary of findings delivered in this work and a description of the wider context and impact of these findings.

Chapter 2

Background

Given the several overlapping themes present in this work, the background chapter is divided into sections with specific focus topics. It begins with the general topic of remote sensing of Earth, followed by a section on the phenomenon of clouds in the atmosphere, and after that, more technical topics of cloud detection, removal, and image synthesis.

2.1 Remote Sensing of the Earth

Remote Sensing is a broad term coined in the 1960s [5] for a technique of extracting information about environment without immediate physical contact between the measuring device and the observed scene [1]. In the context of environmental sensing (which is what the term remote sensing often implicitly refers to), the information-carrying medium is most often some type of electromagnetic radiation. Depending on whether the sensing instrument transmits electromagnetic waves to enable imaging, it can be classified as either active or passive measurement.

The history of remote sensing of the Earth was determined by two key inventions, one related to seeing and one related to flying. Both correspond to the two key components necessary for carrying out a remote sensing observation, an instrument capable of acquiring images must be placed in a position with a suitable view, which often happens to be in the air. Consequently, some of the first known attempts at remote

sensing occurred shortly after the first successful inventions enabling humans to fly objects and take photographs. Early examples of that include several successful attempts of taking photographs from balloons made in the second half of the 19th century [1], or the invention of mounting cameras to carrier pigeons [5]. These techniques could already provide a new kind of perspective for understanding the environment, however, it was usually limited to acquisitions from sparse paths taken by the sensing instrument. In the case of pigeon photography, this path was not completely predictable and the camera angles could be random.

As it turns out, what really enabled remote sensing of Earth with a wide coverage with high-quality captures was the use of Earth-orbiting satellites, a technology conceived initially by Herman Potočnik in *The Problem of Space Travel - The Rocket Motor* as early as 1929 [6] and realised for the first time with the successful launch of Soviet Sputnik I satellite in 1957 [5]. This was soon followed by the first full-scale weather satellite launched in 1960 by NASA, equipped with a wide-angle TV camera [5]. The next important invention was to replace the camera with line radiometers, capable of scanning the Earth's surface line by line to form a larger image. In the following decades, a large number of Earth observation satellites have been launched by various organisations. A notable example of that is the Landsat program, started by NASA in 1972 and running continuously until today, with the last launch of Landsat 9 having occurred on 27 September 2021 [7].

At the time of writing this work in 2023, the Committee on Earth Observation Satellites (CEOS) reports 183 active missions, with another 157 at various stages of preparation [8]. Some of the sources often featured in the scientific literature [7] are shown in Table 2.1. The table summarises the modality of each instrument as well as the data availability. Despite the high number of sources, access to relevant Earth observation data is not straightforward. Satellite image products are often commercialised and sold to provide financial support to the programmes. Even if free, some sources can only be accessed after successful approval of a project proposal. The two most prominent exceptions to this are the Sentinel and Landsat programmes, run by the European Space Agency (ESA) and NASA, which offer open and free access to

Table 2.1: Selected Source for Remote Earth Observation.

Source	Modality	Free Open Access
ISRO - Cartosat	RGB+NIR	✗
ISRO - INSAT-3DR	Mulitspectral	✗
ASI - COSMO-SkyMed	SAR	✗
CSA - Radarsat	SAR	✓
DLR - TerraSAR	SAR	✗
CMA - Fengyun	Mulitspectral	✓
CNSA - Gaofen-2	RGB+NIR	✗
CNSA - Gaofen-3	SAR	✗
NASA - Landsat-8/9	Mulitspectral	✓
NASA - MODIS	Mulitspectral	✓
NASA - ASTER	Mulitspectral	✓
ESA - Sentinel-1	SAR	✓
ESA - Sentinel-2	Mulitspectral	✓
ESA - Sentinel-3	Mulitspectral	✓
EUMETSAT - MetOp	Mulitspectral	✓
Commercial		
Airbus - SPOT6/7	red green blue (RGB)+NIR	✗
Airbus - Pleiades	RGB+NIR	✗
Airbus - Pleiades Neo	RGB*+NIR	✗
Maxar - WorldView 2	Mulitspectral	✗
Maxar - WorldView 3	Mulitspectral	✗
Planet - SkySat	RGB+NIR	✗
Planet - Planetscope	RGB*+NIR	✗
ICEYE	SAR	✗

several sources of satellite Earth observation data.

The majority of the commonly used Earth observation devices rely on the propagation of electromagnetic waves through the space between the sensor and the Earth's surface. To this day, the most common method of measuring the environment is via the optical medium, by taking photographs. This can involve a conventional type of camera sensitive to the three colours perceived by humans, or imaging in different bands of the electromagnetic spectrum. multi-spectral imaging (MSI) is a sensing technique involving the acquisition of multiple optical bands each with a relatively wide bandwidth. The number of bands in a MSI sensor will usually be in the order of 10 or 20 channels. This is different from hyperspectral imaging (HSI), where hundreds of channels with much narrower bandwidths are acquired to better approximate the interaction of the

scene with a diverse set of carrier frequencies.

The main motivation for acquiring images beyond the visible bands is to observe different types of phenomena in the scene, not apparent (or maybe difficult to detect) in the RGB bands, for example, aerosols or temperature changes.

Another popular type of sensing is synthetic-aperture radar (SAR), which is a fundamentally different technology from optical sensing. Instead of detecting the electromagnetic energy emitted or reflected from the Earth, SAR relies on transmitting its own electromagnetic carrier signal and processing the reflections returned from the ground. The control over the transmitted carrier signal allows for several sensing modes that are generally not possible with passive optical sensing, such as polarimetric or interferometric modes. The polarimetric SAR relies on measuring the polarisation of the returned carrier wave, which is affected by the scattering properties of the reflecting objects. The interferometric SAR relies on the comparison of the phase of the reflected component from multiple acquisitions, which enables the detection of subtle distance shifts in the scene. This wide range of functionality, combined with the fact that the radar signals can penetrate clouds and do not depend on the sunlight make SAR an attractive source of information besides optical data. However, it is important to acknowledge that SAR images are generally harder to interpret due to the speckle noise and side-looking sensing direction.

2.2 Phenomenon of Clouds

Just like the water on Earth is responsible for the blue components in the *Blue Marble* (Figure 1.1 in Chapter 1), it is also responsible for the white clouds obscuring some portions of the ground surface. This is because atmospheric clouds are inherently a water-based phenomenon and occur when liquid or frozen particles of water vapour are suspended in the air.

Air containing water vapour can form a cloud (in other words, achieve saturation), by either increasing moisture level or by cooling. Cooling can be achieved by either transferring heat outside or via adiabatic expansion (with no heat transfer). Furthermore, for clouds to form, the air must contain aerosols, which provide a surface for

water condensation. These are known as cloud condensation nuclei [9].

Given this mechanism of cloud formation, it can be assumed that clouds are an integral and unavoidable phenomenon on a planet with a substantial content of surface water and the type of atmosphere and temperature range that Earth happens to have.

Cloud physics is a wide and currently explored topic on its own [10], where the behaviour of clouds in the atmosphere can be used as a source for understanding weather and climate. In the scope of this work, clouds are primarily treated as obstacles in the context of observing ground surface from space. For that reason, their physical appearance may be of more interest than the processes contributing to their presence. Recognised cloud types can be grouped into high-altitude clouds (cirrus type), medium-altitude (alto type), and low-altitude. Another important feature is their approximate orientation, with cumuliform clouds being vertically developed (along the path of elevated air) and stratiform clouds horizontally developed. The temperature of the cloud is also an important factor (the temperature is generally inversely proportional to altitude), because warmer clouds, containing mostly water droplets, tend to have more defined edges than cold clouds containing ice crystals. The cold cloud edges are more stretched due to the longer transition time between ice and vapour (as opposed to liquid and vapour) [9].

In the cloud detection and cloud removal literature, the clouds are often classified into thin clouds and thick clouds [11]. These do not necessarily have a rigorous definition and are generally discriminated based on whether the cloud appears semi-transparent, resulting in some part of the ground image being perceivable.

2.3 Satellite Cloud Detection

In many cases, the simple approach to avoid the issue of the cloud presence in satellite image analysis is to filter them out and exclude from analysis, and that, in turn, requires the cloud-affected portions of the image to be identified. The practice of classifying whether a pixel in a satellite image is affected by clouds is commonly known as the cloud detection task. Note that this definition is not entirely consistent with what detection

means in the field of computer vision and instead resembles image segmentation.

The space of existing cloud detection solutions can be divided into knowledge-driven and data-driven categories [11], based on whether the detection function is designed via hand-engineered features by humans (knowledge-driven) or learned from some volume of data. Similarly to other areas of machine learning, feature engineering can lead to a good level of generalisation, but often at the expense of certain pitfalls, such as the frequent omission of thin clouds [11].

The knowledge-driven approaches were the first type of solution to be explored, as early as 1988 [12] on AVHRR data, followed by other works on MODIS data [13], POLDER data [14], or SPOT data [15]. For the types of satellites with a constant viewing angle, multi-temporal detection methods have been explored too, such as MAJA [16]. Another important challenge addressed was to design algorithms capable of detecting clouds without access to a thermal acquisition band [16, 17] (which has often been an important part of the input data capable of differentiating between colder clouds and warmer ground surface).

For the two most popular sensor sources, cloud masks are often now provided as part of the product, including Fmask [18] for Landsat, and Sen2Cor [19, 20] for Sentinel-2. Both of these methods are rule-based. Another popular choice for Sentinel-2 data is s2cloudless [21], which is based on gradient boosting.

The more recent approaches to cloud detection largely focus on the deep learning approaches, most commonly developing solutions based on the convolutional neural network architecture, such as the early use of LeNet [22, 23],convolutional neural networks (CNNs) with superpixel preprocessing [24], or CNNs applied to small patch input [25]. Further advancements include the application of deep pyramind network s [26], PCANet [27], multi-scale convolutional feature fusion [28], UNet architecture [29, 30], transfer learning [31], or MobileNetV2 [11]. However, in many cases, these trained networks will only work for one specific sensor that they were designed for, which motivated some research on cross-sensor domain adaptation [32].

The high number of developed solutions to cloud detection and the fast advancement of the field also sparked some interest in benchmarking techniques compatible with the

same sensor, such as the validation of Sentinel-2 cloud masks [33, 34], Landsat [35], or both [36, 37].

2.4 Satellite Cloud Removal

The task of cloud removal involves translating an image affected by the presence of clouds into a corresponding equivalent with no clouds present. The first mentions of the possibility of cloud removal from satellite images have been published as early as 1977 with the short paper Mitchell [38] describing a method based on linear filtering and estimation of noise statistics. The approach relied on the assumptions of the clouds being semi-transparent, the cloud-free region being darker than the clouds, and the cloud shape being of lower (spatial) frequency. The manuscript contains several visual examples computed on Landsat MSI data, but other than that, the evaluation of the method was rather limited. Other approaches followed in the next decades, often focused on the compositing approaches [39–42]. A slightly different approach was proposed in [43], where regression trees were used to predict cloud-free pixel values in the regions affected by clouds and cloud shadows. Following that, more techniques relying on predictive models were introduced, such as linear predictor ensembles and Support Vector Machines [44]. Another approach was based on comparing the image to the statistics extracted from an artificial cloud prototype [45]. Further developments include uses of geometric flow and bandlet transform to facilitate inpainting of masked area [46]. The inpainting approach (where the mask is assumed to be available) [47] was also explored for other types of missing regions, such as sensor faults [18, 48, 49]. Still, many approaches in the following years would still assume that the cloud-affected regions can be replaced with the content from images taken at different times [50, 51]. Dictionary learning [52] and homomorphic filtering were explored [53] around the same time. An early proposal to use a SAR image as a source of guidance for the cloud removal task was introduced in [54]. In general, the methods operating on a single image would assume that the clouds were not opaque and that a substantial reflection from the ground is passing through the cloud [38, 53, 55, 56]. Another technique for

inpainting proposed in [57] was based on matrix completion.

Starting from 2016, a rise in interest in deep learning occurred, with a heavy focus on the generative modelling method, and more specifically generative adversarial networks (GANs). This began with [58] proposing McGAN tasked to transform RGB+NIR images into cloud-free RGB and a cloud mask. The model was trained on synthetic cloud images computed by alpha blending a Perlin noise sample with a cloud-free image. This approach required matched pairs of cloudy and cloud-free images, which motivated the use of synthetic data. To address this limitation, the CycleGAN approach was adapted for the cloud removal task in [59]. The simulated cloud approach was still explored in some later works. This included the techniques incorporating SAR data into the input of the generator network [60, 61] (also related was the concurrent work on the translation of SAR images to optical using conditional GANs [7, 62]) or the cloudy image arithmetic, where a real cloud would be extracted from a cloudy image and mixed into a cloud-free image to generate a realistic cloudy image with matched cloud-free ground truth [63].

In some of the other works, datasets for training GANs were created by pairing cloud-free images with temporally proximate cloudy images of the same region [64–71]. Another common theme was to start incorporating multi-temporal data into the GAN input [64, 71], or multi-source (most often SAR) [71, 72].

It should also be noted that not all of the approaches rely on a purely GAN-based framework. Another popular solution is to use CNNs without adversarial losses [35, 73–78].

Finally, less conventional cloud removal techniques based on deep learning have been proposed, including deep spatio-temporal prior combined with low-rank tensor singular value decomposition [79], internal learning on image sequences [80], or a transformer-based architecture of Former-CR [81]. It is generally uncommon in the last few years to see publications on cloud removal that do not make use of deep learning. One exception of that is [82], where cloud removal of cloud trajectory is proposed that does not involve neural networks.

The contributions made in this work are motivated by exploring different learning

strategies for cloud removal with deep neural networks, such as learning from within the sample or learning from language-based models. Not only does it propose several new approaches to the task, but it also compares them to the more conventional approach involving learning from datasets. Finally, since both training and evaluation of these models highly depend on the quality of data, a novel framework for simulating realistic clouds in satellite images is proposed with experiments demonstrating the utility of this data source for both training and validating cloud removal solutions.

2.5 Image Synthesis Techniques

The task of cloud removal can be interpreted as an example of a wider class of problems within the theme of image synthesis. This theme is more general and correspondingly, the size of the literature published on the topics is far larger. Since a large portion of modern cloud removal solutions are inspired by image synthesis techniques proposed within the computer vision literature, the following section provides a summary of the progress to date achieved within this area.

Image synthesis is a complex process that can take many forms, such as unconditional synthesis, image translation, or image completion. As a learning task, all these different forms can be interpreted as a task of approximating a probability distribution $q(x)$ of some data source and then sampling new images $x \sim q(x)$ from that distribution. In the case of conditional variants, such as image translation or image completion, the distribution $q(x)$ is conditioned on a separate input variable c, yielding $q(x|c)$. So far, the key advances in image synthesis have been achieved mostly by optimizing powerful parametric models over a relatively large volume of images acquired from $q(x)$. Hence, the main bulk of technical advances has occurred after the rise of deep learning in 2012 [83], which generally combines deep neural networks, parallel computing platforms based on GPUs, and large datasets.

Before AlexNet

Some research on image synthesis was conducted before the era of large-scale deep learning began, but it was still rather far from generating high-resolution images.

The early approaches would often focus on the simpler problem of texture synthesis, where good quality results could be achieved by applying specialised algorithms. Examples of early texture include texture synthesis by non-parametric sampling [84] or image quilting for texture synthesis and transfer [85]. In non-parametric sampling, a texture is modelled using a Markov Random Field, where the probability distribution of a given pixel colour is derived from the discrete distribution of the nearest neighbours of the patch surrounding the pixel.

Another related work of image quilting [85] introduced a method for stitching existing patches from an observed texture and harmonizing the overlapping regions by finding minimum error boundary cuts in an algorithmic fashion. This technique can be used to generate new images based on a single source image. Without any explicit type of learning, the method is able to synthesise non-trivial samples of the observed texture. A related use of the technique is texture transfer, where the patches extracted from the source texture are arranged in space based on a correspondence map (for example luminance) derived from a source image to inject information about image structure.

The domain of texture is usually interpreted as the domain of images with highly repeatable patterns, where the local patches have little dependence on their position in the scope of the full image. This means that the methods of texture synthesis are generally applicable in a narrow portion of the wider image domain and will be inappropriate for many complex image generative problems. Yet, texture-based methods can generate images with relatively good visual quality, as demonstrated in [85].

In approximately the same period, more work focused on other generative tasks has been published too. An example of that is the image inpainting algorithm published in [86], which generates the inpainted region based on the information in the neighbouring regions by connecting isophote lines (lines of equal brightness). This rather simple constraint is able to generate visually-pleasing inpaintings as demonstrated in [86]. Other notable approaches have managed to complete larger regions of the image, such

as fragment-based image completion [87] or image completion with structure propagation [88]. In the case of fragment-based completion, the method is capable of composing the inpainting based on existing fragments using adaptive neighbourhood.

Another relevant line of work from the period prior to the deep learning era focuses on the techniques of mixing large portions of existing images, a fundamentally different approach to image synthesis. An early example of that was introduced as content-based image synthesis [89], where the user could specify a mask and the desired inpainting content. The technique would then search a database with a carefully designed vocabulary and combine the images using an extension of the method of image quilting [85]. Semantic photo synthesis [90] made further progress by allowing the user to draw multiple regions and link them to specific desired labels (and also relying on a database of 16,000 images rather than 50). The power of larger sets of images was explored further in the work on scene completion using a million photographs [91]. It is one of the first works, where the information used for inpainting is extracted from the domain beyond the inpainted image, even though no learning is involved. Instead, it elevates the existing technique of gist scene descriptors [92] and finds matches in the database using a simple nearest neighbour approach. This is followed by local context matching based on colour error and then blending. As a result, not only are the scenes inpainted with high-quality real content but also multiple solutions can be found based on the same input.

At least one more noteworthy contribution was the work on deep belief nets [93], where an early application of generating new samples from the MNIST dataset was presented. It was one of the earliest approaches to the problem of image generation based on a neural network. Yet, the MNIST dataset of handwritten digits represents a rather narrow domain, since it contains 10-digit classes, with each sample being a centred greyscale image of 28 by 28 pixels. For comparison, the state-of-the-art generators of today, such as StableDiffusion [94], will generate high-quality RGB images of 512 by 512 pixels with complex structural content from a wide domain of natural images, often specified in a zero-shot manner using text prompts.

After AlexNet

The publication of the AlexNet paper [83] proved to be a turning point in computer vision research. Deep neural networks have been explored before, but this was the first time large-scale data (ImageNet classification [95]), relatively deep neural network models, and high computing capacity were combined to deliver a solution far superior to the state of the art at that time. The volume and diversity of training data combined with large models are now the key feature of the state-of-the-art solutions for a wide range of tasks, including image classification, image segmentation, image synthesis, or even outside of the visual domain, such as natural language processing. Furthermore, the frameworks, such as CUDA, capable of computing output and the internal gradients of deep neural network models in a highly parallelised fashion considerably decrease the training time.

The deep image classifiers were the first big success of deep learning applied for processing visual data. The early works largely relied on convolutional neural network architectures, but there were certain tasks that the classifier architecture was not suitable for. The classifiers would be designed to compress the information present in the image to a vector of much lower dimensionality. In the case of AlexNet, an input image of 224×224×3 (height, width, and 3 colour channels) would be converted to a vector with 1,000 of dimensions corresponding to the classes of the ImageNet challenge. In the works on deep representations in [96–98] the architectures did not take the spatial context into account and modelled simple grayscale images (digits and faces) of low resolution as a flattened vector (for example, MNIST digits with a 28-pixel border was modelled as a 784-feature vector). Yet, in the problem of image generation, the output of the deep neural network is an image, which will generally be a relatively large tensor containing a specific arrangement of pixels in space so the flattened representation can be problematic for high-resolution data with colour channels.

This, among other reasons, motivated the introduction of the convolutional architecture used for GANs [99], one of the first works where this type of generation of colour images was demonstrated on the large-scale dataset of CIFAR-10. GANs were designed as a set of two models, one tasked to generate samples (generator) and the

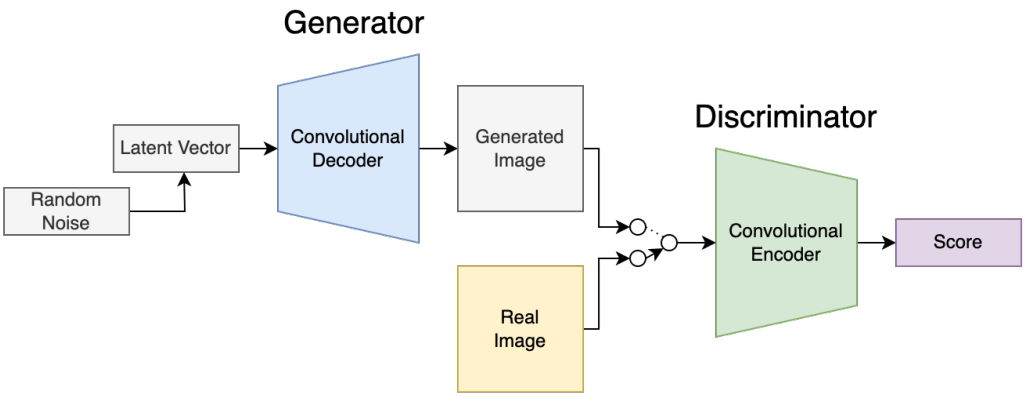


Figure 2.1: Diagram of the Generative Adversarial Network architecture

other tasked to differentiate between real and synthetic samples (discriminator), as shown in Figure 2.1.

The generator and discriminator were trained as two agents competing against each other. The discriminator consists of a convolutional encoder, similar to a classifier architecture, and computing a score approximating the likelihood of the input sample, the value of which should be high for real samples x and low for samples x' outside of the domain. This is done by minimizing the following discriminator loss $\mathcal{L}_D$, where $D()$ indicates the output of the discriminator:

$$\mathcal{L}_D = \log D(x) + \log(1 - D(x') \tag{2.1}$$

Since the samples $G(z)$ generated by the generator $G()$ with latent noise z are not real, the discriminator should assign low confidence to them, so the loss is in fact:

$$\mathcal{L}_D = \log D(x) + \log(1 - D(G(z)) \qquad z \sim \mathcal{N}(0, 1) \tag{2.2}$$

On the other hand, the objective of the generator is to generate samples resembling real samples, which should confuse the discriminator by minimizing the following loss $\mathcal{L}_G$:

$$\mathcal{L}_G = \log(1 - D(G(z)) \qquad z \sim \mathcal{N}(0, 1) \tag{2.3}$$

With these losses, both networks are trained jointly in a minimax setting, where the usual approach is to repeatedly switch between the parameter updates of the generator and the discriminator based on their individual losses. In the seminal paper on GANs by Goodfellow [99], successful generations of colour low-resolution images based on CIFAR-10 dataset [100] were achieved.

Another relatively popular generative approach introduced around the same period is the variational autoencoder (VAE) [98]. In this case, the convolutional encoder is used to encode a real image x from the observed distribution into a latent code $E(x)$ (generally, of lower dimensionality), which can then be fed through the decoder $D()$ module to reconstruct the exact same image, as shown in Figure 2.2. Minimizing the autoencoder reconstruction loss $\mathcal{L}_{\text{AE}}$ alone can allow to learn a compressed latent space that describes the observed data distribution.

$$\mathcal{L}_{\text{AE}} = d(x, E(D(x))) \tag{2.4}$$

Note that the distance metric $d()$ could be selected based on the designer's preference as long as it can represent the error between the two samples well.

However, the reconstruction alone is not sufficient for approximating the data distribution and sampling from it. Presumably, new samples could be generated by the decoder, but at this point, there is no mechanism employed for finding latent codes that correspond to samples within the distribution. To allow this, the latent space is modelled as a Gaussian distribution with mean and variance parameters generated by the encoder. This is shown in Figure 2.2. In order to allow sampling to inference, Kullback-Leibler divergence loss $\mathcal{L}_{\text{KL}}$ is minimised, which measures a distance between the latent distribution and a zero-mean unit-variance Gaussian during training. This means that upon inference, a random code from $\mathcal{N}(0, 1)$ can be generated and decoded into a new sample.

$$\mathcal{L}_{\text{AE}} = d(x, E(D(x))) \tag{2.5}$$

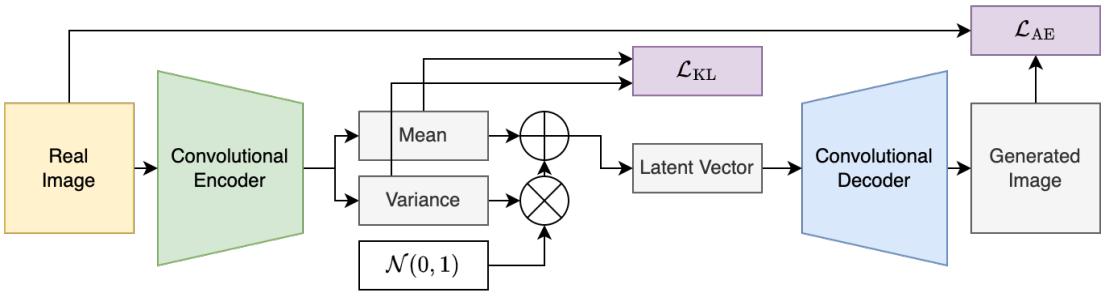


Figure 2.2: Diagram of the Variational Autoencoder Architecture

Given a latent code z, a Gaussian normal prior $p(z)$ and the distribution $q(z|x)$ parameterised by the encoder output $E(x)$, the $\mathcal{L}_{\text{KL}}$ loss is computed as:

$$\mathcal{L}_{\text{KL}} = \log q(z|x) - \log p(z) \tag{2.6}$$

The total loss $\mathcal{L}_{\text{VAE}}$ optimised during the training of a VAE is equal to

$$\mathcal{L}_{\text{VAE}} = \mathcal{L}_{\text{AE}} + \lambda \mathcal{L}_{\text{KL}}, \tag{2.7}$$

with a λ hyperparameter controlling the ratio between the reconstruction loss and the divergence loss. Unlike GAN, VAE is trained with a single non-competing loss, which is generally easier to train than a two-agent setting.

Coincidentally, not long after the introduction of GANs and VAEs, the seminal paper on the denoising diffusion models has been published [101]. However, this type of generative approach only became widely popular in 2020, when many more papers on the topic were published with greatly improved results [102–105].

However, the main principle of denoising diffusion has not changed, and it is based on using a deep neural network to reverse a forward degradation process, most commonly formulated as a Gaussian process. The forward degradation process q is defined based on a time step t, which indicates the number of degradation steps taken away from the original distribution at $t = 0$:

$$q(x_t|x_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t}x_{t-1}, \beta_t I) \tag{2.8}$$

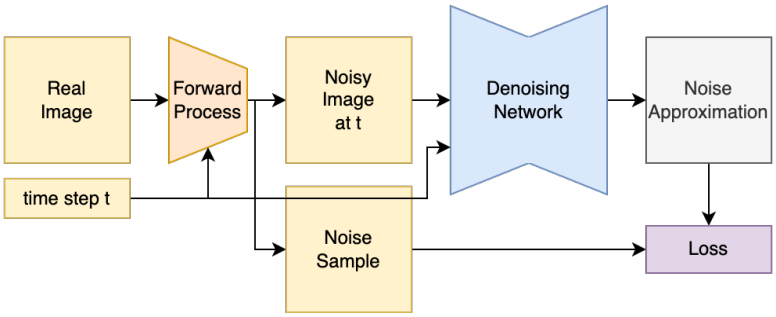


Figure 2.3: Diagram of training a denoising diffusion process.

This means that an image sample at a step t comes from a Gaussian distribution with a mean based on the image x_{t-1} previous step $t-1$ in the forward process and a specific variance β_t (which is defined through the variance schedule, part of the hyperparameter configuration). After the total number of steps T (also dependent on the β_t schedule), the forward process distribution approaches a pure normal Gaussian distribution. The reverse process of transforming the normal Gaussian to the data distribution constitutes a generative model.

Also, it can be shown [103] that for an initial sample x_0 (no degradation applied) a state from any time step t of the Gaussian forward process can be derived with $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=0}^{t} a_i$ as:

$$q(x_t|x_{t-1}) = \mathcal{N}(\sqrt{\bar{\alpha}_t}x_0, 1 - \bar{\alpha}_t I) \tag{2.9}$$

This shortcut allows to compute degraded samples at any time step t in a single step.

As shown in Figure 2.3 the model can be trained to predict the noise present in a given sample x_t computed based on a clean sample x_0 and a time step t (generally sampled from a uniform distribution).

The resulting loss for a denoising diffusion model is as simple as the distance $d()$ (usually ℓ_1 or ℓ_2 norm of difference) between the predicted noise sample $\hat{epsilon}$ and

the true noise sample ϵ:

$$\mathcal{L}_{\text{Diffusion}} = d(\hat{\epsilon}, \epsilon) \tag{2.10}$$

This makes it easier to train than a GAN (since there are no competing losses), but generally much more expensive to train and sample from.

A model capable of accurate prediction of ϵ can be used to step through the complete chain of the reverse process, starting with a sample drawn from a normal Gaussian at stage $t = T$ and finishing with an approximation of a real sample at $t = 0$. This process is known as sampling and many types of samplers have been proposed to date [102–105]. The most conventional sampler is known as denoising diffusion probabilistic model (DDPM) [103] and it simply attempts to reverse every step of the operation by approximating a Gaussian distribution of the previous step x_{t-1}:

$$x_{t-1} \sim \mathcal{N}(\tilde{\mu}_\theta, \sigma_t^2 I) \tag{2.11}$$

In DDPM, the variance σ_t of the reverse distribution is assumed to be independent of the sample x_t and set to $\sigma_t^2 = \tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$ [103], while the mean is approximated using the $\hat{\epsilon}$ prediction (which is directly computed by the neural network).

$$\tilde{\mu}_\theta = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\hat{\epsilon}) \tag{2.12}$$

This means that a diffusion model trained with $T = 1,000$ diffusion steps (a common number) will require 1,000 sequential forward passes through the neural network if DDPM sampling is used, making it about 1,000 times slower than a GAN or VAE model of comparable size. Later proposed samplers, such as denoising diffusion implicit model (DDIM) [102] aim to reduce the number of network evaluations to reduce this cost and improve sample quality, however, the inference cost of diffusion-based generative models can be expected to always be higher than GANs and VAEs.

In the few years that followed the introduction of these three generative frameworks, GANs were considered to deliver optimal quality of synthesis (while diffusion models were largely unexplored for several years after their introduction) and as a result, it was more commonly used for subsequent developments in the area of image genera-

tion [106, 107]. Radford et al. introduced a deep convolutional GAN (DCGAN) [108], where latent vector arithmetic was demonstrated for the first time. The arithmetic involved simple manipulations of latent codes of the images in order to achieve specific semantic changes in the generated output. For example, by averaging the latent vector corresponding to three images with a shared property (such as glasses), these averaged codes could be added and subtracted to achieve meaningful output corresponding to specific combinations of the relevant concepts. In the example, the mean latent vector of a 'man without glasses' was subtracted from the vector of 'man with glasses' (to approximate the 'with glasses' property) and then added to the vector of 'woman without glasses', resulting in images of women wearing glasses.

Not long after the introduction of GANs as a method for generative modelling, conditional GANs (cGANs) were explored by adding some type of condition to the generator input. This eventually lead to the application of GANs for image-to-image translation, for example, the pix2pix network [109], where the Generator would take an image from one domain as the input and output another domain as the output, as shown in Figure 2.4. It was also found that the inclusion of a random latent vector was not a necessary component and in pix2pix, this component was omitted. The approach relied on paired image examples, such as images with their segmentation maps. It was also shown that with the presence of paired data, a pixel-wise loss, such as difference magnitude, was beneficial and was hence added to the standard loss of the generator.

In a related line of work to pix2pix, CycleGAN was proposed to lift the requirement for paired image data, which was quite restrictive [110]. CycleGAN made it possible to train image-to-image translation without matched image pairs by using two generator networks, one for each direction of change between the two domains, as shown in Figure 2.5. The diagram in Figure 2.5 illustrates a cycle from domain A to domain B and back to A, all of which could be learned without a matched sample from domain B. This relies on another loss (apart from the adversarial loss propagated through the domain-specific discriminator), which is computed between the input image and the image derived by passing through both generators in a cycle. A corresponding approach would be followed for an input image from domain B, with the Generator A

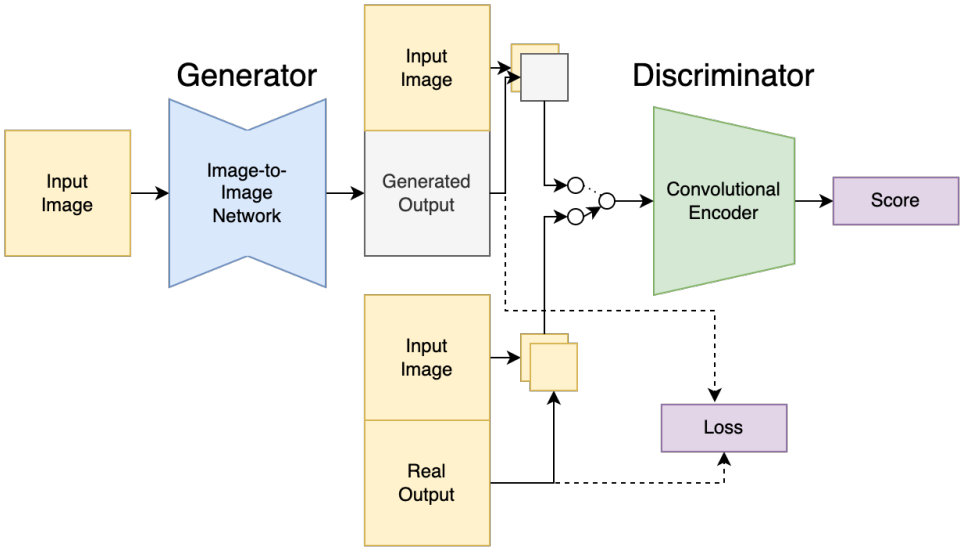


Figure 2.4: Diagram of the pix2pix network [109] capable of image-to-image translation.

being used first, and the second discriminator for domain A used for the score.

With CycleGAN, it became possible to translate between domains where matched images would be difficult or impossible, such as images of horses to images of zebras, or paintings to real photographs.

Another considerable improvement in the quality of generated photographs was enabled by the introduction of StyleGAN architecture [111]. In this case, the main difference was the design of the generator network, where the latent code was used to modulate the internal features of the convolutional network via adaptive instance normalisation. It was also found that the input latent vector was not necessary anymore and it was replaced by a learned constant vector. The architecture increased the fidelity of results and latent factors with higher disentanglement (control over semantically meaningful features).

In the follow-up work on StyleGAN2, the adaptive instance normalisation mechanism was simplified and redesigned as the modulation of the convolutional weights and furthermore, more regularisation mechanisms were proposed [112]. This resulted in further improvements in the synthesised image quality. Finally, StyleGAN3 [113] developed this architecture further by enforcing translation and rotation equivariance,

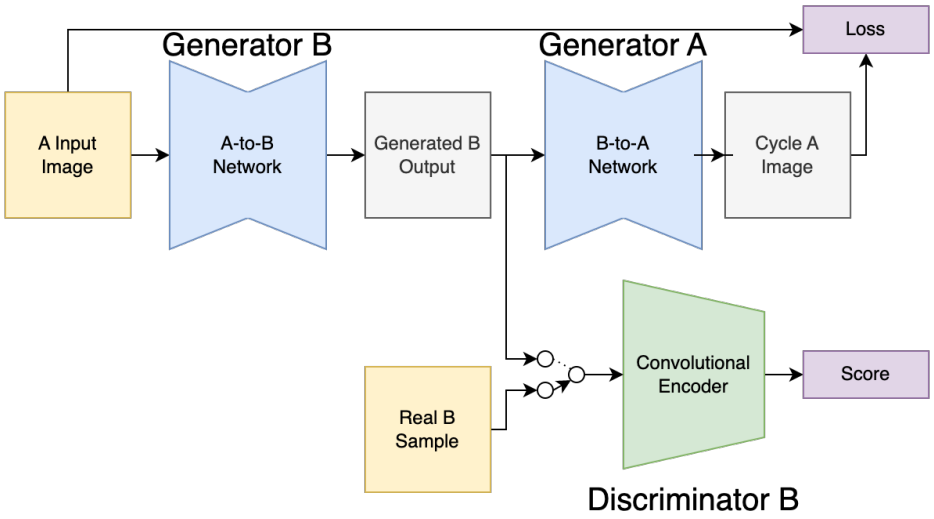


Figure 2.5: Example of the CycleGAN training for a single input sample from domain A. The same two generators (in reverse order) would be used for an input sample from domain B, along with another discriminator (for domain A, instead of B).

which nullified the texture-sticking artefacts.

Another big advancement in the area of image synthesis occurred around 2020 when the denoising diffusion was further explored in literature, in works such as DDPM [103], DDIM [102], or [105]. One considerable obstacle related to the wider use of these models was the cost of training and inference, which were significantly higher than for GAN equivalents. This topic is now heavily researched, and one of the major breakthroughs was the introduction of latent diffusion models [94], which involved the same generative process but applied to the compressed latent representation of a pre-trained image autoencoder (such as one based on VQGAN [114]).

The use of latent diffusion models has proven to be particularly useful in the domain of text-to-image generation. For the first few years of the rise of deep generative modelling (following the introduction of GANs and VAEs), the majority of models were trained to generate one specific type of data, such as faces (StyleGAN [111–113]), horses (CycleGAN [110]), or building facades (pix2pix [109]). These were indeed much richer domains than the previously explored datasets of MNIST of CIFAR-10, however, they were still very focused on a narrow theme. However, in order for a more universal

image generator to be useful, some type of conditioning mechanism for controlling the synthesis would be beneficial. Consequently, the written text, a natural medium for humans to express their thoughts, was explored as a conditioning mechanism in some early works such as [115]. However, in recent years, the quality of such solutions has significantly increased, with a prominent example of DALLE by OpenAI [116] achieving considerably higher quality compared to earlier works. DALLE was designed as an auto-regressive model, inspired by the robustness of the transformer architecture [117], and essentially treated the text-to-image generation equivalently to sequence generation (for which the transformer architecture is suitable), by compressing an image to 1024 compressed tokens, treated in the same way as text tokens. This approach was enough to generate samples based on text description with an unprecedented level of quality and flexibility. The autoregressive approach based on sequence generation has ignited wider interest in the topic of text-to-image generation, but the subsequent research has mostly moved on to diffusion-based generators, with contributions such as GLIDE [104], DALLE-2 [118], Imagen [119], or StableDiffusion [94]. The majority of these models have achieved the generation of high-resolution images in a zero-shot manner by conditioning on text input. The last one, StableDiffusion, has had a unique impact on the research community and industry as it was the first model with open-source and available trained parameters for anyone to use, which has triggered the currently ongoing developments.

The conditional image generation (such as image-to-image translation explored in the works of CycleGAN [110] or pix2pix [109]) can be used to solve a variety of common image inverse problems (problems where only the observed sample is a deteriorated or limited view of the true image), such as image inpainting or super-resolution.

Consequently, a number of works used similar approaches for the task of image inpainting, including context encoders [120], local and global discriminators [121], partial convolutions [122], contextural attention [123], residual aggregation [124], recurrent feature reasoning [65], comodulated GANs [125], Fourier convolutions [126], or region-wise operations [127]. More recently, the diffusion models have also been designed for this task, which either rely on an adjusted sampling algorithm as in RePaint [128], or

architectural change as in Palette [129]. The recent developments, such as LaMA [126] can perform image inpainting on large images with visually convincing completion.

Similarly, super-resolution has been explored in many works, such as super-resolving GAN (SRGAN) [130], enhanced SRGAN (ESRGAN) [131], pixel-recursive super-resolution [132], facial priors for face super-resolution (FSRNET) [133], latent space exploration (PULSE) [134], or diffusion with iterative refinement (SR3) [135].

Deep Internal Learning

The practice of machine learning is often categorised as either supervised or unsupervised. The supervised learning mode refers to the conditions, where the labels, or more generally, the desired network output, is available and used within the optimised objective. Alternatively, unsupervised learning covers the techniques attempting to learn representations from a set of data, without any task-specific labels. However, both supervised or unsupervised learning generally assumes that a large set of training samples is available. This scenario is referred to as external learning.

Another, arguably less mainstream, area of image synthesis research explores the potential of learning from the single test input sample alone rather than extracting the priors from an external dataset. This is known as deep internal learning [136], a technique involving deep learning applied within the context of a single sample. Several potential advantages of internal learning can be identified. First, there is no risk of dataset bias as such, since no dataset is involved (instead, the specific method itself embodies a certain bias). Second, a lot of internal learning techniques can be easily adjusted for different signal modalities, for example, non-RGB images, which can be important for certain domains, such as MSI satellite imagery. Given these benefits, internal learning is explored as a potential solution in this work.

An early example of employing deep neural networks in an internal learning regime and using the term of deep internal learning was published in 2018 [136] in a work focused on zero-shot super-resolution (ZSSR), which trained deep CNNs based on a single example, relying on the earlier findings in the literature regarding patch recurrence in images [137, 138]. This was done by training a convolutional neural network

to super-resolve artificially downsampled crops of the test image with the original crop used as ground truth, and then, applying the same network to the test image to go beyond the original resolution.

Another line of work explored the prior enforced by the sole use of convolutional neural network architecture, starting with the introduction of Deep Image Prior (DIP) [139]. It has been found that optimizing randomly initialised parameters of a CNN enforces a strong constraint on the type of images that can be used, which can be employed for a variety of synthetic tasks, such as inpainting, super-resolution, or denoising.

The approach of using convolutional architectures as a source of prior in zero-shot settings has been further explored in works, such as Double-DIP [140], capable of decomposing mixed images into two sources of a mixture, or in a system designed for video inpainting [141]. Convolutions have also been used for super-resolving videos in the temporal dimensions [142].

Another useful technique employed in several works is the inclusion of adversarial losses in an internal learning framework. This has resulted in techniques such as Internal GAN (InGAN) [143], which uses adversarial loss to train a generator to synthesise different versions of the same image that share the same patch statistics. Other related works include the use of adversarial losses for zero-shot super-resolution [144], or a more general SinGAN approach suitable for a variety of tasks such as harmonisation, super-resolution, or animation [145]. In the case of SinGAN, the generator is constructed using networks transforming noisy input into likely patches at a specific scale, and stacking modules for several scales on top of each other. For each scale, a discriminator operating solely at that scale is used for the adversarial loss. A similar method of TuiGAN [146] has also been proposed for image-to-image translation, by including two generators to switch between two domains and enable a cyclic loss.

Finally, other works performing internal learning using additional losses from a pre-trained feature extractor (such as VGG19) have also been proposed [147, 148]. Another line of work investigates internal single image synthesis based on patch nearest neighbours (no deep neural networks involved) and achieving high-quality output [149]. In the same work, it has been found that adversarial neural network approaches are

capable of synthesizing new similar patches not observed in the input (where nearest-neighbour approaches cannot), but this comes at a cost of relatively long optimisation process (often taking hours on the recent GPU models). The patch nearest neighbours can arrive at a solution in a matter of seconds. It should also be stated that the internal learning approaches without adversarial losses are generally much faster to train and can generate output in a matter of several minutes.

Neural Style Transfer

The mentioned internal learning approaches in [147, 148] rely on the gradients propagated from large pre-trained external networks. This is related to a whole area of work in the theme of neural style transfer [150–159].

As early as 2016, Gatys et al. proposed that the style and content of images could be extracted from different sets of features inside a pre-trained deep classifier network, such as VGG [160]. This fact was used to define perceptual losses for style and content, which could then be used to optimise random input to minimise these losses and match the style and content of two source images.

Later advancements include training a model that can translate an input image into a trained style in a single forward pass of an image-to-image translation network (instead of following an optimisation procedure at test time) [151], mixing any style and content input in real-time with adaptive instance normalisation [152], separate style and content encoders [161], attentional-based style networks [153], multi-adaptation architecture [154], additional structural losses and decoders [158], manifold alignment of the style and content encoded features [159], adaptive attention normalisation [157], a combination of internal-external and contrastive learning [156], and progressive attentional manifold alignment [162].

2.6 Problem Formulation

As part of the background, what follows is a more precise formulation of the cloud removal problem. It is possible to interpret it as an inverse problem, which corresponds

to the reconstruction of an unknown signal y based on a limited observation x [163]. The observation y is the output of a forward process, a potentially non-invertible function $D()$ applied to x.

$$y = D(x) \tag{2.13}$$

The forward process could involve many types of degradation, such as masking, downsampling, blurring, or additive noise. The presence of clouds can be treated as a mixture (via element-wise multiplication $\odot$) between two sources x (cloud-free image) and c (cloud component), based on a mask M.

$$y = D(x) = M \odot x + c \odot (1 - M) \tag{2.14}$$

This model is quite flexible since it can easily model any type of effect related to clouds. If the mask M equals 1, then the pixel is unaffected by the cloud. Otherwise, the pixel is mixed with c, which is either the colour of the cloud reflection or the resulting light colour received due to cloud shadow.

Cloud removal is defined as a reverse operation $R(y)$ applied to the observed input satellite image y (which could be cloudy or cloud-free). The reverse operation $R()$ is aimed to counteract the degradation operation D that models the presence of clouds in the image based on an ideal cloud-free representation x of the image.

$$R(y) = R(D(x)) \approx x \tag{2.15}$$

However, the degradation operation D can destroy some of the information in the original sample x, depending on the value of M. In that case, the reverse operation $R()$ must approximate a whole distribution of potential samples $p(x|y)$ rather than one specific sample.

$$R(y) \sim p(x|y) \tag{2.16}$$

This distribution can generally be expected to be rather wide, meaning that almost

every cloudy sample y could correspond to a very large number of possible cloud-free images with non-zero likelihood, that could all have been a feasible original sample. This may make both training and evaluating the network problematic, and hence, a potential solution is to narrow this distribution by introducing more conditions y_c.

$$R(y) \sim p(x|y, y_c) \tag{2.17}$$

These conditions could be images coming from different sensors, a different point in time, or they could be any other piece of information useful for predicting the original image.

2.7 Metrics

The quality of the solutions and approximations for the problem of cloud removal can be assessed in a quantitative fashion by computing the distance between the tested approximation $\hat{x}$ and ground truth reference x. However, it is not immediately clear what type of distance, or a metric, should be used to measure the similarity between the two.

A simple approach is to compute some type of pixel-wise error and average it. In such cases, the similarity computation is performed independently for every pixel and hence, there is no spatial context involved. One example of that is the mean absolute error (MAE), where the absolute difference between $\hat{x}$ and x for each of the N pixels in the image is computed:

$$\text{MAE}(\hat{x}, x) = \frac{\sum^N |\hat{x} - x|}{N} \tag{2.18}$$

Other popular pixel-wise metrics, related to the second order, are mean square error (MSE) and root mean square error (RMSE), equal to the average squared difference and the square root thereof:

$$\text{MSE}(\hat{x}, x) = \frac{\sum^N |\hat{x} - x|^2}{N} \tag{2.19}$$

$$\mathrm{RMSE}(\hat{x}, x) = \sqrt{\frac{\sum^{N} |\hat{x} - x|^2}{N}} \tag{2.20}$$

A common pixel-wise metric in the field of image compression is peak signal-to-noise ratio (PSNR) [164], which is defined as the logarithmic ratio between the power of the image signal and the power of the present distortion:

$$\mathrm{PSNR}(\hat{x}, x) = 20 \log_{10} \frac{MAX_x}{\mathrm{RMSE}} \tag{2.21}$$

The metric can be defined via the ratio of the maximum value of the image representation MAX_x to the MSE.

These metrics have no context of space. In contrast, structural similarity index (SSIM) [165] does consider this context, and is defined as the product of three factors: *luminance*, *contrast*, and *structure*. These three components are computed across many patch neighbourhoods, most often by applying an 11-by-11 pixel Gaussian kernel with σ=1.5 pixel. More specifically, the approach of applying a local window across the images and averaging is technically mean SSIM, or MSSIM, but in many works and packages it is simply referred to as SSIM and in this work, the same assumption applies in the reported results.

The *luminance* factor l is based on the means $\mu_{\hat{p}}$ and μ_p of the two source patches $\hat{p}$ and p at a given location (factors C_i are static parameters used for stability):

$$l(\hat{p}_i, p_i) = \frac{2\mu_{\hat{p}}\mu_p + C_1}{\mu_{\hat{p}}^2 + \mu_p^2 + C_1} \tag{2.22}$$

The *contrast* c is a similar measure applied to variance values $\sigma_{\hat{x}}$ and σ_x for each source patch:

$$c(\hat{p}_i, p_i) = \frac{2\sigma_{\hat{p}}\sigma_p + C_2}{\sigma_{\hat{p}}^2 + \sigma_p^2 + C_2} \tag{2.23}$$

The third factor of *structure* s is slightly different as it is based on the covariance $\sigma_{\hat{p}p}$:

$$s(\hat{p}_i, p_i) = \frac{\sigma_{\hat{p}p} + C_3}{\sigma_{\hat{p}}\sigma_p + C_3} \tag{2.24}$$

The exact local value of SSIM is equal to a product of these three factors:

$$\text{SSIM}(\hat{p}_i, p_i) = l(\hat{p}_i, p_i)^\alpha c(\hat{p}_i, p_i)^\beta s(\hat{p}_i, p_i)^\gamma \tag{2.25}$$

And finally, for N total neighbourhoods i over which a window has been applied, the average SSIM for the two images $\hat{x}$ and x is equal to:

$$\text{MSSIM}(\hat{x}, x) = \sum_i^N \text{SSIM}(\hat{p}_i, p_i) \tag{2.26}$$

This value of mean SSIM is generally reported as SSIM throughout this work.

The computer vision community, especially after the growth of deep generative models for images, has also explored the topic of image comparison metrics. In many contexts, it is helpful to determine the perceptual similarity between two images or two sets of images. This motivated the definition of perceptual metrics, which aim to compare high-level abstract features and appearances in images. This includes the work on the style and content losses based on deep perceptual losses [151], Fréchet Inception distance [166], LPIPS [167], or DISTS [168]. However, these metrics are based on the features learned by models on generic datasets, such as ImageNet [95] and attempt to emulate human perception. It is not entirely clear whether the similarity expressed by these metrics amounts to an improvement for remote sensing tasks and hence, has not been used much in the literature on satellite cloud removal.

Another solution to measure the utility of a given cloud removal solution is to test the performance on a proxy test, for example, satellite image segmentation, for which well-defined ground truth exists. That allows one to immediately understand the impact of the cloud removal tool on a task of interest. It can be expected that the cloud removal tools are going to be deployed for more narrow tasks in many cases, so their performance could be relative to the specific domain they are subsequently employed for. However, this type of evaluation requires the selection of the end task and there is a risk that the resulting conclusions could only apply to that task. Consequently, this

approach has not been explored extensively for assessing general cloud removal models.

In the field of cloud removal, a variety of metrics has been used, however, a large overlap can be observed between individual manuscripts evaluating the performance of cloud removal tools. The majority of works will report the SSIM [63, 64, 69–71, 169] supported by some pixel-wise metric such as PSNR [63, 64, 68–71, 169], RMSE [63, 68, 71, 169], or MAE [63, 69, 169]. In some cases, a less conventional metric, such as spectral angle mapper (SAM) (computing the error as the angle between the spectral components) [170] was used [69, 71, 169].

In this work, it is recognised that the existing metrics could have certain limitations. Ultimately, under the assumption that an ideal cloud removal transformation is not attainable, some error between the ground truth and the cloud removal output should be expected. However, the magnitude and impact of this error depend heavily on the end application. The invention of new metrics as well as their relationship to the relevant end tasks could be an important direction of research but lies outside of the scope of this work. Consequently, the existing metrics of SSIM or RMSE with a relatively simple definition provides a good approximation of the general performance and will be used for further analysis.

Chapter 3

Internal Learning from the Sample

Perhaps the most rudimentary level of learning can occur in the context of the sample itself. For example, in the task of image completion, the information contained in the known pixels is often a powerful source of information for predicting the content of the missing pixels. This type of learning regime is often referred to as internal learning (a term popularised in [171], where it is described as "*internal self-supervision*"), as opposed to external learning, when the priors are extracted from a larger number of samples.

The approach of constraining the learning context to a single image may at first seem quite limiting, however, it can bring a certain advantage. Internal learning approaches will generally offer higher flexibility in regard to the network topology and data source than externally trained models. An externally trained model requires the same topology to be preserved in order to reuse the optimised weights. In contrast, an internal learning approach allows for this topology to be adjusted based on the processed sample (for example, the size of the neural network and, conveniently for multi-spectral images, the number of channels in the image representation) since the parameters are learned from scratch. As it will be shown, this is highly beneficial for satellite imagery, which comes in a variety of spatial and channel shapes. Another advantage is that internal learning does not suffer from dataset bias to the extent the externally trained networks do.

Naturally, this comes at the cost of a less-specific and hence, likely less powerful prior. Finally, there are no training costs associated with internal learning, and consequently, no costs of acquiring and curating data. At the same time, it is crucial to acknowledge that there are at least two potential disadvantages of internal learning. First, for many tasks, the solutions may simply be less powerful than alternative data-based techniques. Second, the inference routine requires a short optimisation process, which will generally result in a more costly inference compared to an external model.

There already exists a body of literature focused on the applications and techniques of deep internal learning, a term referring to the use of deep learning methods in the context of the single sample [139, 145, 171]. This includes many solutions to tasks such as inpainting [139], super-resolution [139, 145, 171], or denoising [139]. However, the majority of works focus on general computer vision applications and there has only been a limited number of works focused on the application of deep internal learning to satellite image processing [80]. In this chapter, this gap is filled by setting the primary objective to propose and evaluate architectures that can incorporate additional sources, such as Sentinel-1 data into the synthesis process.

To explore the utility of the internal learning practice in the context of satellite image cloud removal, this work proposes several techniques of varying complexity. These techniques are largely based on a convolutional topology and aim to fulfil the task of inpainting Sentinel-2 images based on additional informing sources, such as Sentinel-1 SAR acquisition, or a historical representation of Sentinel-2 capture from the same region. The task of inpainting can be effectively turned into cloud removal if a cloud detection (or more technically, segmentation) tool is used to identify the areas affected by clouds or shadows.

In this chapter, the methods for satellite image inpainting are evaluated to an open-source dataset containing a record of samples from a region in Scotland for the years 2019 and 2020 (available at https://zenodo.org/record/5903334) [172] to compare their performance. The example samples from the dataset are shown in Figure 3.1. Furthermore, the best-performing approach from this chapter is contrasted against other types of learning explored in subsequent chapters on a dataset containing samples

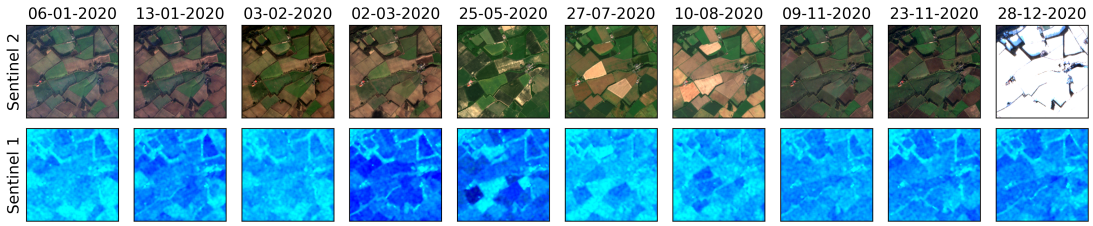


Figure 3.1: Selected samples from the Scotland dataset, consisting of pairs of cloud-free Sentinel-2 and Sentinel-1 images.

from a wider set of locations. This can be found in Chapter 6.

In summary, this chapter brings the following contributions. First, three novel solutions for solving synthetic tasks on satellite images internally are introduced. Their utility and versatility are demonstrated in two use cases, inpainting (later used for cloud removal) and super-resolution. Since these techniques are based on an internal optimisation process performed at inference, a convergence analysis for identifying optimal solutions is conducted. Finally, the proposed techniques are evaluated for multi-spectral image inpainting in a region-oriented context over a period of one year.

3.1 Deep Image Prior

Deep Image Prior, introduced in [139], refers to a technique of fitting a randomly initialised deep CNN to produce a source image observation as output, as illustrated in Figure 3.7. The term *Deep Image Prior* indicates that the sole topology of a CNN already constitutes a strong priors about the domain of natural images. As a result, such models exhibit a certain *impedance* to noise and naturally prioritise correlations present in the data.

The processed image x can be any type of incomplete observation of visual data, such as a masked image, low-resolution image, or image with added noise, which can be modelled as the output of a specific degradation operation D applied to the true image x_0.

$$x = D(x_0) \tag{3.1}$$

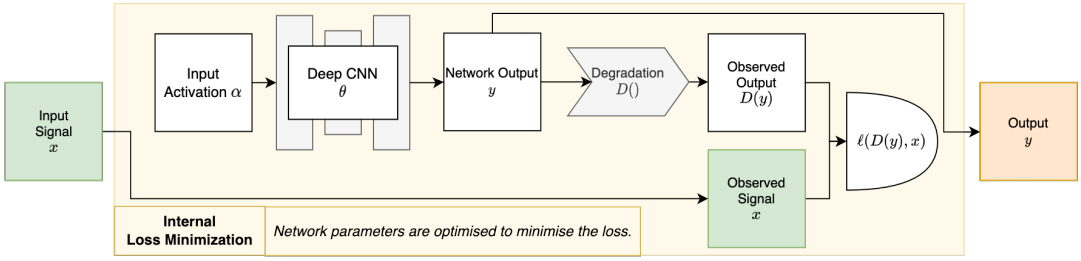


Figure 3.2: Diagram of the Deep Image Prior technique.

By propagating any activation signal α (in practice, this is often a sample from uniform noise, or a mesh grid of coordinate-based values as described in [139]) through a CNN with parameters θ, the output y can be derived as a value of the function f_θ parameterised by the network.

$$y = f_\theta(\alpha) \tag{3.2}$$

It can be expected that output y will not have any relevant value for randomly initialised network parameters θ. However, these parameters can be optimised with respect to the distance d between the network output y and the limited observation x, based on the known degradation operation $D()$ (such as a mask or a downsampling kernel).

$$\mathcal{L}_\theta = \ell(D(y), x) \tag{3.3}$$

When the loss function $\mathcal{L}_\theta$ is minimised, the convolutional kernels are optimised based on the gradients propagated through the degradation operation from the observed component of the image. For example, if the degradation $D()$ is a mask, only the gradients from the non-masked pixels will be propagated. As a result, a lot of the information about the structure and textures in the known region is encoded into the weights of the network, and will often naturally transfer to the full scope of the image y.

This effect closely links to the effect described as *noise impedance* in the original Deep Image Prior paper [139]. Figure 3.3 demonstrates that a CNN architecture will

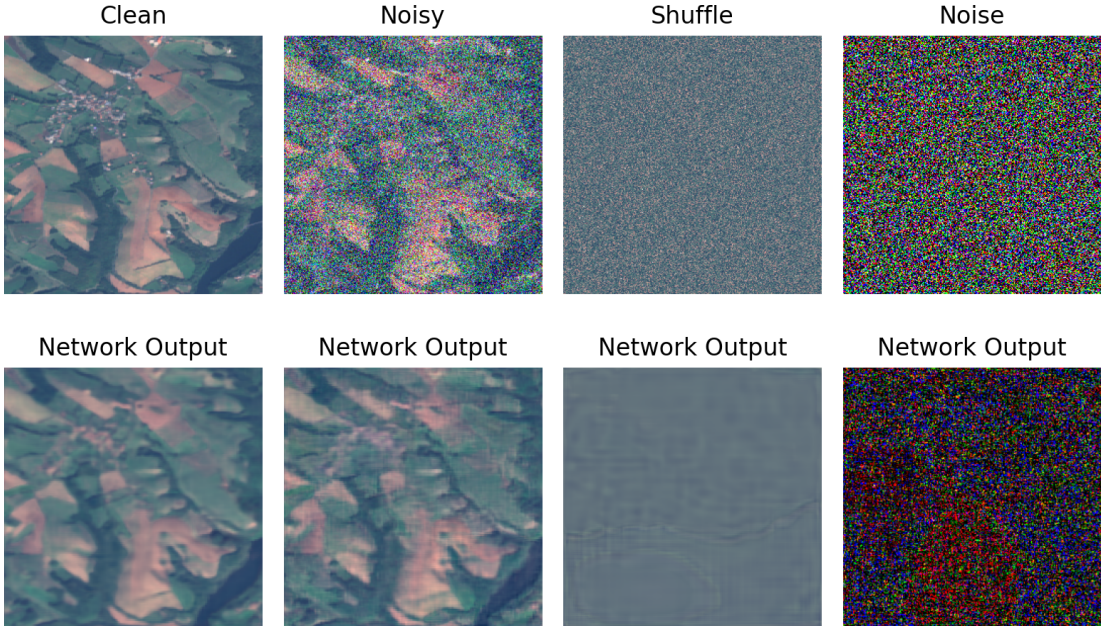


Figure 3.3: Examples of source images and the output of the convolutional module optimised over 2,000 update steps. The clean image is a Sentinel-2 image from the test dataset. Noisy contains additional noise, while shuffle contains the original image with shuffled pixels. Noise contains noise from a normal Gaussian distribution.

converge towards structured data output, such as natural images, much faster than towards unstructured output, such as noise. The experiment has been conducted by optimizing a standard convolutional network (described in more detail in the next paragraph) to 4 types of images: i) a clean satellite image, ii) the same satellite image with a level of added Gaussian noise, iii) the clean satellite image with randomly shuffled pixels, and iv) a sample of uniform noise. Figure 3.3 shows these images in the top row and the network output after 2,000 optimisation steps in the bottom row (an optimisation step is a single update of network parameters based on the gradients computed using the internal loss). For this experiment, the degradation operation $D(y)$ has been set to an identity operation $I()$, meaning that all pixels contribute to the loss. This is done in order to show that even when a complete signal is observed, the CNN network will converge to structured signals much faster than to unstructured ones. The evolution of loss for each image is shown in Figure 3.4 to further confirm that point. The effect of noise impedance can be explained by the fact that in structure data, a higher alignment between gradients from different parts of the image can be

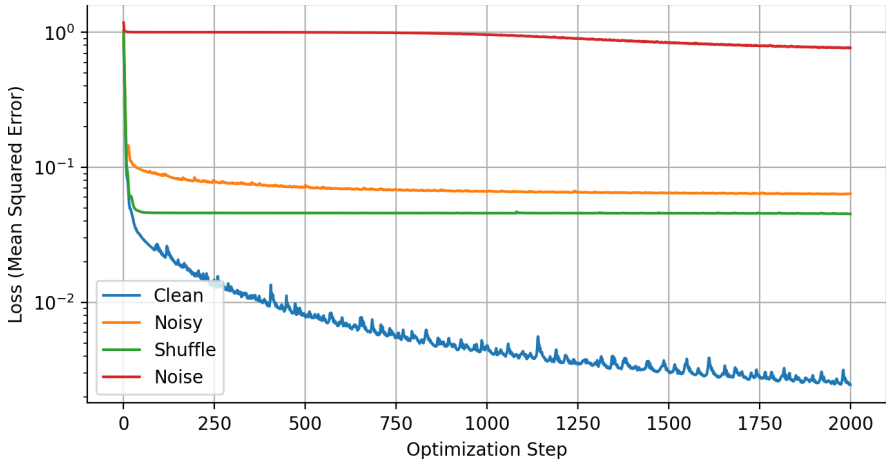


Figure 3.4: Loss trace for the 4 example images (Figure 3.3) over 2,000 optimisation steps with the Deep Image Prior technique.

expected, whereas in random noisy images, these gradients will not be correlated.

In summary, the process for this synthesis approach starts with a randomly initialised neural network (more details below), the parameters of which are optimized to minimize the difference between the network output (in response to a pre-defined static activation input) and the observed signal (such as partially incomplete image). After a certain number of optimization steps the network output is used as the output of the reconstruction process.

In the work introducing Deep Image Prior [139], a simple convolutional architecture (referred to as SkipNetwork, and quite similar to U-Net) is used with an encoder-decoder topology and additional skip connections across each level. Each task presented in that work is solved with a slightly different configuration (that has been motivated by empirical insights in an attempt to increase the performance for each task).

The SkipNetwork architecture is shown in Figure 3.5 and consists of three types of blocks. The network input is processed in sequence by a set of N downsampling blocks (green), producing N intermediate feature maps, each half the size of the previous. Furthermore, each computed feature map is optionally fed into a corresponding skip block (yellow) to compute a skip feature map that can later be injected into the decoding stage. The decoding stage begins by feeding the N^{th} feature map to the lowest level upsampling block (pink). Similarly, to the encoding process, the decoding involves a

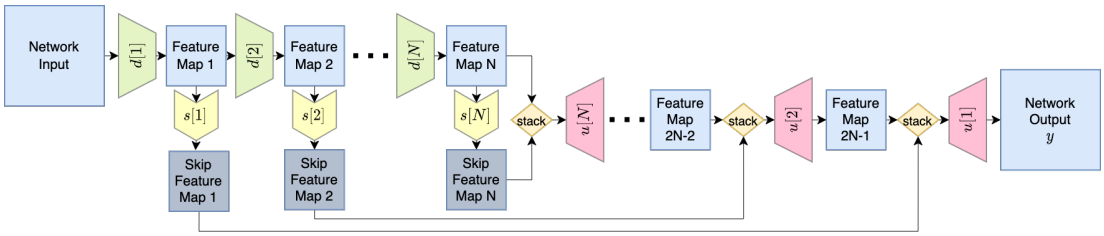
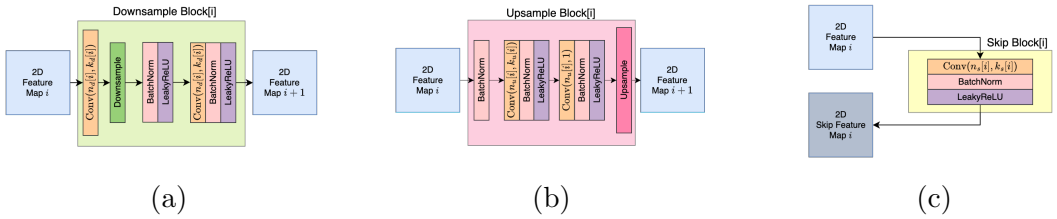


Figure 3.5: Architecture of the SkipNetwork used as the backbone for solutions involving Deep Image Prior. Figure based on Fig.21 in [139].



(a) (b) (c)

Figure 3.6: Three building blocks of a SkipNetwork (a) a Downsample Block (b) an Upsample Block (c) an (optional) Skip Block.

sequence of upsampling operations applied in a chain to the feature map. At each step, the features from a skip block at the corresponding level (if available) are combined with the feature map via concatenation and then treated as input to the next upsampling block.

The exact implementation of each block is shown in Figure 3.6. A downsampling block (Figure 3.6(a)) at level i contains a convolution with kernel size $k_d[i]$ and $n_d[i]$ output channels. Downsampling is performed by setting the stride of the first convolution to 2 (leading to an output half the size), this aspect is still represented by a separate downsample block in green. This is followed by a sequence of a batch norm, a leaky ReLU activation, and another convolution (this time stride of 1 and no downsampling) with the same kernel size and output channels, followed by another batch norm and leaky ReLU. The content of an upsampling block is shown in Figure 3.6(b) and it includes a batch norm, followed by a sequence of two groups of operations, each containing a convolution with kernel size $k_u(i)$ and output channels $n_u(i)$, a batch norm and leaky ReLU activation. Finally, an upsampling operation is performed to increase the spatial size of features using nearest neighbour upsampling. The optional

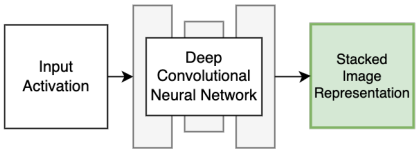


Figure 3.7: Diagram of the *Stacked* technique applied for satellite images based on Deep Image Prior.

Table 3.1: Stacked architecture parameters for inpainting

Parameter	Value
Input activation α	meshgrid
Downsampling block channels n_d	[16, 32, 64, 128, 128, 128]
Downsampling block kernels k_d	[3, 3, 3, 3, 3, 3]
Upsampling blocks n_u	[16, 32, 64, 128, 128, 128]
Upsampling kernels k_u	[5, 5, 5, 5, 5, 5]
Skip blocks n_s	None
Skip kernels k_s	None

skip block shown in Figure 3.6(c) contains only a single convolution with a kernel size $k_s[i]$ and output channels $n_s[i]$, followed by a batch norm and a leakyReLU activation. If used, they are concatenated with upsampled features from the corresponding level in the decoder part.

In this work, a large portion of the configuration is based on the large-hole inpainting use case from the Deep Image Prior paper [139] and the architecture illustrated in Figure 3.5 with the parameters listed in Table 3.1 is used. To instil an additional smoothness prior, a mesh grid activation α is used, as suggested in the original paper. The mesh grid input contains 2 channels, each with a value between 0.0 and 1.0 corresponding to the position of a given pixel within each spatial axis.

The original Deep Image Prior is applied to natural RGB images. In this chapter, we explore the potential of using multiple spatially aligned images as a representation (for example, the inpainted Sentinel-2 image concatenated with a Sentinel-1 representation). To accommodate additional data sources for satellite image inpainting (such as SAR image, or historical optical image), the channels of the output layer are ad-

justed appropriately, and the observed image representation consists of a stack of the inpainted signal with potential support representations. The abstract representation of this approach, referred to as *stacked* variant from here on, is included in Figure 3.7.

The stacked approach described above can yield a reasonable level of performance. However, the approach of stacking signal representations may not be entirely appropriate when these come from different domains. This motivates the introduction of a wider framework of multi-modal convolutional parameterisation network (MCPN) introduced in the following section.

3.2 Proposed Method: Multi-Modal Convolutional Parameterisation Networks

Combining the aligned images into a single representation, as done in the stacked approach, might be too restrictive. For example, radar images can be expected to contain very different types of textures and higher levels of noise compared to optical data. As a result, the stacking technique requires the convolutional kernels in the last layers of the network to potentially learn a very diverse set of textures, which could potentially lead to suboptimal performance. This motivates the introduction of the MCPN technique, which seeks a trade-off between information sharing between signal domains and the freedom to easily reconstruct each domain.

The MCPN consists of a single core network (very similar to the main SkipNetwork used in the stacked approach) for producing a shared signal representation, as shown in Figure 3.8. The core synthesis network is responsible for producing a shared core signal from which all domain-specific signals can be derived. The derivation is carried out by domain-specific convolutional heads that transform the shared core signal to individual target domains. In effect, spatial information sharing between the domains is enforced by relying on the same shared core signal. Finally, a set of domain cycle heads is used to convert each domain target signal back to the shared core signal and promote consistency of inpaintings. This arrangement yields two loss terms optimised by the network, the domain-specific loss $\mathcal{L}_D(M)$ computed between the synthesised domain

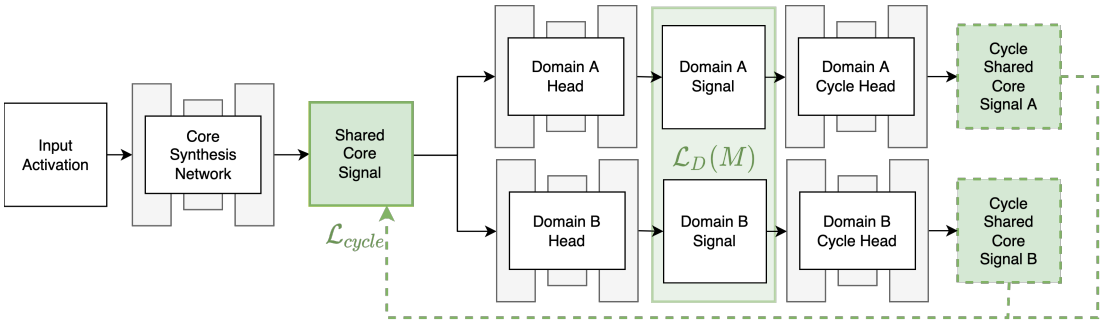


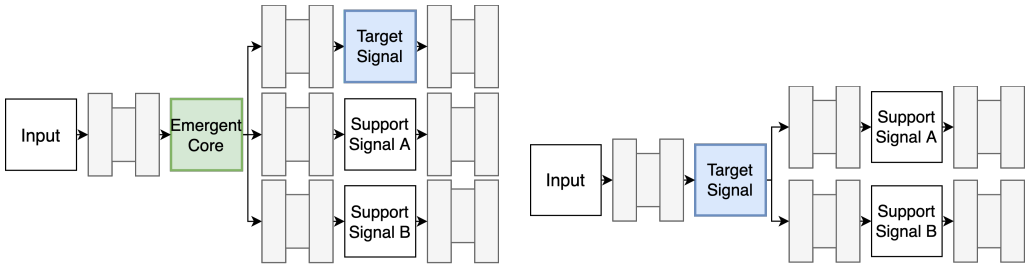
Figure 3.8: Diagram of the MCPN approach, consisting of a core synthesis network and domain-specific heads. Each domain path is also supported by a cyclic head to preserve shared structure between domains.

signals and the existing target reference, applied with an appropriate domain-specific mask M, and the cycle consistency loss $\mathcal{L}_{cycle}$ computed as the difference between the shared core signal and the outputs of the cycle heads. Both losses $\mathcal{L}_D(M)$, $\mathcal{L}_{cycle}$ are computed as MSE between the respective inputs.

The shared core signal is learned in an emergent fashion by backpropagating from the sum of individual domain-specific reconstruction losses $\mathcal{L}_D(M)$ and the cyclic terms $\mathcal{L}_{cycle}$. This variant of MCPN is referred to as *Emergent Core* MCPN and is further illustrated in Figure 3.9(a). Another possibility is to use the synthesised signal of interest, such as an incomplete optical image, as the shared core representation, effectively dropping one of the domain-specific branches. Hence, $\mathcal{L}_D(M)$ is defined as a sum of a loss directly computed on the synthesised signal of interest at the output of the core network, plus the domain-specific losses, at the output of the head networks. This variant is referred to as *Direct Core* MCPN, as shown in Figure 3.9(b).

3.2.1 MCPN Framework Configuration

The capacities of the core network and the domain-specific heads determine how much information is contained in the shared core signal. As the capacity of the domain-specific heads decreases, the possible transforms between the shared core and individual output images are simpler. This capacity can be controlled by the number of layers, their width, and the activation functions applied to the networks. In this study, the core network is identical to the SkipNetwork employed in Deep Image Prior [139] for the in-



(a) Emergent Core (b) Direct Core

Figure 3.9: The Emergent Core variant (a) allows for a shared core signal to be learned in an emergent fashion, synthesizing each domain signal, including the target image (blue) using specific heads. The Direct Core variant (b) instead uses the target image (blue) as the shared core signal, leading to one less domain head network

painting task with the configuration where $n_d = n_u = [16, 32, 64, 128, 128, 128]$ (where n_d and n_u are the numbers of channels of the downsampling and upsampling submodules, respectively) and skip modules of 4 channels. The domain-specific networks are also composed of similar elements, but contain only two stages of [32, 32] channels, along with skip modules also with [32, 32]. For the Emergent variant, downsampling and upsampling operations are maintained as in the Deep Image Prior reference network. For the Direct variant, stride of 1 is used for all layers, resulting in preserved representation shape and no spatial bottleneck.

Further important factors influencing the performance are the size of the convolutional kernels in the domain-specific heads and the number of channels of the shared core representation. If the domain-specific kernels are set to the size of 1×1, then all pixels of the shared representations are processed independently, which forces the shared representation to contain a lot of information for every output pixel. If the kernel size is increased, then the local neighbourhood information is passed on to the head network, meaning that individual shared pixels do not have to contain full global context information. In this study, based on exploratory analysis, it has been found that a core representation with 8 channels for the Emergent variant, and head kernel sizes of 3×3 provide an appropriate baseline configuration. For the Direct variant, the number of channels in the core representation is by definition equal to the number of channels in the target representation.

Table 3.2: Parameters used for the MCPN architecture used for satellite image inpainting

Parameter	Value
Input activation α	meshgrid
Downsampling block channels n_d	[16, 32, 64, 128, 128, 128]
Downsampling block kernels k_d	[3, 3, 3, 3, 3, 3]
Upsampling blocks n_u	[16, 32, 64, 128, 128, 128]
Upsampling kernels k_u	[5, 5, 5, 5, 5, 5]
Skip blocks n_s	[4, 4, 4, 4, 4, 4]
Skip kernels k_s	[5, 5, 5, 5, 5, 5]
Head Network Base	[32, 32]
Head Network Skip	[32, 32]
Head Kernel Size	[5, 5]
Head Activation	None

In the evaluation section, and for all resulting images, the configuration listed in Table 3.2 is used for MCPN, unless otherwise stated. Optimisation is carried out by employing an AdamW optimiser with standard parameter values (betas of (0.9, 0.999), eps of 10^{-8}, and a weight decay of 0.01). The learning rate and the number of optimisation steps are determined based on the convergence discussion in Section 3.3.

3.3 Detecting Model Convergence

Comparison of convolutional parameterisation approaches (the two proposed MCPN variants and the Stacked baseline) is challenging because they may require different learning rates to allow stable convergence and a different number of optimisation steps. Hence, setting the same learning rate and applying the same number of weight updates for all architectures may put some of the models at a disadvantage and bias the evaluation. To explore this effect, a set of experiments was carried out where the performance computed based on known ground truth is traced for 20,000 optimisation steps (this number has been selected to gain more perspective than the usual 2,000-4,000 steps as in [139]). The two metrics used to measure the quality of a synthesised image with reference to a specific ground truth are SSIM and RMSE.

In the seminal work on the Deep Image Prior [139], the output was produced using the weights obtained after applying a fixed number of optimiser steps, depending on

the task, usually a few thousand steps were used. Alternatively, an adaptive strategy for detecting a suitable convergence state can be devised. One solution for adaptive convergence detection is to measure the performance of the synthesis on the known, non-masked region and base a stopping criterion on that quantity, for example, RMSE between the source image and the network output in the non-masked region. This is proposed here as a known reconstruction RMSE strategy. However, in some scenarios, the reconstruction error of the known region could monotonically decrease or saturate without ever reaching a minimum, while the error in the inpainting could be increasing, thus yielding a poor solution.

Hence, another adaptive approach is proposed that takes into consideration the inpainted region too. This is done by measuring the quality of inpainting as the similarity of texture patches between the inpainted and known region of the image, termed as patch consistency metric. The metric is computed as the Fréchet distance between the two distributions of low-level features of a pre-trained Inception network [173] in response to the inpainting region and the distribution of features from the known region, in a similar fashion to single image Fréchet inception distance (SIFID) [145]. The metric requires the computation of a feature map $\mathcal{F}_{source}$ of the source image and a feature map $\mathcal{F}_{output}$ of the image produced by the network. Patch representations of the known and inpainted regions can be obtained by applying the inpainting mask to the $\mathcal{F}_{output}$ and the inverse of the mask to $\mathcal{F}_{source}$. Since the size of the feature maps is reduced in the layers of the Inception model, the mask $\mathcal{M}$ must also be reduced to apply it to the feature maps $\mathcal{F}_{source}$ and $\mathcal{F}_{output}$. This is achieved by downsampling the mask M in the same manner as the features to obtain a downsampled feature mask $\mathcal{M}_F$. The feature mask $\mathcal{M}_F$ equals to 1.0 if and only if a given feature is affected by any pixel from the inpainting region. This way, the features extracted with the mask $\mathcal{M}_F$ correspond to all features affected by the synthesised pixels, and the features extracted with the inverse mask $\mathcal{M}'_F$ filter out features that are only affected by the known region. The downsampling strategy is such that any feature influenced by the boundary effect is assigned to the inpainted region of the mask.

Figure 3.10 demonstrates how the values of the two adaptive convergence metrics

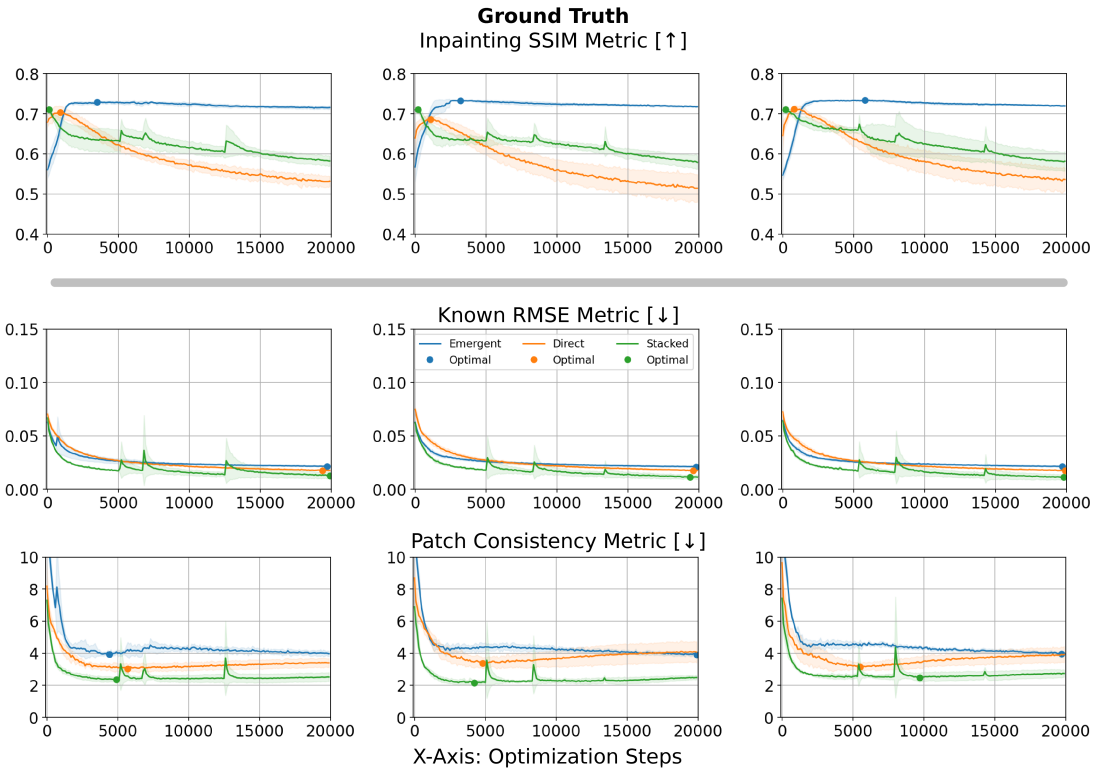


Figure 3.10: Values of the ground truth inpainting SSIM and two considered convergence metrics (Known Region RMSE and Patch Consistency) evolving over 20,000 optimisation steps.

and ground truth inpainting SSIM evolve over the optimisation process. The rows in order, correspond to i) ground truth inpainting SSIM, ii) known reconstruction RMSE, and iii) patch consistency. The three columns correspond to the tested learning rates of 10^{-4}, 10^{-3}, and 10^{-2}. The traces have been obtained for 4 repeated runs of inpainting a single Sentinel-2 sample based on Sentinel-1 informing data. The analysis of the convergence detection methods is performed for the SAR-to-optical synthesis because it represents the most challenging scenario where the gap between domains is considerable. The inpainting mask used covers the whole image except for a border of 50 pixels around the image's periphery.

The top row in Figure 3.10 contains the traces recording the value of the inpainting SSIM, which has been computed with reference to the ground truth. The maximum value of the trace indicates the top performance that could be achieved by each model, but extracting it requires knowledge of the ground truth, not available in practice. The maximum values of inpainting SSIM tend to occur within the first 5,000 steps for all three model types, regardless of the learning rate (with the exception of Emergent Core at 10^{-2} learning rate, which occurs at around 6,000 steps). For the Stacked baseline (green line), this maximum value appears to be reached early into the process. This relates to the dynamics of Deep Image Prior convergence [139], where the low-frequency components are fit first before the fine detail. In effect, the low-frequency approximation at the beginning of the optimisation scores better than later solutions where high-frequency components are synthesised. The second and third rows contain the traces of the adaptive convergence detection metrics. The known region RMSE decreases monotonically (with occasional local spikes), and may be a poor choice for a proxy metric. In the third row, the patch consistency reaches a minimum closer to the top performance states identified by the ground truth SSIM, but the two do not seem to align particularly well; for example, the minimum patch consistency is achieved long after the top performing SSIM for Direct and Stacked model variants.

As a supplement to Figure 3.10, Table 3.3 contains the average of extreme values of the inpainting SSIM, known region RMSE and patch consistency for four repetitions. Furthermore, for each record, the table presents the mean and standard deviation of the

number of optimisation steps after which the extreme value was reached. The Inpainting SSIM provides an indication of which learning rate results in the best performance for the tested sample. It is further apparent, that the minimised known region RMSE occurs very late in the training process and it is possible that it could keep decreasing further with more steps in the experiment. The patch consistency does reach a minimum value closer to the top-performing Inpainting SSIM. Lastly, based on this result, the learning rate for each model has been selected, with 10^{-3} for MCPN Emergent and 10^{-2} for MCPN Direct and the Stacked baseline.

Using these learning rates, a broader experiment to find the optimal convergence detection technique is conducted on the whole dataset of cloud-free images from Scotland (described in Section 3.4.1), with 4 repetitions on each sample and with the same mask leaving out a 50-pixel border in the image. The average maximum inpainting SSIM that can be achieved is presented in the first column of Table 3.4, while the remaining columns contain the average inpainting SSIM obtained using the three convergence detection strategies (4,000 steps, known region RMSE and patch consistency). The Known Region RMSE method is resulting in the top inpainting SSIM for the MCPN Emergent variant. This is likely caused by the optimisation dynamic that can be observed in the traces of Figure 3.10 (top row, blue lines), where the MCPN Emergent solution reaches a fairly stable plateau, where on average, it may be more beneficial to train longer to ensure all samples in the dataset can reach a stable plateau. The traces of the other two methods (top row, orange and green lines) do not exhibit this level of stability, as the Inpainting SSIM tends to decrease monotonically. This results in higher sensitivity in the number of optimisation steps and in particular, the top inpainting SSIM is achieved early and consistently in fewer than 4,000 steps. In contrast, the other convergence strategies prefer solutions after 4,000 steps (as shown in Table 3.3) and they are likely to yield lower inpainting SSIM for MCPN Direct and the Stacked baseline, as shown Table 3.4. The rest of the experiments in this chapter are conducted with model-specific learning rates.

Table 3.3: Optimal values of the ground truth reference and the two adaptive convergence methods, along with the number of steps after which they were obtained

Method	LR	Inpainting SSIM (GT) ↑	Known RMSE ↓	Patch Consistency ↓
MCPN Emergent	10^{-4}	0.732 at 3600 ± 1579	0.022 at 19325 ± 491	3.782 at 8600 ± 5839
	10^{-3}	0.735 at 3725 ± 914	0.021 at 19575 ± 449	3.805 at 15275 ± 7553
	10^{-2}	0.735 at 3575 ± 1380	0.021 at 19525 ± 363	3.875 at 15000 ± 7510
MCPN Direct	10^{-4}	0.706 at 800 ± 254	0.017 at 19400 ± 494	2.989 at 5375 ± 1028
	10^{-3}	0.689 at 1350 ± 390	0.018 at 19600 ± 212	3.322 at 6425 ± 1987
	10^{-2}	0.714 at 1000 ± 158	0.017 at 19850 ± 50	3.025 at 5750 ± 1425
Stacked	10^{-4}	0.713 at 75 ± 43	0.011 at 17650 ± 3332	2.261 at 9700 ± 5980
	10^{-3}	0.712 at 1400 ± 2136	0.011 at 19450 ± 384	2.032 at 9025 ± 4935
	10^{-2}	0.716 at 3450 ± 3377	0.011 at 16525 ± 5063	2.367 at 8675 ± 4246

Table 3.4: Mean inpainting SSIM values for different convergence detection strategies achieved for 4 repetitions carried out on the entire Scotland dataset (SAR guidance)

Method	LR	Ideal (GT)	4,000 Steps	Known RMSE	Patch Consistency
MCPN Emergent	10^{-3}	0.677	0.626	**0.669**	0.637
MCPN Direct	10^{-2}	0.663	**0.611**	0.521	0.601
Stacked	10^{-2}	0.650	**0.573**	0.545	0.570

3.4 Evaluation of DIP and MCPN

The proposed internal learning methods are evaluated on two common tasks related to remote sensing applications: (a) image inpainting and (b) super-resolution. Furthermore, to demonstrate their applicability beyond the domain of satellite images, tests on image completion with other multi-modal image translation datasets are carried out.

3.4.1 Satellite Image Inpainting

For the inpainting, a test dataset comprising of 340 inpainting samples is used, which has been created using a framework described [174]. The dataset contains pairs of temporally proximate Sentinel-1 and Sentinel-2 images for a period of 2 years. More specifically, the clear sky images from Scotland in year 2020 are used as targets for the inpainting task, and the clear sky images from 2019 are averaged and used as historical informing prior. The dataset also contains cloud masks supplied with Sentinel-2 data, which were obtained for dates when cloud coverage was between 10% and 50%. These 17 masks are combined with 20 cloud-free images from 2020 to yield the resulting

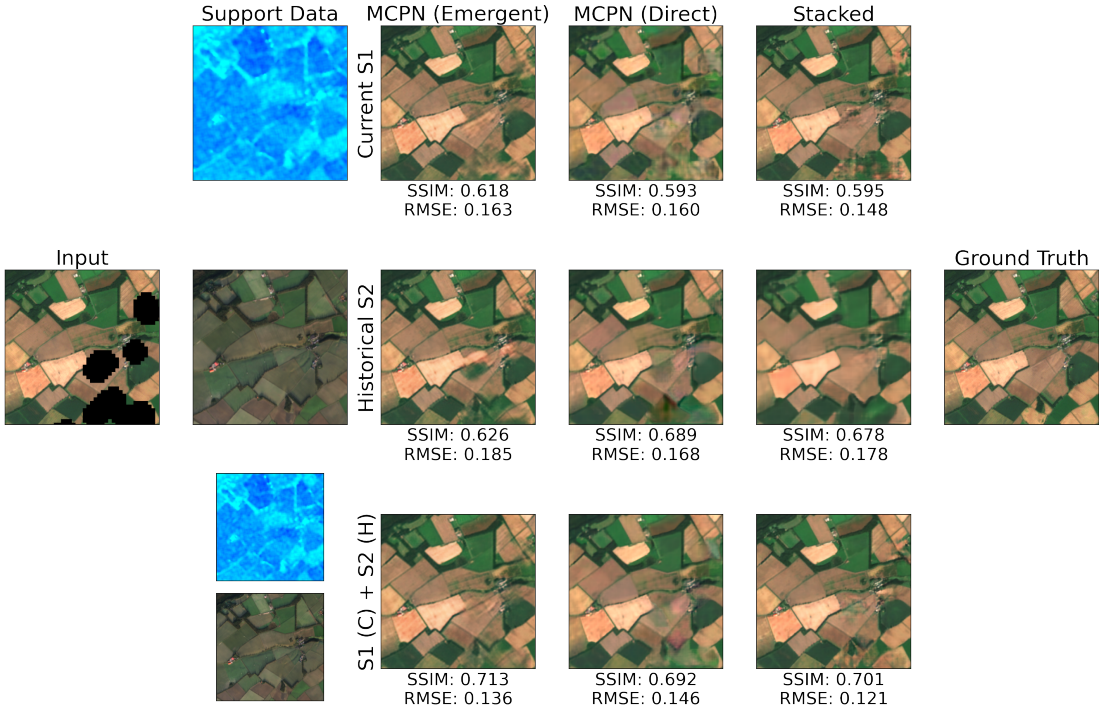


Figure 3.11: Comparison of reconstructed Sentinel-2 image for 3 types of support data (one per row) and the three tested methods

340 inference samples. The shape of each image is 256 by 256 pixels, and the same pre-processing routine as in [77] is followed.

Similarly to the experiments in Section 3.3, where convergence dynamics were studied, all models are optimised for 20,000 steps in order to compare several convergence detection methods. The obtained peak SSIM level for the inpainted region is shown in Table 3.5. This value corresponds to the highest quality of the inpainting during the full optimisation process of 20,000 steps. In order to identify this value, access to ground truth is needed, so it is crucial to note that the value is reported for analytic purposes. Although this is infeasible when deployed on new samples, it provides an indication of the upper-bound performance of each method within the test dataset.

For the challenging case where Sentinel-1 is the informing signal, the Emergent variant of MCPN offers higher inpainted SSIM, compared to the other two methods. It can be observed that the whole image SSIM is drastically lower for the Stacked compared to MCPN Emergent. This is primarily caused by the fact that the extracted

Table 3.5: Results for the Scotland Dataset: Peak Performance. The reported are the maximum metrics achieved throughout the optimisation process of 20,000 steps. Naturally, this requires the knowledge of the ground truth in order to identify the optimal image, so the numbers can be treated as a limit of the achievable performance, rather than a practical measure.

Dataset			MCPN (Emergent Core)	MCPN (Direct Core)	Stacked
Current Sentinel-1	Whole	SSIM ↑	**0.854** ± 0.041	0.760 ± 0.052	0.743 ± 0.053
		RMSE ↓	**0.079** ± 0.053	0.088 ± 0.029	0.092 ± 0.041
	Inpainting	SSIM ↑	**0.665** ± 0.082	0.657 ± 0.072	0.661 ± 0.077
		RMSE ↓	0.137 ± 0.090	**0.130** ± 0.053	0.131 ± 0.063
Historical Sentinel-2	Whole	SSIM ↑	**0.879** ± 0.044	0.853 ± 0.069	0.879 ± 0.062
		RMSE ↓	0.081 ± 0.062	0.072 ± 0.026	**0.071** ± 0.046
	Inpainting	SSIM ↑	0.719 ± 0.113	0.714 ± 0.068	**0.738** ± 0.090
		RMSE ↓	0.142 ± 0.108	**0.120** ± 0.039	0.120 ± 0.069
Current Sentinel-1 + Historical Sentinel-2	Whole	SSIM ↑	**0.876** ± 0.036	0.838 ± 0.056	0.869 ± 0.065
		RMSE ↓	0.071 ± 0.055	0.071 ± 0.030	**0.066** ± 0.038
	Inpainting	SSIM ↑	**0.743** ± 0.098	0.694 ± 0.064	0.741 ± 0.083
		RMSE ↓	0.121 ± 0.101	0.117 ± 0.050	**0.111** ± 0.059

images yielding maximised SSIM for the inpainting region are often premature in the case of the Stacked approach and the MCPN direct. For the historical Sentinel-2 case, the Stacked method achieves the highest SSIM, which could be attributed to the less severe domain shift between the informing and synthesised signals. For the case of combined Sentinel-1 and Sentinel-2 informing sources, the Emergent variant results in higher SSIM for both the inpainted region and the whole image. In terms of peak performance, the direct variant of MCPN is not as performant as the other methods, which could potentially be attributed to the bottleneck aspect of the architecture. However, as described in Section 3.4.2 this MCPN variant is beneficial for the image super-resolution task.

In practice, one of the convergence detection methods described in Section 3.3, must be employed; namely constant number of steps, patch consistency metric, or RMSE of the known pixels. It has been found that the optimisation dynamics of each synthesis method are quite different and hence different convergence detection techniques are appropriate. Experimentally it was discovered that for this dataset, the Known Region RMSE metric works best for the Emergent variant of MCPN, while the constant of 4,000 steps is most beneficial for the Direct MCPN and the baseline Stacked approach.

Table 3.6: Results for the Scotland Dataset: Achievable Performance with Optimal Convergence Detection. The samples for the experiment were selected based on computable proxy metrics, and hence, the same can be done for any new samples, without the knowledge of the ground truth.

Dataset			MCPN (Emergent Core) Known RMSE	MCPN (Direct Core) 4,000 steps	Stacked 4,000 steps
Current Sentinel-1	Whole	SSIM ↑	**0.859** ± 0.041	0.824 ± 0.045	0.837 ± 0.063
		RMSE ↓	**0.079** ± 0.048	0.081 ± 0.033	0.086 ± 0.050
	Inpainting	SSIM ↑	**0.638** ± 0.081	0.601 ± 0.068	0.576 ± 0.080
		RMSE ↓	0.141 ± 0.082	**0.140** ± 0.055	0.149 ± 0.071
Historical Sentinel-2	Whole	SSIM ↑	**0.880** ± 0.043	0.864 ± 0.041	0.875 ± 0.063
		RMSE ↓	0.079 ± 0.051	**0.075** ± 0.028	0.086 ± 0.059
	Inpainting	SSIM ↑	0.698 ± 0.107	0.692 ± 0.075	**0.703** ± 0.105
		RMSE ↓	0.142 ± 0.089	**0.131** ± 0.044	0.147 ± 0.089
Current Sentinel-1 + Historical Sentinel-2	Whole	SSIM ↑	**0.882** ± 0.036	0.861 ± 0.039	0.879 ± 0.055
		RMSE ↓	**0.069** ± 0.048	0.071 ± 0.032	0.074 ± 0.049
	Inpainting	SSIM ↑	**0.735** ± 0.096	0.679 ± 0.070	0.713 ± 0.100
		RMSE ↓	**0.120** ± 0.087	0.124 ± 0.054	0.128 ± 0.076

Hence, the performance resulting from these choices is contained in Table 3.6, indicating the quality of synthesis that can be realistically achieved.

The highest quality of inpainting (as well as reconstruction of the known region) is achieved by employing the Emergent MCPN framework for both current Sentinel-1 and historical Sentinel-2 images. Furthermore, the introduction of historical data from the same modality brings significantly higher benefits compared to the current cross-modal Sentinel-1 representation. The domain shift between the informing and synthesised signals (as in the case of Sentinel-1) remains difficult to handle for the convolutional parameterisation models. However, the use of MCPN scheme offers a significant improvement in inpainting quality, where the inpainting SSIM of MCPN Emergent and Direct are 0.638 and 0.601, compared to 0.576 achieved by the Stacked approach.

To explore how the inpainting quality changes for different sizes of the synthesised region, a sweep is conducted, where 4 clear-sky images from across the year are inpainted using a square mask with a varying area. Furthermore, both inward synthesis (where the synthesised region is fully surrounded by non-masked pixels) and outward synthesis (where the synthesised region is not surrounded by non-masked pixels) are ex-

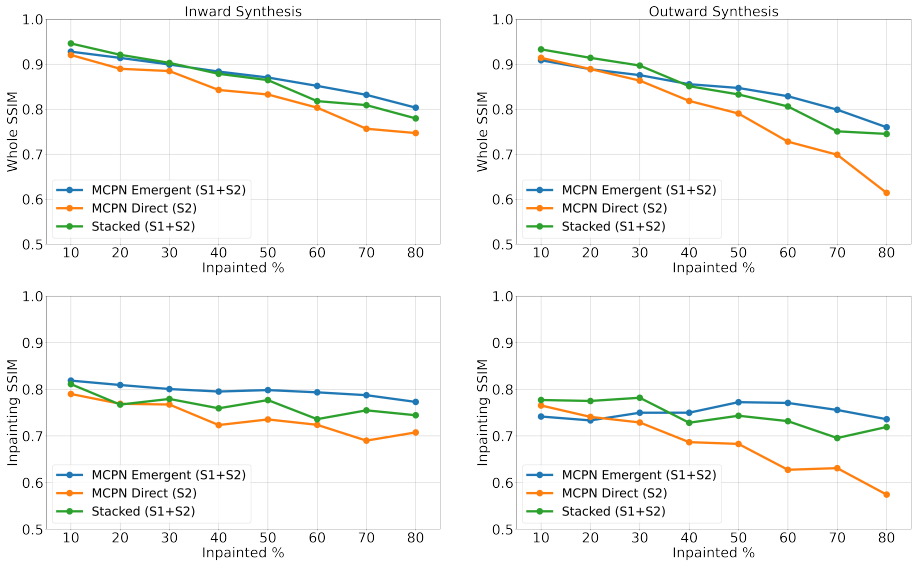


Figure 3.12: Traces indicating the changes of SSIM values for different percentage of missing pixels

plored by applying an inverse mask. The results of the sweep are shown in Figure 3.12, where the left column corresponds to inward synthesis and the right column to outward synthesis. The two metrics of whole-image SSIM and inpainting SSIM are recorded for all three synthesis methods, each using the supporting data that provides the highest performance in Table 3.6 (S2 for MCPN Direct, and S1+S2 for MCPN Emergent and Stacked).

The Emergent variant of MCPN (blue line) is leading significantly for all metrics if the missing region covers 40% of the image or more. The Direct variant of MCPN is outperformed by the Stacked method, which is consistent with the performance reported in Table 3.6 (0.692 Inpainting SSIM for MCPN Direct and 0.713 for Stacked).

3.4.2 Guided Satellite Image Super-Resolution

The two MCPN variants as well as the Stacked baseline can all be readily used to perform a super-resolution operation when high-resolution guidance coming from another signal is available. This can be achieved by employing a downsampling operation to the output of the target domain head and backpropagating gradients from a low-resolution source through it. Apart from this operation, the architectures require no

changes. Additional informing sources (such as the historical optical mean), inherently in high resolution, can be synthesised as standard images along with the super-resolved image. This fusion of sources at different resolutions can help with producing structurally coherent super-resolution output. In all presented experiments, the bilinear downsampling operation is used for the low-resolution sample. It is worth noting previous literature addressing a similar problem, and commonly referring to it as guided super-resolution [175–180]. However, most of the previous work [177, 179, 180] focuses on the task of super-resolving a single-channel depth image, based on a corresponding three-channel RGB image of higher resolution. This makes the application of many existing models to new problem settings difficult. Furthermore, MCPN constitutes a fully unsupervised framework, where no pretraining is carried out. In that regard, the PixTransform approach [179] is particularly appropriate as a baseline since it also operates without supervised training. With minimal changes applied, to accommodate for 3 channels in the super-resolved image (rather than 1, as in the depth image), it has been used for comparison in the conducted experiments. Furthermore, a common, externally-trained, baseline of EDSR [181] is tested as well (in this case, the low-resolution image is super-resolved without any guide image).

With this adjustment, all three proposed synthesis methods are employed to upsample an inherently low-resolution source. Here, Band 9 Sentinel-2 with SWIR data, with a resolution of 60 metres, is super-resolved by using the current RGB bands (Bands 4, 3, and 2) with 10 m resolution as the informing signal. The results are shown in Figure 3.13, along with the two employed baselines of PixTransform [179] and EDSR [181]. In this case, the target upscaling factor is close to 6, and for EDSR it is achieved by passing the image through the model with factor 2 followed by the model with factor 4. The result of these two consecutive EDSR passes yields an 8 times larger image, which is then interpolated down to 256×256px resolution with a bilinear operation. Similarly, since the PixTransform tool requires the upscaling factor to be a whole integer, the image is first upsampled to 64×64 pixels and then supplied as the low-resolution source. In the case of MCPN, any upscaling factor can be achieved by substituting an exact downsampling operation into the process, making it more flexible than the com-

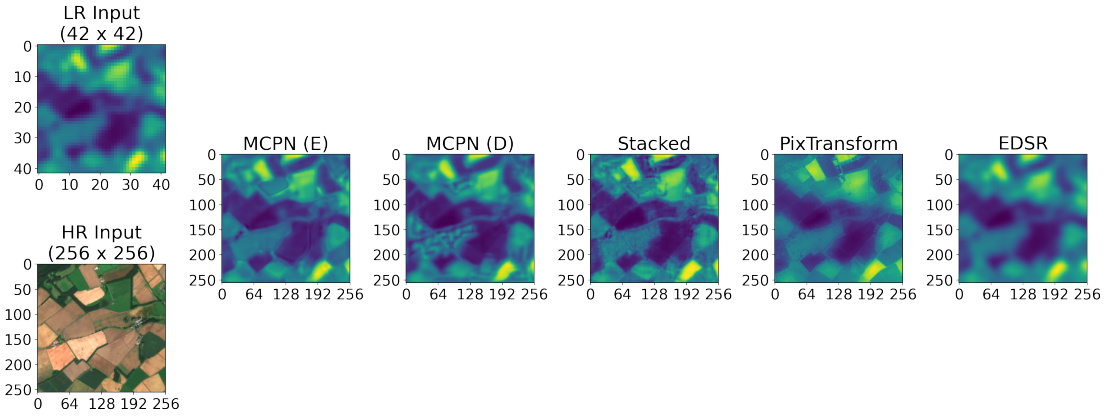


Figure 3.13: With the support of the image from the RGB bands (10 m resolution), the low-resolution SWIR image (Band 9 with 60 m resolution) is upsampled. This is an exploratory result since no ground truth exists for a high-resolution SWIR source

pared baselines. Since high-resolution ground truth for Band 9 of Sentinel-2 does not exist, it is challenging to compare these results beyond visual impression. The EDSR method results in a significant perceived blur while the convolutional parameterisation methods increase the fidelity of the image. The output of the Stacked baseline and the Emergent Core MCPN appears to contain more fine details propagated from the RGB image compared to the Direct Core MCPN. The PixTransform output appears to produce high-quality fine details compared to the other methods, but it also appears to yield reduced contrast in some parts of the image.

To conduct a quantitative evaluation, pairs of high-resolution images (standard resolution of Sentinel-2) and their corresponding downsampled versions are used to provide ground truth. The experiment has been conducted on 20 clear-sky RGB images from Scotland from the year 2020 with the supporting information of the historical average from the year 2019 (similar to the inpainting example). Each of the 20 images was subject to bilinear downsampling to compute the low-resolution input. The super-resolution performance achieved by each method is shown in Table 3.7, with the SSIM and RMSE values for three upscaling factors of 16, 8, and 4.

MCPN Direct approach achieves superior performance for the task of super-resolution, consistently outperforming all other methods for all scaling factors. An example of the super-resolved outputs and corresponding ground truth is shown in Figure 3.14, where

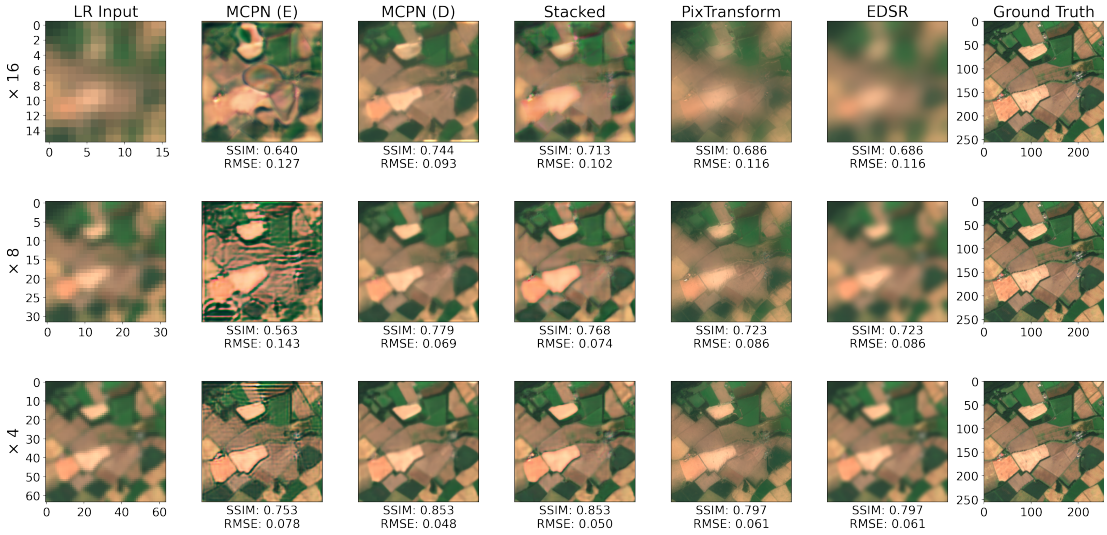


Figure 3.14: Example of Super-Resolution performance for several upscaling factors: ×16 (top row), ×8 (middle row), ×4 (bottom row). The informing high-resolution source used for the experiment was the historical optical mean from the previous year

Table 3.7: Results for the Super-Resolution Task (Historical High-Resolution Optical Reference Used for Super-Resolving Current Downsampled Optical (Achievable Performance)

Factor		MCPN (Emergent Core) 4,000 steps	MCPN (Direct Core) Known RMSE	Stacked 4,000 steps	PixTransform [179]	EDSR [181]
×16	SSIM ↑	0.487 ± 0.137	**0.733** ± 0.068	0.719 ± 0.072	0.718 ± 0.060	0.699 ± 0.055
	RMSE ↓	0.184 ± 0.057	**0.085** ± 0.047	0.094 ± 0.060	0.094 ± 0.046	0.098 ± 0.045
×8	SSIM ↑	0.584 ± 0.168	**0.782** ± 0.049	0.771 ± 0.085	0.758 ± 0.052	0.727 ± 0.050
	RMSE ↓	0.135 ± 0.067	**0.064** ± 0.029	0.076 ± 0.061	0.076 ± 0.039	0.080 ± 0.040
×4	SSIM ↑	0.685 ± 0.137	**0.847** ± 0.029	0.825 ± 0.084	0.815 ± 0.044	0.789 ± 0.038
	RMSE ↓	0.104 ± 0.068	**0.047** ± 0.017	0.062 ± 0.057	0.061 ± 0.031	0.061 ± 0.031

MCPN Direct produces the highest quality output, especially for the larger factors, while MCPN Emergent introduces more artefacts to the super-resolved image since the correspondence between the two domain signals is not as constrained as in the case of the Direct Core variant. The PixTransform method appears to do well with reconstructing fine details in the scene (such as sharp region borders), but at the same time, it creates a considerable colour distortion, which ultimately leads to performance inferior to MCPN Direct Core.

3.4.3 Other Multi-Modal Image Translation Tasks

The same techniques of encoding spatially aligned images using convolutional networks can be applied to other tasks than satellite images, as long as spatially-aligned multi-modal data is available. As an example of that capability, image inpainting is performed on 4 other datasets that contain aligned multi-domain data: Facades, Maps, Night-to-Day, and CityScapes [182]. The task involves filling the square area in the middle of each image within a border of 50 pixels around the image periphery (this translates to about 37% of fill area for 256×256px images).

The four datasets representing other types of tasks can be categorised into those containing a shallow descriptive guide, such as a segmentation mask (Facades, Maps, and CityScapes), and those containing a rich natural image guide (Night-to-Day). Depending on this aspect, a different convergence detection technique may be appropriate, and hence, the best performing one is applied on a per-dataset and per-synthesis-method basis, as shown in Table 3.8 by indicating § as the 4,000 steps technique, † as Known Region RMSE, and ‡ as Patch Consistency.

Results in Table 3.8 demonstrate that the MCPN variants can achieve superior performance to the Stacked baseline, for all tasks. For the datasets of Maps, Night-to-Day, and Cityscapes, the MCPN Emergent variant outperforms both other methods. For the task of inpainting Facade images based on a segmentation map, the Direct variant of MCPN exhibits higher performance than the Emergent variant. The Stacked baseline yields the lowest inpainting SSIM across all datasets, with the exception of the Night-to-Day, where it performs better than the Direct Core MCPN.

Selected results are shown in Figure 3.15. The tendency of the Stacked method to produce inpaintings inconsistent with the rest of the image is apparent, contributing to the highest errors associated with that technique. The Emergent MCPN produces higher structural distortions compared to the Direct MCPN.

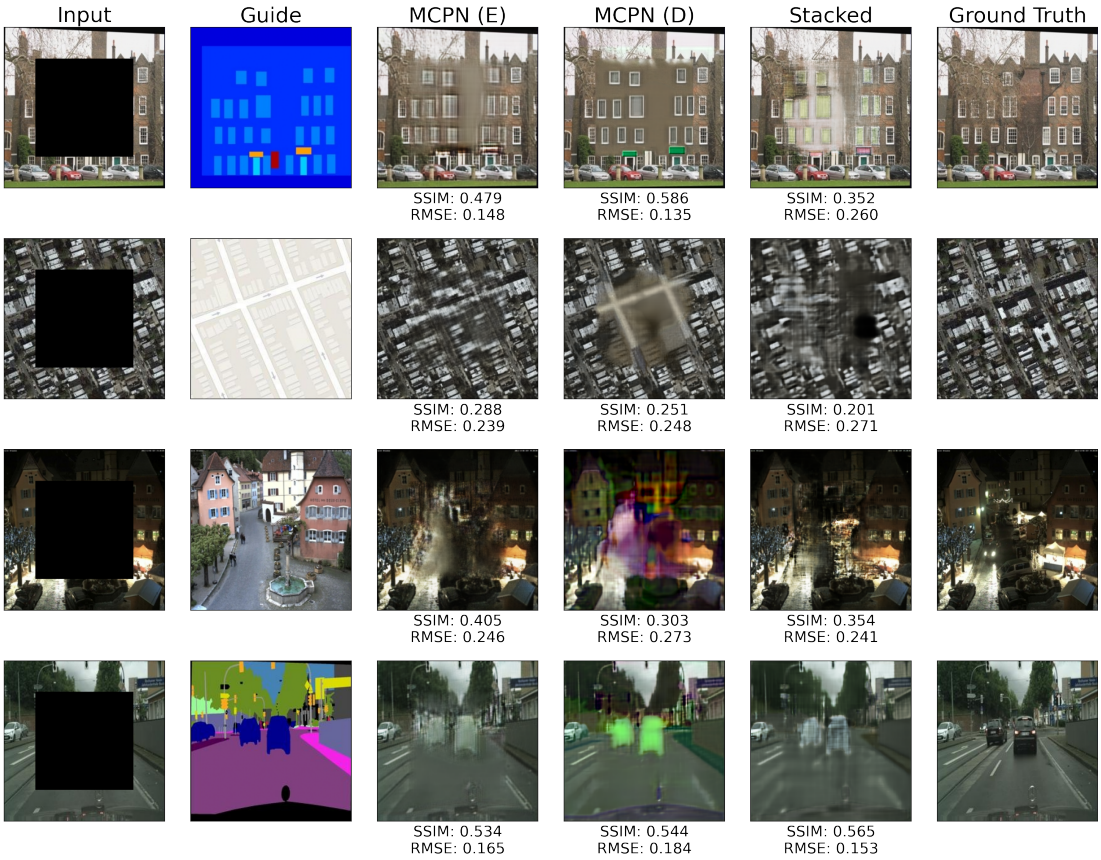


Figure 3.15: Example output for the three tested inpainting methods for each experiment dataset (from top to bottom: Facades, Map-to-Aerial, Night-to-Day, Cityscapes). The reported metric values correspond to the inpainted area only

Table 3.8: Inpainting Results for the four spatially-aligned multi-domain datasets. Mean values along with corresponding standard deviations are reported. Metrics for both whole image comparison (Whole) and inpainting comparison (Inpainting) are shown. (Achievable Performance). § - 4,000 steps, † - Known Reconstruction RMSE, ‡ - Patch Consistency

Dataset			MCPN (Emergent Core)	MCPN (Direct Core)	Stacked
Facades (Segmentation → Building)	Whole	SSIM ↑	0.763 ± 0.044§	**0.784** ± 0.058†	0.720 ± 0.100†
		RMSE ↓	**0.108** ± 0.031§	0.113 ± 0.040†	0.118 ± 0.038†
	Inpainting	SSIM ↑	0.476 ± 0.109§	**0.505** ± 0.144†	0.453 ± 0.130†
		RMSE ↓	**0.172** ± 0.051§	0.183 ± 0.067†	0.184 ± 0.061†
Maps (Map → Aerial)	Whole	SSIM ↑	**0.791** ± 0.070†	0.768 ± 0.074§	0.744 ± 0.076§
		RMSE ↓	0.085 ± 0.031†	**0.085** ± 0.030§	0.113 ± 0.038§
	Inpainting	SSIM ↑	**0.512** ± 0.174†	0.510 ± 0.175§	0.404 ± 0.169§
		RMSE ↓	0.137 ± 0.051†	**0.134** ± 0.050§	0.183 ± 0.061§
Night-to-Day (Day → Night)	Whole	SSIM ↑	**0.870** ± 0.068†	0.769 ± 0.093‡	0.828 ± 0.103‡
		RMSE ↓	**0.075** ± 0.041†	0.100 ± 0.038‡	0.096 ± 0.055‡
	Inpainting	SSIM ↑	**0.709** ± 0.167†	0.576 ± 0.148‡	0.644 ± 0.178‡
		RMSE ↓	0.121 ± 0.067†	**0.157** ± 0.060‡	0.150 ± 0.076‡
Cityscapes (Segmentation → Street)	Whole	SSIM ↑	**0.822** ± 0.031§	0.793 ± 0.041§	0.802 ± 0.047§
		RMSE ↓	0.093 ± 0.030§	**0.092** ± 0.023§	0.093 ± 0.028§
	Inpainting	SSIM ↑	**0.613** ± 0.077§	0.610 ± 0.071§	0.608 ± 0.077§
		RMSE ↓	0.150 ± 0.050§	**0.143** ± 0.037§	0.147 ± 0.047§

3.5 Summary

The internal learning methods proposed in this chapter demonstrate the capability of parameterising spatially aligned signals from multiple domains using convolutional neural network architectures, harnessing the powerful priors induced by the deep convolutional topology. This capability enables an internal solution to several image inverse tasks, such as image inpainting or super-resolution, and easy application of additional image-based guides. By definition, an MCPN model can readily be applied for any number of domains, and it has been shown that it can work with domain shifts as large as between image segmentation maps and corresponding natural images.

The presented results identify several trade-offs relating to the application of internal learning techniques based on a CNN architecture. First, the motivation for exploring this type of solutions was that the same method can be applied to treat images of various channel shapes. This property is particularly beneficial when working with satellite images, which have varying numbers of channels and different value distributions depending on the type of pre-processing applied beforehand.

The techniques presented in the chapter have been shown to be capable of several different tasks, such as image completion or super-resolution, and can flexibly accept various guidance signals at inference time, which is beneficial for satellite image processing applications, where different sources of support data, such as SAR or historical captures may be available.

Chapter 4

Learning from Language

The previous chapter has explored the potential of learning from a single sample with no pre-training on the specific task of cloud removal, by relying on the priors extracted from the inference sample. In this chapter, another source of priors is explored, embarking on an attempt to learn from beyond the domain of the downstream task using language-based models. This may appear familiar, as it follows the same principle as the wide set of techniques commonly referred to as *transfer learning* [183]. However, transfer learning [183] was largely popularised within the machine learning field long before the rise of large-scale language-based models. As an example, a common transfer learning practice is to initialise a network backbone with parameters trained on the ImageNet classification task [184–186] to reduce the burden of learning the task of interest. However, that usually still involves quite costly fine-tuning. In this work, a special focus is put on use cases where the additional training costs are either absent or trivially low (such as fine-tuning in a matter of several minutes) owing to the emerging capabilities of large language-based models trained on a mix of text and visual data [94, 187] produced recently.

Instead of fine-tuning a deep neural network on the task of cloud removal, it is explored how to directly use general language-based models for processing clouds in satellite imagery. More specifically, two techniques are proposed in this chapter, (i) employment of the pre-trained CLIP model for cloud presence detection, and (ii) employment of a pre-trained text-to-image StableDiffusion model for cloud removal. It is

shown that transferring the knowledge gained in the wider domain of combined text and images can lead to a non-trivial performance in a zero-shot setting, where the exact type of operation is defined upon inference.

So far, the exploration of the applications of language-based models to satellite image processing has been quite limited. At the time of writing, three published works on the topic can be identified [188–190]. The first two [188, 189] were focused on visual question answering, however, these methods focused on the setting, where custom models are trained from scratch on task-specific datasets. On the other hand, the work described in [190] involved fine-tuning a pre-trained CLIP model on satellite images.

This work aims to fill the gap by exploring the potential uses of language-based models for processing clouds in satellite images. More specifically, the aim is to avoid the need for extensive fine-tuning and instead rely on the emergent capabilities of existing models. This approach can potentially lead to multiple benefits, ranging from a flexible sensor-agnostic formulation, a powerful set of biases drawn from large-scale image data, and reduced risks of overfitting that can normally occur during conventional dataset-based training.

This chapter brings two major contributions in this context. First, the representations learned by the open-source CLIP model are used for detecting the presence of clouds, with several approaches tested on datasets containing both Sentinel-2 and Landsat images. Second, a brand new set of methods is proposed and tested for conditional MSI inpainting of satellite images based on StableDiffusion pre-trained models.

4.1 Detecting Presence of Clouds with CLIP

As the first use-case for language-based models, it is demonstrated that models trained on general datasets of text and images have learned representations that are any useful for processing clouds in satellite images, and the level of achievable performance is reported.

This is motivated by two contexts. First, the understanding of the phenomenon of clouds in satellite images by general language-based visual models determines whether they are suitable for more complex tasks, such as cloud removal. Secondly, successful

detection of clouds in satellite images is useful on its own, because the imagery affected by clouds is not usable for many tasks. Hence, it is common to filter those data instances out and exclude them from the analysis. In order to successfully filter this undesired portion of data, cloud detection mechanisms are often used.

4.1.1 Summary of the CLIP Model

The recent years, especially after the introduction of the transformer architecture [117], have brought a significant rise in deep learning solutions based on large-scale training on text data. Shortly after, natural language supervision has been applied to visual data, and in [187], language has been used to support the use-case of zero-shot classification of images.

In the previous convention of deep learning classifiers, a model would be trained to produce confidence for a predetermined number of classes, following an approach similar to the seminal work of AlexNet [83]. In such a case, a convolutional architecture was designed to produce 1,000-dimensional output, where 1,000 corresponded to the number of classes in the ImageNet task. As can be expected, this approach leads to fairly rigid classifiers, meaning that there is little space for applying those models to new classification tasks. In such a context, the common approach to transferring knowledge learned from one task to another is to finetune on the new task directly. This, however, can bring certain disadvantages as it requires a new dataset to be provided, specific for the new task, and it introduces the cost and inconvenience of executing and designing another training process.

For those reasons, the inclusion of text-based conditioning in the models was an important step in the development of versatile deep learning classifiers. Text input data allows adjusting the behaviour of the model in a zero-shot setting. Instead of having the model choose between the 1,000 static classes of ImageNet, the user of the CLIP model can ask to assign an image to one of 10 new classes specified by them via text during inference.

In the case of CLIP, this assignment, or classification, is carried out by measuring the degree of alignment between the embeddings encoded from text and the embed-

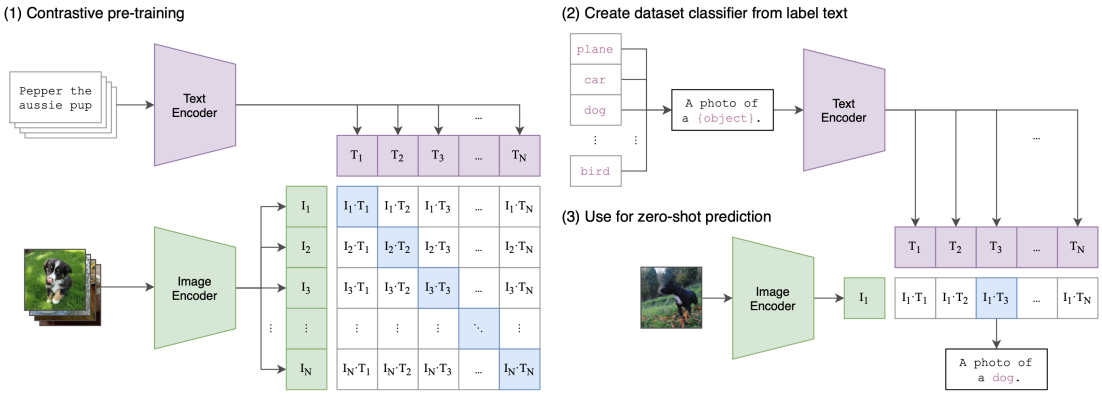


Figure 4.1: Overview of Contrastive Language-Image Pre-training (CLIP) method. Figure extracted from [187]

dings encoded from the image. As shown in Figure 4.1, the embeddings are computed using two separate encoder modules, one for text data and another one for images. The alignment can be measured by computing a measure of alignment between the embeddings, such as cosine similarity, as shown in the right-hand side of Figure 4.1.

The capabilities of the CLIP model are largely owed to two key factors, the dataset it has been trained on and the method of training. The dataset, named WebImageText (WIT), was built specifically with the intention of training the model on large-scale data since the datasets existing at the time did not have sufficient volume. The WIT dataset contains 400 million pairs of text and images downloaded from the web and covering a very wide range of concepts, where queries were based on the presence of a term in English Wikipedia (at least 100 occurrences of a term required). The method of training was another crucial component in CLIP's success. Specifically, it relies on maximizing the cosine similarity between text and image embeddings from corresponding pairs in a batch and minimizing the similarities between the text and image embeddings across pairs. This is demonstrated in the left portion of Figure 4.1, where a batch of N text-image pairs yields N^2 similarity values, where only N of them (the diagonal blue squares) are maximised as they correspond to the similarity within individual matched pairs from the dataset.

The capability of the CLIP model to recognise cloud-affected satellite images can be expected to rely on at least one key factor, that is the data the model has been trained

on. At the time of writing this, the authors of the CLIP model have not disclosed further details about the training dataset other than the short description in the paper of how the dataset was built. However, since the trained model is open-source it is possible to measure the capability of the model to process clouds in satellite imagery in an empirical manner.

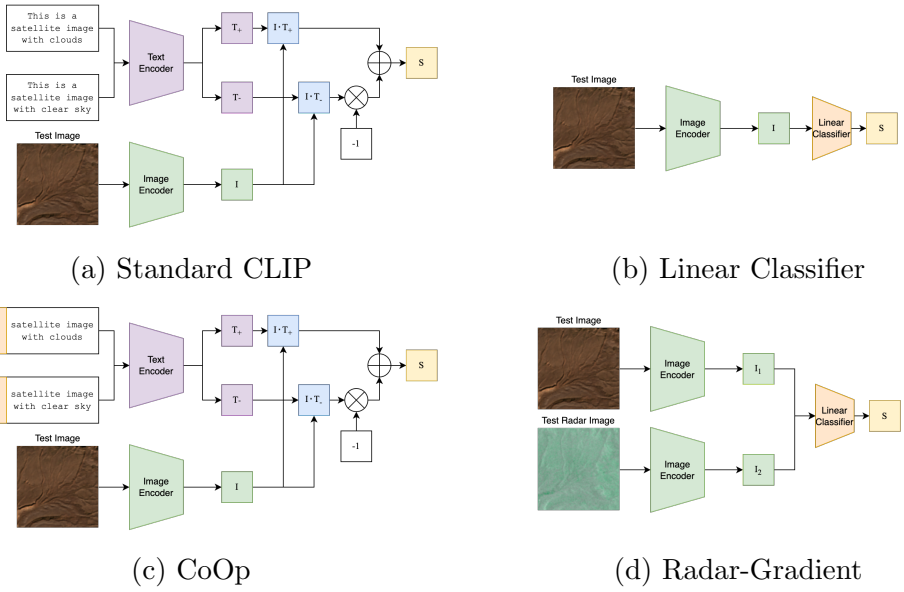
4.1.2 Proposed Solutions for Cloud Presence Detection

The approach to applying CLIP to satellite MSI images is not immediately obvious; the CLIP model operates on RGB images, while a typical solution to detect clouds in satellite imagery involves more than the RGB visible bands, such as infrared, and is often sensor-specific. Some past works have explored the potential of an RGB-only cloud detection model [26], but the task is considered significantly more challenging.

There exist several available CLIP models published along with the original manuscript, each with a different image encoder. The official CLIP python package [191] contains 5 ResNet-based models ('RN50', 'RN101', 'RN50x4', 'RN50x16', 'RN50x64') and 4 ViT-based models ('ViT-B/32', 'ViT-B/16', 'ViT-L/14', 'ViT-L/14@336px'), corresponding to different sizes of models tested in the original paper. This variety in models corresponds to the two explored architecture types (ResNet and vision transformer) and different model sizes with different levels of the trade-off between compute cost and model capacity. For the experiments conducted with satellite images, the most efficient vision transformer model in standard resolution is used ('ViT-B/32') to minimise inference cost. Since the model expects a constant input shape of 224 by 224 pixels, bilinear interpolation with anti-aliasing is applied to the input before passing through the network.

Four methods of employing the pre-trained CLIP model are considered here, all shown in Figure 4.2. The first method (Figure 4.2(a)) is fully zero-shot and relies on the embeddings encoded from two text prompts, one corresponding to the presence of the clouds in the image and the other one to the absence. The text prompts were arbitrarily selected as *"This is a satellite image with clouds"* and *"This is a satellite image with clear sky"* with no attempt to improve them. Following the approach

described in the CLIP paper [187], cosine similarity between the test image embedding and the two text embeddings is computed, and the text embedding with the higher similarity determines the assigned class (cloud versus cloud-free).



(a) Standard CLIP

(b) Linear Classifier

(c) CoOp

(d) Radar-Gradient

Figure 4.2: Explored methods of CLIP-based cloud presence detection. Method (a) is the standard approach, while (b)-(d) techniques are based on a low-cost fine-tuning on a small dataset.

The other three methods rely on a minor fine-tuning stage, which involves only 1,000 optimisation steps with a batch size of 10 (on samples outside of the test dataset), which takes no more than a few minutes on a single consumer-grade GPU. The optimised components are displayed in orange colour in Figure 4.2.

The second method (Figure 4.2(b)) involves training a linear classifier on top of the image embeddings encoded by the CLIP image encoder. For this setting, text prompts are not required, however, it is important to recognise that the image encoder weights still come from the training process based on the similarity between text and image inputs) and the approach resembles many techniques from the domain of transfer learning [183]. Another related method (Figure 4.2(c)) is based on the context optimisation (CoOp) method [192], where additional context is prepended to the core class prompts, consisting of a fixed number (in this experiment, 16) of tunable tokens of the same dimensionality as text tokens. This context is then optimised during the

fine-tuning process. Hence, methods (b) and (c) approach the fine-tuning process from two opposite directions, where the linear classifier method attempts to optimise weights applied to features encoded by the image encoder, while the CoOp method attempts to optimise the text input before it is processed by the text encoder.

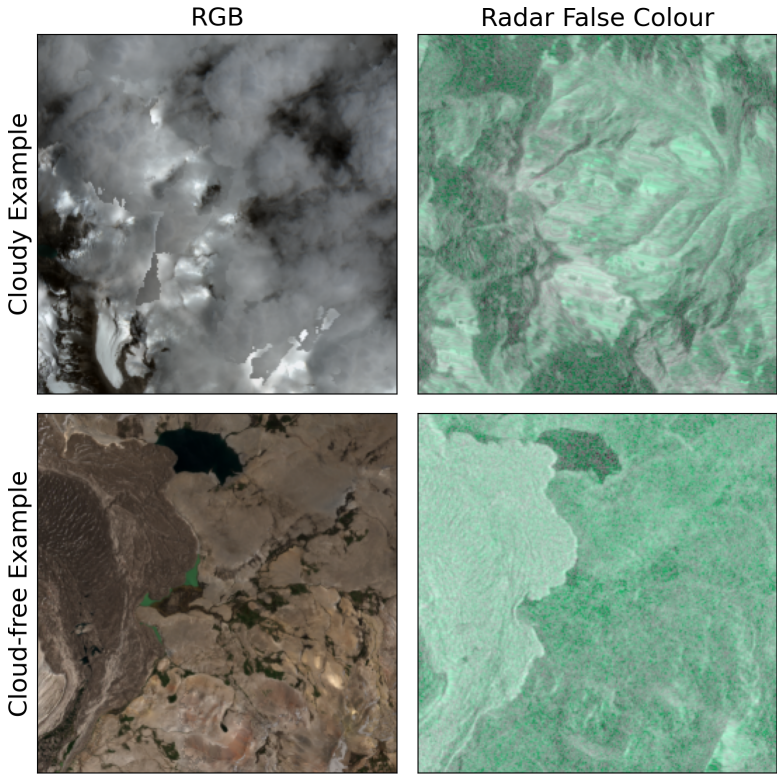


Figure 4.3: Examples from the CloudSEN12 [11] test dataset.

Finally, the fourth method shown in Figure 4.2(d) is a novel approach proposed here, where a reference signal from a separate image, in this case a radar representation is used as a source of information. This context is injected by training a linear probe classifier based on the image encodings of both RGB test image data and a false-colour composite of the SAR data (Sentinel-1 VV, VH, and mean of the two channels are encoded as 3 input channels). Interestingly, even though radar false-colour images might lie outside of the domain that CLIP was originally trained on, the performance achieved by this method indicates that the image encoder is still capable of extracting features that are useful for cloud presence detection.

The approaches are tested on two benchmark datasets: (i) CloudSEN12 [11], con-

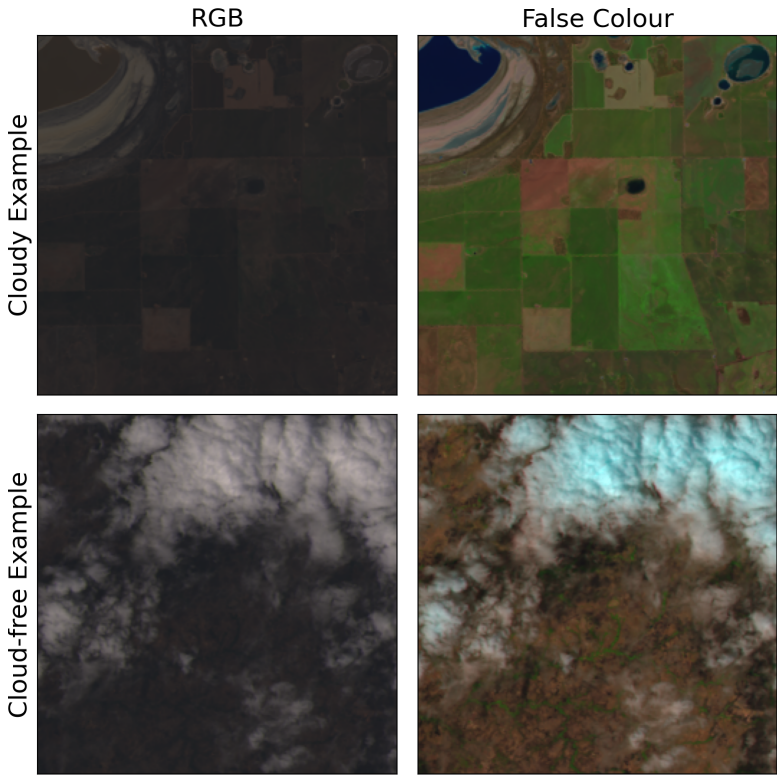


Figure 4.4: Examples from the SPARCS [193] test dataset.

taining Sentinel-2 and Sentinel-1 data (test dataset contains 195 cloud-free images and 780 cloudy images), and (ii) SPARCS [193], containing Landsat-8 imagery (containing 40 cloud-free images and 88 cloudy ones). By testing on two datasets with Sentinel-2 and Landsat-8 data, it is possible to measure the transferability of the proposed methods. Example samples from the CloudSEN12 test dataset are shown in Figures 4.3 and 4.4.

Another relevant aspect is that the annotators of the SPARCS dataset while labelling the images, have been shown false-colour images with bands B6 (SWIR), B5 (NIR), and B4 (Red) assigned to RGB channels, respectively [193]. While these images are artificial in the sense that the channels do not correspond to RGB intensities, they can be interpreted by a CLIP model. Hence, two versions of the SPARCS dataset are tested here, one with the RGB bands and one with the false-colour images observed by the annotators. Example samples (both real-colour and false-colour) from the SPARCS test dataset are shown in Figure 4.4.

The achieved performance is reported in Table 4.1 as true positive rate (TPR), the fraction of all cloudy images detected as cloudy (Equation 4.1); true negative rate (TNR), the fraction of all cloud-free images detected as cloud-free (Equation 4.2); and F1 score, a harmonic mean between the ratio of correct predictions among all cloudy samples and the ratio of correct predictions from all samples classified as cloudy (Equation 4.3).

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{4.1}$$

$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FN}} \tag{4.2}$$

$$\mathrm{F1} = \frac{2 \cdot \mathrm{TP}}{2 \cdot \mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{4.3}$$

As mentioned earlier, three types of test data are used (corresponding to three sets of three columns in the table), starting with CloudSEN12 data with RGB Sentinel-2 input, and then Landsat-8 data from the SPARCS dataset with either RGB bands or B6-B4 false colour composite bands. The rows in the table correspond to different methods explored in this work:

1. Text Prompts: zero-shot classification with CLIP using text prompts
2. Linear Probe: linear probe fine-tuning
3. CoOp: Context Optimization technique
4. Radar: linear probe appleid to image and radar-based input (which can only be applied to CloudSEN12 which provides both Sentinel-2 RGB and Sentinel-1 SAR data)

Furthermore, the variants 2. and 3. are fine-tuned on one type of 3-channel input and can be tested on another type of 3-channel data from a different sensor. This is indicated by an additional letter, where (a) is used to signify training and testing data coming from the same sensor, and (b) indicates the transfer across the sensor type.

Table 4.1: Performance of cloud presence detection techniques for the tested datasets and detection methods.

Test Dataset	CloudSEN12			SPARCS					
Modality	S2/RGB			L8/RGB			L8/B6-B4		
	TPR	TNR	F1	TPR	TNR	F1	TPR	TNR	F1
1. Text Prompts	0.929	0.638	0.919	0.922	0.737	0.907	0.900	0.737	0.895
Trained on:	S2/RGB			L8/RGB			L8/B6-B4		
2a. Linear Probe	0.924	0.975	0.957	0.856	1.000	0.922	0.822	1.000	0.902
3a. CoOp	0.936	0.980	0.964	0.878	0.921	0.919	0.822	0.974	0.897
4a. Radar	0.930	0.960	0.959	N/A	N/A	N/A	N/A	N/A	N/A
Trained on:	L8/B6-B4			S2/RGB			S2/RGB		
2b. Linear Probe	0.961	0.759	0.950	0.811	1.000	0.896	0.811	1.000	0.896
3b. CoOp	0.988	0.578	0.943	0.789	1.000	0.882	0.844	0.974	0.910

The results in the first row with the zero-shot text prompt performance indicate that the CLIP-based model combined with the employed text prompts can achieve a high performance of at least 0.9 true positive rate, which means that the model can be quite reliable at picking up the cloudy samples. However, consistently across all three test datasets, the true negative rate is considerably lower, with values of 0.638 for Sentinel-2 data and 0.737 for Landsat-8 data (regardless of the representation), which means that more cloud-free images are classified as cloudy.

The true negative rate is considerably improved by fine-tuning. For the CloudSEN12 dataset, the true negative rate increases to 0.975 for the linear probe approach (2a), 0.980 for the CoOp approach (3a) and 0.960 for the radar-based variant (4a). The true positive rate is consistently lower, meaning that some of the true positives are as a result traded off for true negatives.

For the models fine-tuned on the Landsat-8 data, a similar effect is observed, with a very high true negative rate, and the true positive rate decreasing considerably from the level achieved in the fully zero-shot setting.

The transferability is tested by applying the models from Sentinel-2 to the SPARCS dataset and the model trained on Landsat-8 B6-B4 to the Sentinel-2 images. In this case, the Sentinel-2 models appear to transfer better than the Landsat-8 models as the L8/B6-B4 model suffers a large decrease of TNR when applied to Sentinel-2 RGB data.

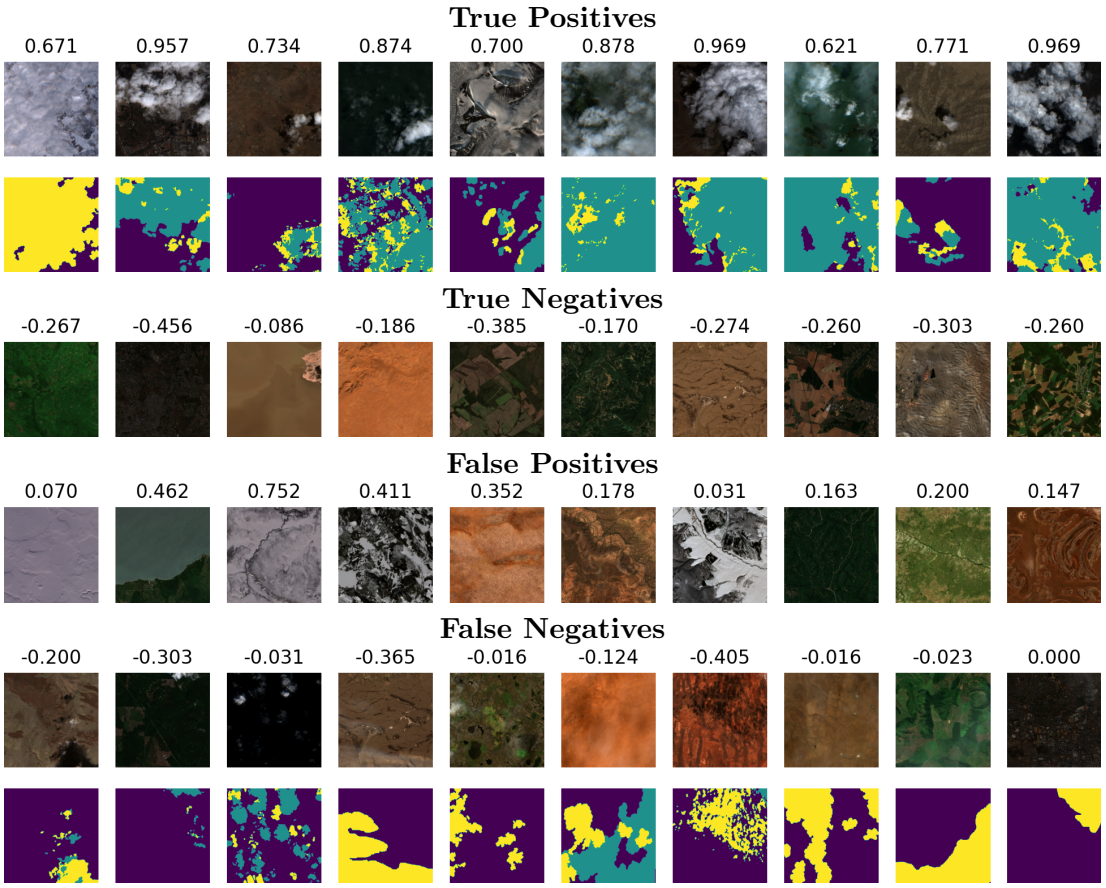


Figure 4.5: Examples of the predictions produced by the CLIP model in a zero-shot setting (no fine-tuning). The values above images correspond to the difference between cosine similarity of the positive prompt and the negative prompt (high positive value indicates overlap with the positive label).

However, a decrease in performance is observed upon transfer across modalities, which could mean that the discriminative relationships of the CLIP encodings differ to some degree depending on the sensor type and do not transfer as well, however, this would need to be confirmed by further experimentation.

Further insight into the model's behaviour can be obtained by inspecting some of the samples from the four potential prediction types (true positive, true negative, false positive, and false negative). A random 10 samples from each group are shown in Figure 4.5 for (1) the zero-shot approach based on text prompts and in Figure 4.6 for (2a) the linear probe trained on the Sentinel-2 data.

It is apparent from Figure 4.5 that the model is capable of picking up clouds, some

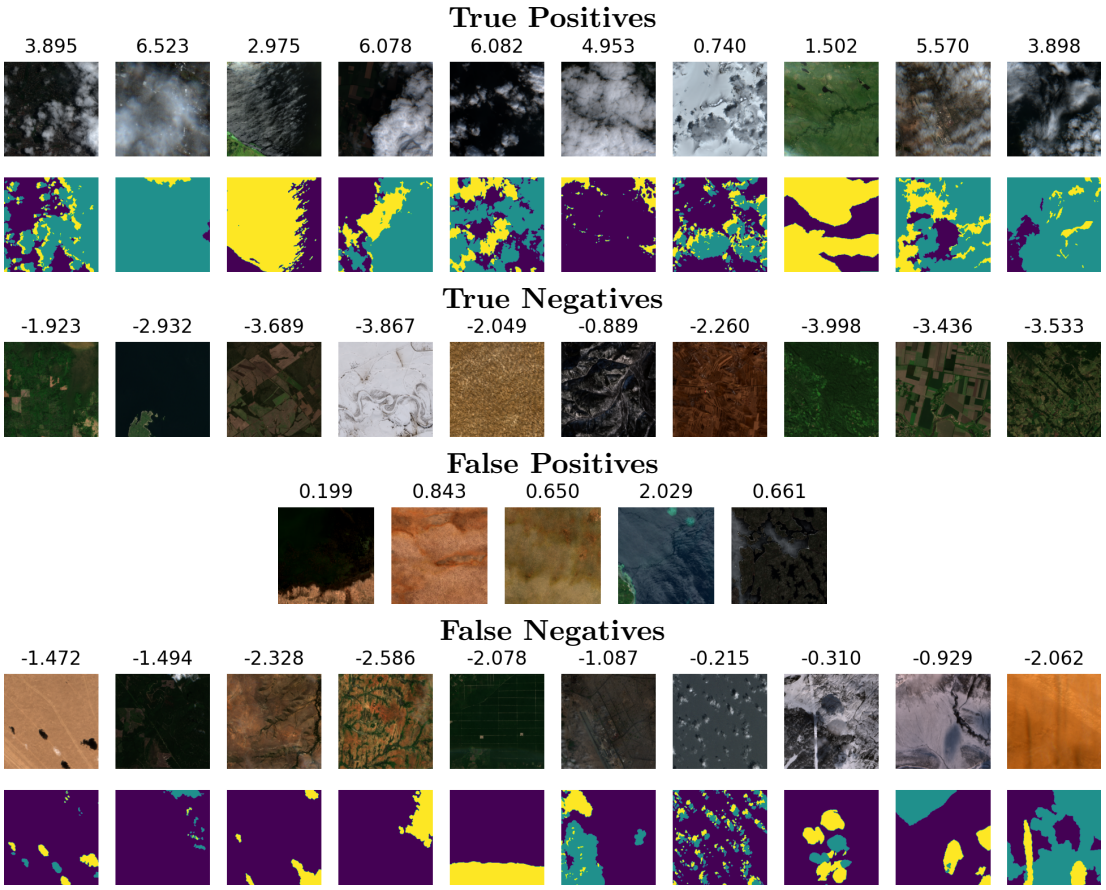


Figure 4.6: Examples of the predictions produced by the CLIP model with a linear probe trained on Sentinel-2 data. In this case, the values above images correspond to the score computed by the linear classifier (positive values correspond to the positive class).

of which are semi-transparent, such as the one in the fifth column of true positives. However, it also picks up quite a lot of false positives, samples which should mostly be trivial to the human eye or a pre-trained classifier. False negatives generally contain difficult examples of thin clouds or clouds covering a very limited local region.

A model fine-tuned on Sentinel-2 date for 1,000 gradient steps performs considerably better as shown in Figure 4.6. The number of false positives is reduced to only 5 samples, while the false negatives include predominantly visually challenging examples. This is still quite similar to the sets of samples identified for the zero-shot model, which could mean that the fine-tuning does not drastically change the types of errors made by the cloud presence detector, but rather their prevalence.

The results in Table 4.1 lead to the following conclusions. First, it has been found that the features learned from large-scale datasets containing text-image pairs represent non-trivial knowledge for processing satellite imagery, as shown in the example of detecting cloud presence. Second, while the fully zero-shot approach based on text prompts can lead to non-trivial performance, the accuracy of the classification can be increased at a low cost by a short fine-tuning stage. In that case, training a single linear layer on top of the image features appears to be the most effective technique. Lastly, it has been demonstrated that the technique can be applied across different modalities (in this case Sentinel-2 and Landsat 8), where the model fine-tuned on one data type can lead to good performance on another one.

4.2 Satellite Image Inpainting Using Text-to-Image Models

The experiments show that the CLIP zero-shot classifier pre-trained on a wide-context dataset containing text-image pairs can detect the presence of clouds in satellite images. This hints at the potential of employing general pre-trained models for processing satellite images and motivates the exploration of related solutions for synthetic tasks.

The topic of text-conditioned image synthesis has gained considerable traction in the recent years [94, 104, 105, 116, 118, 119, 194, 195], with big improvements achieved in the quality of generation. Similarly to CLIP, the solutions are often based on large-scale datasets containing pairs of text and images, leading to models encompassing rich and powerful priors. There are several generative architectures commonly used for text-to-image pipelines. One of the earliest successful approaches, DALL-E [116], was based on the transformer architecture [117] trained in an autoregressive fashion, achieved by training the image as a sequence of tokens (corresponding to compressed image regions) similar to the tokens derived from the text. This approach allowed to treat the problem of text-to-image synthesis in a manner very similar to regular text synthesis (since the joint text-image representation was expressed as a sequence of tokens), and several other methods were based on a similar principle, such as CogView [194] or

Parti [195]. However, the increased interest in text-to-image synthesis coincided with another important advancement in the field of generative modelling, namely, denoising diffusion of images [94, 104, 118, 119]. It has been shown that diffusion models have the capacity to achieve a higher quality of synthesis than GANs [105], and soon after, several manuscripts have been published where the image synthesis component was designed as a diffusion model, including works such as GLIDE [104], DALL-E 2 [118], Imagen [119], or Latent Diffusion [94].

A fundamental aspect of using text-to-image generative models for research is accessibility. In order to advance research further and develop new solutions, both inference source code and even more crucially, trained model weights need to be provided. Training large-scale models from scratch is costly, for example, the earliest checkpoint of StableDiffusion (1.1) has been trained for 431,000 steps with an effective batch size of 2048 (32 clusters with 8 GPUs with 2 gradient accumulations and 4 samples per GPU) [196]. The associated cost of training has been quoted as 600,000 USD by StabilityAI CEO, Emad Mostaque on Twitter [197]. More advanced checkpoints that were released following version 1.1 can be expected to cost several times more. With these levels of costs, it is difficult to imagine research groups and independent researchers training these models from scratch in order to enable further research. Even if the financial cost of training from scratch is not an obstacle, it is difficult to guarantee that independent training runs of the same pipeline will yield models with equivalent behaviour and knowledge, which means that the comparison of results from multiple studies is not well grounded. Lastly, even if the same training run could be reproducible and the training cost was not a limiting factor, it is important to acknowledge the environmental impact of training large-scale models. As per model card [196], the estimated carbon emissions associated with training the first checkpoint of StableDiffusion were around 11,250 kg. For these reasons, the open sharing of model weights is important for facilitating the further advancement of science. Among the models mentioned above, only LatentDiffusion [94] (also known as StableDiffusion) and GLIDE [104] provide unrestricted access to model weights, while DALL-E [116], DALL-E 2 [118], Imagen [119], and Parti [195] do not give access to the model weights nor inference scripts. This

study will be focused on pre-trained StableDiffusion models due to their high quality of synthesis as well as the open-source code base and access to model weights.

At this point, it is important to again consider the data the pre-trained models have been trained on. Unlike CLIP [187], StableDiffusion has been trained on a widely available dataset from the LAION (Large-scale Artificial Intelligence Open Network) organisation. The base dataset, LAION-5B [198], has been created by scraping a large public web archive (Common Crawl[1]) in search for image components accompanied by alt (alternative) text in HTML source code, followed by several filtering stages. As a result, 2.32 billion images with English descriptions make up the LAION-2B subset of the dataset. This dataset is expected a comparable level of prevalence of satellite images to the level encountered in the world wide web since 2008 (beginning of the Common Crawl project).

In this work, the potential of employing the StableDiffusion text-to-image model for satellite image inpainting will be explored. Furthermore, two related challenges are addressed; first, a custom pipeline is proposed that combines the Stable Diffusion inpainting model with the ControlNet method [199] in order to allow additional guidance from historical data (the standard version of the Stable Diffusion inpainting model does not accept on additional guidance signals). Second, a technique based on Deep Image Prior [139] is introduced with the aim to use the inpainting in the RGB channels for completing the same region in a larger number of MSI bands in Sentinel-2 data.

4.2.1 Background

Denoising Diffusion of Image Data

StableDiffusion is an instance of a diffusion-based generative model. The denoising diffusion process is a technique of generative modelling based on a chain of degradation operations that links the distribution of observed data x_0 with a simple prior distribution, such as a pure Gaussian. In the conventional setting, the operation of additive Gaussian noise is used as degradation [103], however, other types of degradation have also been explored [200,201]. During training, a neural network is optimised to estimate

[1] https://commoncrawl.org/the-data/get-started

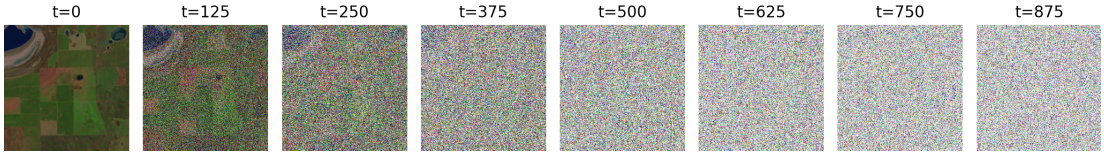


Figure 4.7: Example of a forward diffusion process chain (from left to right). The sample at $t = 0$ is a clean image, while higher values of t correspond to the number of times Gaussian noise (with a variance following a specific schedule) has been added to the sample.

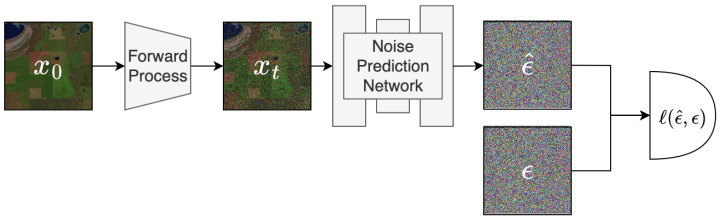


Figure 4.8: Diagram of training a Denoising Diffusion Model. At each step, a random diffusion index t is sampled (equal likelihood) and the network is trained to predict noise sample ϵ contained in the noisy sample x_t.

the parameters required to reverse the degradation process. Most commonly, this parameter is the exact sample ϵ of added normal Gaussian noise present in the degraded sample, which closely relates to the score-based interpretation of these models [202]. The degradation is formulated as the forward process $q(x_{t|t-1})$ applied to an image x_t based on a temporal index t indicating the stage of diffusion (x_0 is the beginning of the diffusion chain, which represents a clean sample). In the case of Gaussian diffusion, the forward process can be defined as

$$q(x_t|x_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t}x_{t-1}, \beta_t I), \tag{4.4}$$

where x_t indicates a sample at the diffusion stage t, and β_t is a parameter obtained from the diffusion schedule, which controls the magnitude of the Gaussian steps taken at each point of the diffusion chain. An example of a forward chain is illustrated in Figure 4.7, for different values of t ranging from 0 (clean sample) to 875 (875 steps of the forward process of Gaussian diffusion).

The corresponding reverse process is based on progressing through the same chain

in the opposite order, and, if the Gaussian steps are small enough, it can also be approximated by a Gaussian; with unknown parameters $\mu(x_t, t)$ and σ_t:

$$q(x_{t-1}|x_t) = \mathcal{N}(\mu(x_t, t), \sigma_t^2 I) \tag{4.5}$$

These parameters depend on the stage of diffusion t and the sample x_t at that stage and are quite difficult to approximate. A deep neural network is used to learn this approximation, most commonly a convolutional encoder-decoder network, such as U-Net [103].

During training, the network is trained with samples x_t, where t is sampled from a uniform distribution across all T indices in the schedule. This is illustrated in Figure 4.8, where the clean sample x_0 is used to compute a noisy sample x_t. Noise is added to the sample x_0 (a clean training sample image) according to the forward process defined in Equation 4.4. The error is computed between the ground truth ϵ and the network output $\hat{\epsilon}$.

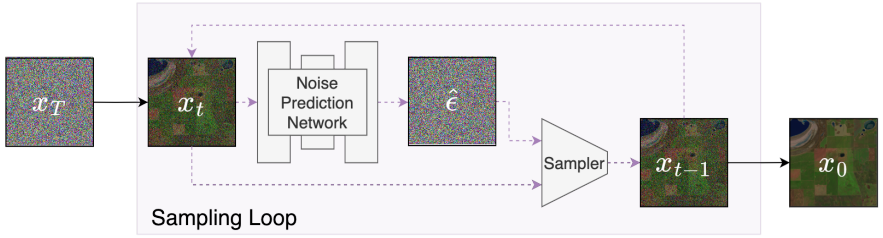


Figure 4.9: Diagram of sampling from a Denoising Diffusion Model. To generate a sample x_T is gradually transformed into x_0 by predicting the noise and stepping through the reverse diffusion chain.

During inference, a model trained to predict ϵ can be used with different sampling methods, such as the conventional DDPM (Denoising Diffusion Probabilistic Model) [103], or DDIM (Denoising Diffusion Implicit Model) [102]. As shown in Figure 4.9, the process begins with a sample from pure Gaussian noise, which is treated as the sample x_T at the end of the chain $t = T$. It then is passed to the sampling loop, where a chosen sampler approximates a chain of samples in the reverse direction, eventually reaching x_0. The presented solution employs a more recent UniPC sampler [203] for inference, as it has been demonstrated to yield excellent performance for

small numbers of inference steps.

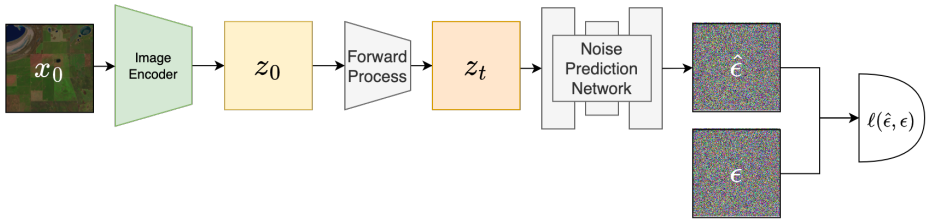


Figure 4.10: Diagram of training a latent diffusion model. In this case, encoder of the autoencoder network is used to compress an image x_0 to a latent code z_0, and the same approach as in conventional diffusion is applied to learn approximating the noise ϵ present in a noisy sample z_t.

Stable Diffusion Text-to-Image Model

Stable Diffusion is an instance of a latent diffusion model [204] focused on the text conditioning modality. Latent diffusion is a type of image diffusion aimed at high-resolution data. Apart from the core network used for the reverse process, it employs an autoencoder to compress image input into a latent space (with lower resolution and 4 latent channels) so that the denoising diffusion process is performed in a more compact domain than high-resolution images. As shown in Figure 4.10, the diffusion process is applied to a latent code z_0 computed by passing the clean sample x_0 through a pre-trained image encoder (obtained via autoencoder training, composed of an encoder and decoder with a compressed latent space [114]). The code z_0 is effectively treated as a sample, and the network is trained to predict a sample present in the noisy state z_t.

The inference is also similar (shown in Figure 4.11), as the process begins with a Gaussian sample that is then assigned to state z_T, and then subject to the sampling loop. At the end of the sampling loop, once z_0 is obtained, the latent code is fed into the image decoder to obtain a clean image sample x_0 in the original domain.

The work on latent diffusion also proposes a unified approach to conditioning the process of image generation. One notable type of conditioning, and the subject of this chapter, is text-based conditioning. In this case, the condition encoder ingests text input y, as shown in Figure 4.12, and uses the encoding $\tau_\theta(y)$ as the key input in the

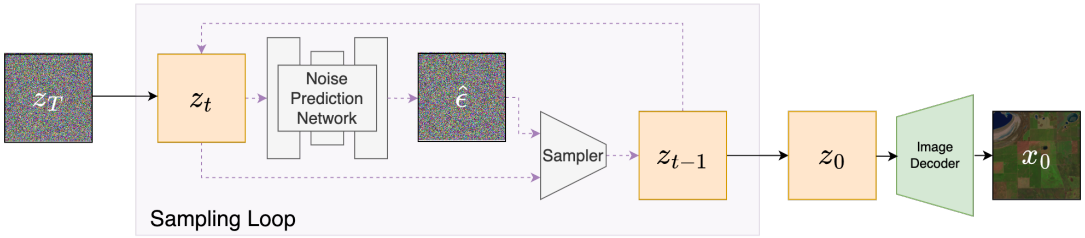


Figure 4.11: Diagram of sampling from a latent diffusion model. In this case, once a latent encoding has been generated by reversing the diffusion chain, the code is fed through the decoder part of the autoencoder network to generate an image.

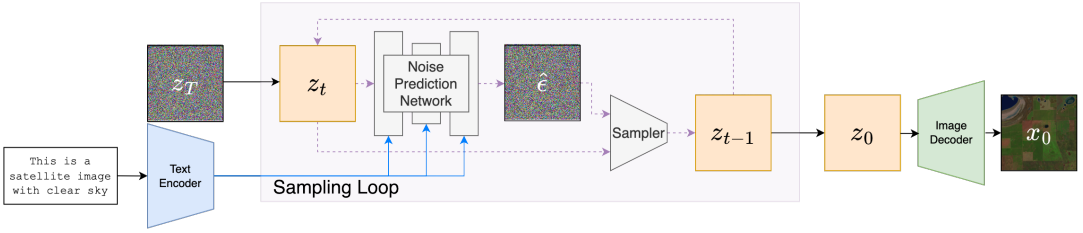


Figure 4.12: Example of conditioning latent diffusion on text. This is achieved by using an additional encoder for text and injecting the text-embedding into cross-attention layers of the neural network.

cross-attention components of the U-Net model. The text encoder and the core U-Net model are trained jointly with the standard loss used to approximate ϵ. The majority of StableDiffusion models released to date follow this approach, where text can be used to condition the synthesis of an image.

In the past few years, several approaches for employing denoising diffusion for the task of image inpainting have already been explored. Some prominent examples include Palette [129], where the network performing denoising diffusion directly in pixel space (no compression) is also provided with an additional channel corresponding to the inpainting mask, effectively serving as an extra condition present in the input. An alternative approach of RePaint [128] shows how a pre-trained denoising diffusion model can be used for inpainting with a different type of sampling technique, where the known regions of the input (passed through the forward process) are mixed with the diffused signal representation. Finally, StableDiffusion [204] also provides inpainting-oriented variants (different from the regular text-only variants), where the underlying latent

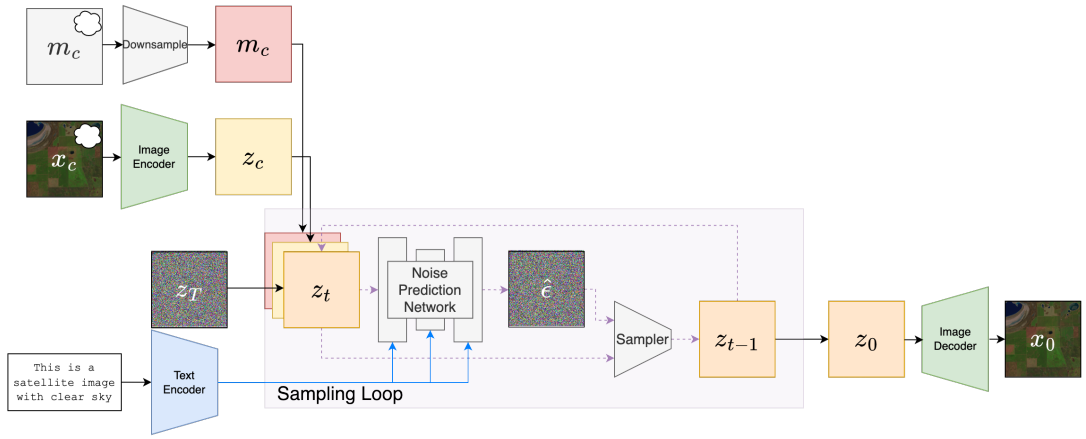


Figure 4.13: Diagram of Stable Diffusion Inpainting approach. The mask is interpolated to the same shape as the latent representations. The masked input image is fed through an image encoder, and along with the downsampled mask, it is concatenated with the generated latent code z_t.

space model besides the 4-channel latent representation accepts an extra channel for the mask (downsampled to match the latent spatial shape). This is demonstrated in Figure 4.13, where it is shown how the additional representations z_c (4-channel latent of the masked condition image) and m_c (1-channel downsampled mask) are concatenated with the input noisy state z_t before entering the core network. The standard cross-attention text-conditioning is employed too.

Lastly, another important component of the technique proposed in this work is ControlNet [199], which enables the injection of additional spatial guidance into the inpainting process. ControlNet has been introduced [199] as an extension to Stable Diffusion with the aim of incorporating more image-based conditions into the synthesis process. The method uses a separate encoding network to inject internal features to the core Stable Diffusion U-Net network based on a pre-selected condition type, such as Canny Edge. This is achieved by training a copy of some of the network's modules and merging the internal features of the copies and the originals via zero-convolution, as shown in Figure 4.14. At the beginning of the training process, a zero-convolution technique (1×1 convolution with all parameters initialised with zeros) nullifies the effect of the added network to ensure the preservation of the features learned by the core original model. It then learns to edit the internal feature representations of the core

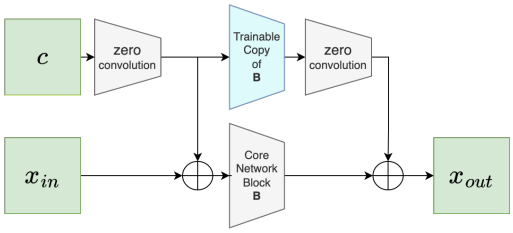


Figure 4.14: Diagram of the zero convolution technique [199]. The zero-convolution elements consist of 1×1 kernels initialised with zero weight and zero bias.

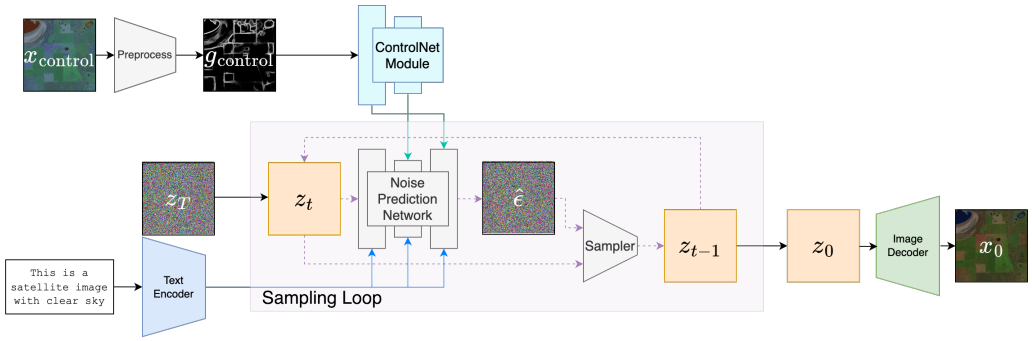


Figure 4.15: Diagram of ControlNet technique applied on top of the text-based StableDiffusion model.

network to satisfy a given control input image. For StableDiffusion, the first 12 encoder blocks and 1 middle block are copied for the ControlNet module, and their external features are injected into the skip connections of the middle block and the 12 decoder blocks.

Figure 4.15 illustrates the full scope of ControlNet applied to StableDiffusion, where it combines the text-based diffusion pipeline (as in Figure 4.12) with an additional component of ControlNet, where a control image x_{control} is passed through a preprocessing stage to obtain the control guide g_{control}, which is then passed to a trained ControlNet module (trained on that specific type of guide). The outputs of the ControlNet module are injected into the middle and decoder blocks of the core StableDiffusion model. In the original ControlNet paper [199], several choices for the conditioning signal g_{control} type are proposed, including Canny edge, Hough lines, user sketches, human pose, or HED boundary detections [205].

4.2.2 Proposed Method: Edge-Guided Inpainting

This work proposes to combine the components described in the above paragraphs into an edge-guided inpainting framework. As mentioned earlier, the Stable Diffusion inpainting model does not accept additional structural guidance into the process. On the other hand, ControlNet has only been applied to the regular text-to-image variants of Stable Diffusion, but not the inpainting one. At a high level, the edge-guided inpainting approach applies a pre-trained ControlNet model with a StableDiffusion inpainting model. As a result, the model can inpaint portions of an image, based on an additional control guide and text condition. This allows to apply StableDiffusion to satellite images, where a portion of a current image must be inpainted using information from a historical example. As shown in Figure 4.16, this is achieved by modulating the internal denoising network representations with the signal from the ControlNet. The full computation process is as follows: the denoised representation z_t is concatenated with the inpainted image condition x_c aand the downsampled mask condition m_c, which are then all fed into the noise prediction network, responsible for approximating $\hat{\epsilon}$. The noise prediction network is also fed with text-based conditioning encoded by the text encoder and the internal features are modulated by the ControlNet module in response to the control signal. This computation of $\hat{\epsilon}$ is used for diffusion reverse process sampling to yield a generated code z_0 starting from a pure noise sample z_T. The generated code z_0 is passed through the image decoder to produce the output generated image x_0.

The pipeline has been tested with a historical edge-guidance and a simple prompt of "a cloud-free satellite image" as text input. However, there is a large degree of flexibility for choosing different text prompts (including potential negative prompts) to guide the output as well as different sources of structural information other than historical image edges.

As a result, the framework can generate clean inpainted data based on the following inputs: the main input image x_c subject to inpainting, the inpainting mask m_c, the optional control guide x_{control}, and an additional (and also optional) text guide. The complete flow of information is visualised in Figure 4.16.

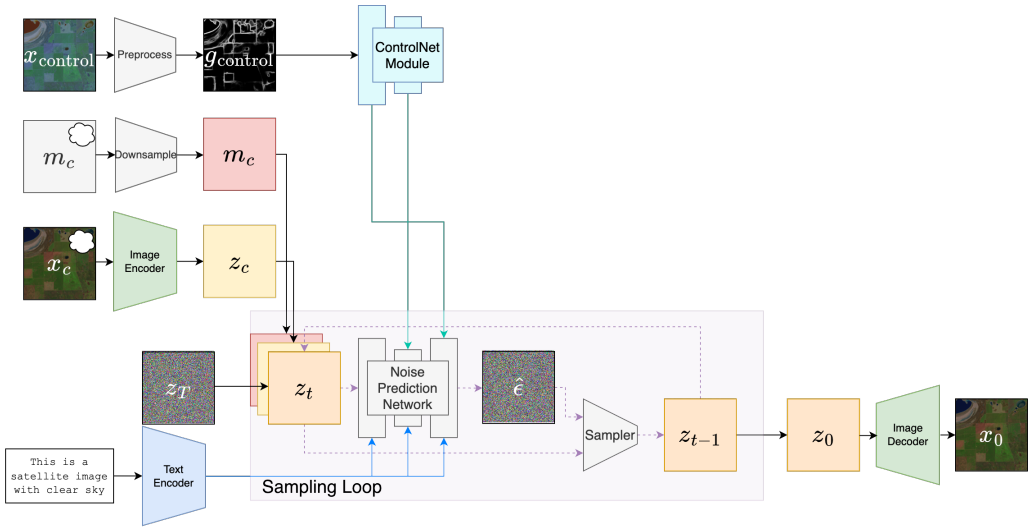


Figure 4.16: The Edge-Guided Inpainting diffusion pipeline used for this work employs a ControlNet approach [199], with an inpainting StableDiffusion backbone.

4.2.3 RGB-to-MSI transfer with Deep Image Prior

The diffusion frameworks trained on large datasets of RGB images can be readily applied to 3-channel satellite data, where the satellite RGB channels are likely the most appropriate to use. Yet, MSI modalities like Sentinel-2 contain more channels in their representation. Hence, inpainting in the RGB space may not be enough, when more channels from the MSI representation are of interest.

In this work, the context of completing the inpainting task without large-scale training on specialised satellite datasets is explored. To fulfil this criterion, the potential of employing the Deep Image Prior approach [139] is investigated, where a randomly initialised convolutional network is used as a source of prior for the inverse task of inpainting. More specifically, a sequential approach is followed, as shown in Figure 4.17, where the RGB images are first inpainted using the proposed edge-guided inpainting diffusion tool, and after that, Deep Image Prior approach is applied to the original MSI image stack, but with the incomplete RGB channels replaced by the inpainted content. In this way, it is possible to test whether the information injected into the process by the text-to-image model can be used to facilitate inpainting in the non-RGB bands.

For the Deep Image Prior component, a SkipNetwork with the same architecture

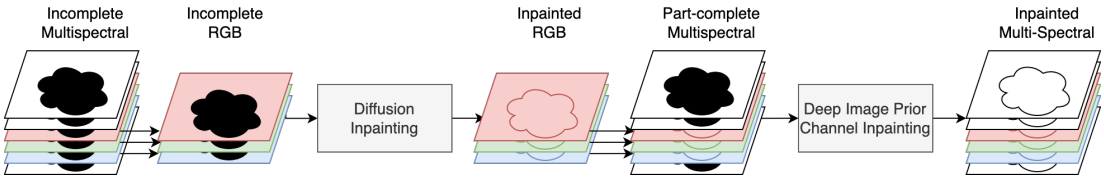


Figure 4.17: Complete pipeline for multi-spectral satellite image inpainting. The process involves two steps, where a pre-trained diffusion model is first applied to RGB data for inpainting, and then a Deep Image Prior [139] approach is used to propagate the inpainting in RGB space to all channels of the multi-spectral representation. The arrows in the first step of the chain represent indexing operation.

as in the stacked variant from Chapter 3 is optimised with MSE loss backpropagated from the known region for 4,000 gradient steps at a learning rate of 0.02. The known region contains available pixels of all non-RGB bands as well as a complete inpainted image of the RGB bands, arranged in the same fashion as the input representation.

4.2.4 Evaluation Method

The test dataset used for this study has been based on SEN12MS-CR-TS dataset [206]. A subset of 888 cloud-free Sentinel-2 test samples has been selected, each paired with the oldest historical cloud-free Sentinel-2 sample from the exact same location. This enables guidance of the inpainting process using historical data. The inpainting masks come from real cloudy samples captured in the same region computed using the s2cloudless tool [37].

The original Sentinel-2 data contain intensity images in [0-10,000] range and are subject to the same type of rescaling and normalisation as in the original dataset paper [77], followed by a clipping operation to constrain the samples to [0,1] range expected by the diffusion frameworks. It is also ensured that the mean value of the sample is not higher than 0.9 before the clipping operation to exclude saturated images from analysis (that is how 888 samples are selected from the initial pool of 1,031 samples).

As shown in Figure 4.17, the process begins with the extraction of the RGB channels from the input sample. These bands are then fed into the inpainting process based on ControlNet and StableDiffusion. The inpainted RGB image is then incorporated into

Figure 4.18: Comparison of the two methods of filling the masked region in the input to the diffusion models. Furthermore, the output achieved with the StableDiffusion Inpainting scheme is shown for reference as a result of using each method.

the original MSI sample (at this point, the RGB bands contain complete filled images, while the remaining channels do not) and supplied into the channel expansion process based on the Deep Image Prio.

The two tested text-based models include the standard StableDiffusion inpainting approach and the proposed Edge-Guided Inpainting approach.

4.2.5 Stable Diffusion Parameter Tests

As in the majority of the pipelines based on the denoising diffusion process, the operation can be controlled via multiple parameters, and these parameters are generally expected to have a considerable influence on the type of generated results. Here, the impact of the following features is analysed: (1) the content of the masked region, (2) the text-guidance scale, (3) the number of sampling steps, and (4) the edge-guidance scale (which only applies to the edge-guided inpainting approach). The results are shown in Table 4.2. For each tested parameter, the remaining parameters are set to the values printed in bold.

First, an important decision to make is whether the inpainted region in the input image should be filled with zeros or perhaps with some other content. This study is heavily focused on the utility of historical optical data and hence, a possible alternative is to fill these missing regions with the values extracted from the historical sample, in order to inject some structure information into the network input. An example of these two input variants is shown in Figure 4.18. It is found that injecting the historical sample to the missing region in the network input is highly beneficial and can greatly increase the quality of the produced output, especially in terms of structure, as

indicated by the large jumps in SSIM in Table 4.2. This effect is found to be strong for both standard StableDiffusion inpainting (inpainted SSIM goes from 0.54 to 0.67) and Edge-Guided Inpainting (inpainted SSIM goes from 0.58 to 0.67). These numbers also indicate that without using this technique, the edge-guided inpainting performs better at reconstructing the missing region, which is consistent with the fact that it receives additional guidance from the historical sample using ControlNet. However, once the technique of historical input filling is employed, both methods appear to achieve a comparable level of synthetic quality.

Another important parameter, commonly discussed in the context of StableDiffusion pipelines, is the classifier-free guidance scale for the text prompt (here referred to as text-guidance scale), as defined in [105]. Since additional information is supplied in the form of historical data (either via ControlNet or input historical filling), it is possible that text-based guidance is not as important in this use case as other priors instilled in the parameters of the StableDiffusion network. To test that, the text-guidance scale has been set to three distinct levels, ranging from 0.0 (no effective text guidance) through 1.0 to 7.5 (StableDiffusion default). Interestingly, while the differences are very small, the text guidance appears to be less useful for the standard StableDiffusion inpainting, with the performance slightly higher for low levels of text-guidance scale, while for the Edge-Guided Inpainting model, it is the opposite. Furthermore, these values are all very close to each other as shown in Table 4.2.

As in all denoising diffusion models, the sampling method and the sampling schedule can make a big impact on the quality of synthesis and also allow to trade off the two factors of the compute cost and output quality. However, since most of the discussions on this topic assume the quality to correspond to the visual attractiveness and similarity to the domain of realistic images, it is not obvious whether the same dynamics will be observed for the task of satellite image synthesis. Furthermore, it is not clear whether more can be gained by increasing the number of sampling steps since a large portion of the necessary information could be coming from the historical reference and the known regions of the unpainted region rather than from the wide domain knowledge. In this study, the UniPC sampler [203] is used for all diffusion sampling, and the main tested

Table 4.2: Parameter test results for the text-based models.

		StableDiffusion Inpainting				Edge-Guided Inpainting			
		SSIM (↑)		RMSE (↓)		SSIM (↑)		RMSE (↓)	
		Whole	Mask	Whole	Mask	Whole	Mask	Whole	Mask
Mask	Blank	0.70	0.54	0.11	0.15	0.71	0.58	0.10	0.14
Content	**Historical**	0.78	0.67	0.10	0.13	0.77	0.67	0.10	0.13
Text-	0.0	0.78	0.67	0.10	0.13	0.77	0.67	0.10	0.13
Guidance	**1.0**	0.78	0.67	0.10	0.13	0.77	0.67	0.10	0.13
Scale	7.5	0.77	0.66	0.11	0.14	0.78	0.68	0.09	0.12
	20	0.78	0.67	0.10	0.13	0.77	0.67	0.10	0.13
Steps	50	0.77	0.66	0.10	0.13	0.77	0.66	0.10	0.13
	100	0.77	0.66	0.10	0.13	0.76	0.66	0.10	0.13
Edge-	0.1	NA				0.78	0.67	0.10	0.13
Guidance	0.5	NA				0.79	0.69	0.09	0.12
Scale	**1.0**	NA				0.77	0.67	0.10	0.13

factor is the number of sampling steps. As in [203], 20 steps are reported to yield good output quality with this sampler and we use this value as a lower limit for this parameter. The synthesis quality is tested for longer sampling chains of 50 and 100 steps, and it is found that the increased number of steps does not have a beneficial impact in this case, and in fact, can lower the quality of the produced output.

Finally, for the edge-guided inpainting method based on ControlNet, the influence of the soft-edge HED conditioning can be controlled by another scale factor, as described in [199], which corresponds to a factor applied to the ControlNet features before they are added to the core network features. Here, the default value of 1.0 is tested, along with 0.5 and 0.1 values that explore a more subtle conditioning scheme. It is found that between the tested values, 0.5 appears to achieve the highest output quality and is hence used for the subsequent section.

4.2.6 Multi-Spectral Inpainting Evaluation

The proposed method is compared against several alternative solutions. First, the standard Stable Diffusion inpainting pipeline (with the same inpainting core as the proposed solution) is tested alongside the proposed edge-guided inpainting as a benchmark to measure the utility of the historical edge data in the first stage of the process

(the language-based inpainting). Second, two variants of Deep Image Prior are tested, where the Deep Image Prior (DIP) method is used as the only inpainting method (as opposed to the two-step approach outlined in the diagram in Figure 4.17 with the RGB Diffusion models). The first variant of DIP, referred to as Direct-DIP, uses the same convolutional neural network with the same parameters and is only supplied with the image to be inpainted. The second variant of DIP (Direct-DIP w/ Historical) receives a stack of the image to be inpainted and the historical sample (where the portion of the mask corresponding to the historical image is set to all 1). Finally, as a purely experimental reference, the performance achieved by applying the same type of DIP-based channel filling as for the RGB inpaintings from Stable Diffusion is reported, but for a case, where the RGB channels are populated with ground truth RGB data. This is done in order to provide a reference of the potential channel fill performance if the RGB inpainting performed in the first stage of the pipeline had no errors at all. This is referred to as the ideal-RGB channel fill in the table.

Table 4.3: Inpainting results computed for all 13 channels of the multispectral images in the test dataset.

Method	SSIM (↑)		RMSE (↓)	
	Whole	Mask	Whole	Mask
SD-Inpainting + DIP Post	0.78	0.65	0.16	0.21
Edge-Guided Inpainting + DIP Post	0.62	0.48	0.37	0.48
Direct-DIP	0.64	0.45	0.38	0.53
Direct-DIP w/ Historical	0.85	0.74	0.14	0.19
Ideal-RGB Channel Fill	0.89	0.82	0.12	0.16

Table 4.4: Inpainting results computed only for the RGB channels of the multispectral images in the test dataset.

Method	SSIM (↑)		RMSE (↓)	
	Whole	Mask	Whole	Mask
SD-Inpainting + DIP Post	0.78	0.67	0.10	0.13
Edge-Guided Inpainting + DIP Post	0.79	0.69	0.09	0.12
Direct-DIP	0.72	0.58	0.23	0.31
Direct-DIP w/ Historical	0.88	0.79	0.08	0.11

For the diffusion-based approaches, all registered metrics in Table 4.3 indicate much lower performance when operating on a stack of 13 MSI channels instead of 3 RGB

channels when compared to Table 4.2. Furthermore, the simpler Stable Diffusion inpainting technique appears to outperform the edge-guided approach for 13-channel data, indicating that the inpainting produced by that method is better aligned with the DIP-based channel-filling technique used in the second stage of the process. Furthermore, while the Direct-DIP results in poor performance, the Direct-DIP with historical data is the most competitive, indicating that the simple method of filling all channels from scratch with Deep Image Prior may be more powerful. Finally, neither method reaches the performance achieved by applying the channel fill with ideal-RGB values supplied, meaning that if the RGB inpainting was perfect, the two-stage technique with channel-filling method would result in the highest performance.

Since the text-to-image models delivered a higher performance for the RGB-only representation, that is another important context to consider. The test results for the RGB channels are reported in Table 4.4, which shows increased performance for the RGB bands across all methods, but for the two-stage methods, the edge-guided inpainting performs marginally better than standard Stable Diffusion inpainting.

Visual examples of the achieved results are shown in Figure 4.19 (RGB bands) and Figure 4.20 (13 bands). It becomes more apparent that Direct-DIP may be struggling to produce good inpainting without any support structural information and with large portions of the image missing. As in the case of all methods, the settings of DIP optimisations could potentially be tuned further, however, that lies outside of the scope of this work. Furthermore, despite the efforts to enforce the structure extracted from a historical sample for the text-based models, both models appear to generate some structurally inconsistent, yet visually appealing additions, which is likely responsible for the lower performance achieved by those methods. In the conducted study, it is the Direct-DIP with historical reference that appears to deliver the best results in both the visual and quantitative context.

4.3 Summary

The models trained on the combination of text and image data can learn powerful representations of the world from large-scale wide-context datasets. This could be

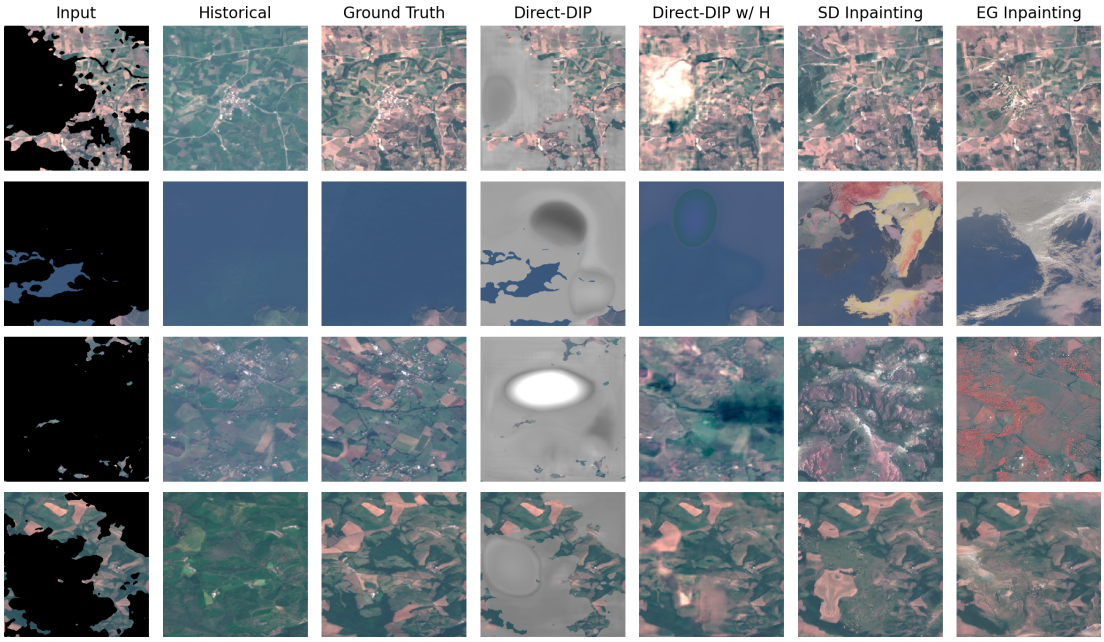


Figure 4.19: RGB visualisation of 4 random samples drawn from the test dataset and the corresponding output from each method. It is shown that the Direct-DIP struggles to perform good quality inpainting with no extra source of information, producing visually incoherent output. The text-based models appear to produce visually coherent, yet inaccurate inpaintings, despite the efforts to inject correct structural information into the process.

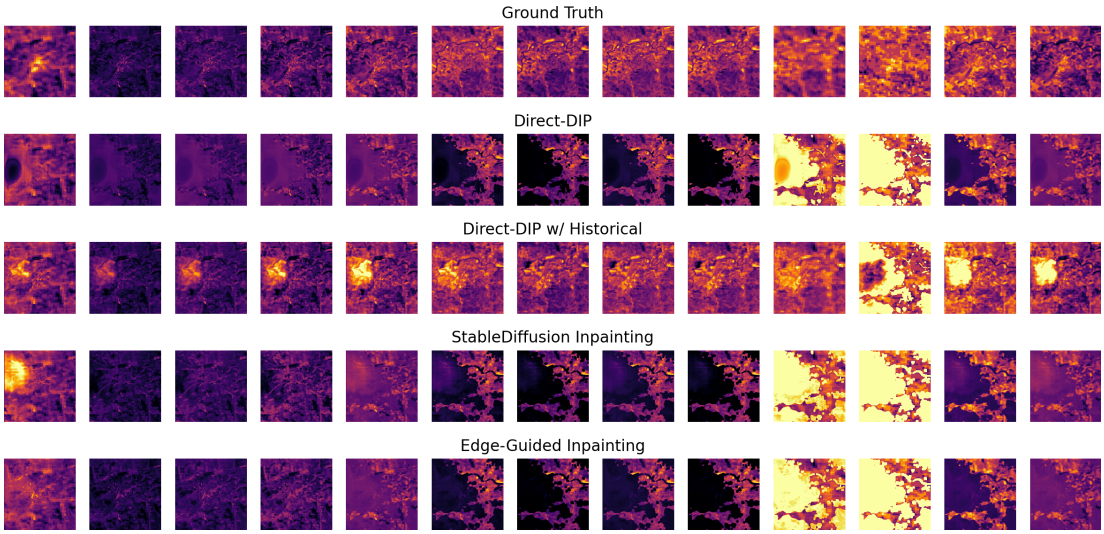


Figure 4.20: Comparison of the method output for all 13 bands of the first sample from Figure 4.19. This further shows certain instabilities and distortions produced in the output representation.

useful for defining new applications for remote sensing data. It has been shown that the CLIP model is capable of cloud presence detection in a zero-shot manner and that low-cost retraining can further improve the performance.

Apart from analyzing existing images, the language-based models can also be used for synthesis and it has been shown that the open-source Stable Diffusion models can be employed for image inpainting. A method of injecting additional spatial guidance (such as a historical image) was proposed as well as a DIP-based channel filling method to propagate the solution produced for the RGB channels into the remaining channels of a MSI representation.

It is shown that, even with structure-oriented adjustments, the general-purpose diffusion models may not be immediately performant for the task of MSI satellite image inpainting. Their high synthetic capability, while visually pleasing, appears to lead to added unnecessary distortions in the output. It is likely for those reasons, that the Direct-DIP baseline with a historical guide has been found to yield much higher performance, owing to the simple convolutional priors that the method is based on and easy access to structural guidance information.

There are several reasons why the application of text-based models may still be deemed promising and motivate further work on the topic. First, the synthesis process can be controlled via text-prompt in a zero-shot manner, which could enable a variety of different applications in other areas, such as image augmentation. However, for the specific task of image inpainting, these methods appear suboptimal when used in a zero-shot setting.

Chapter 5

Simulation of Clouds in Optical Satellite Images

A large number of cloud removal techniques are trained and evaluated on datasets containing real cloud-free images paired with cloudy observations from a proximate period, which may be a poor approximation of the ground truth. The difference between the cloud-free proxy image and the factual state behind the cloud can be problematic in the context of evaluating solutions (and potentially, this could also introduce noise at the stage of training). As shown in Figure 5.1, the datasets containing real pairs of cloudy and cloud-free images will often exhibit inconsistencies due to the changes occurring on the ground between the acquisitions. The figure contains an image pair from a commonly used SEN12MS-CR dataset [72], demonstrating that many details present on the ground are not consistent across images (for example, one field in the second crop goes has a bright brown hue in the clean image and green hue in the cloudy image). Consequently, inaccurate ground truth samples could be used for evaluation or learning if this approach is followed.

With the intention to expand the evaluation process and make it more reliable, the technique of training and evaluating on simulated data is explored. The clear advantage of the simulated data is that a synthetic cloud can be added to a cloud-free satellite image with a guarantee that the ground surface remains the same.

The chapter begins by highlighting the data-related challenges with a summary of

Figure 5.1: The approach of using pairs of real data often results in fundamental differences of the ground surfaces. Examples from SEN12MS-CR [72]. It is apparent, that some of areas significantly change their state among acquisitions.

existing approaches to obtaining paired cloudy data for cloud removal tasks is provided, which is followed by the introduction of analysis of the novel tool developed as part of this work, designed to generate an unlimited amount of synthetic pairs of cloudy and cloud-free satellite images.

Apart from the advantages of simulated data in the context of model evaluation, introducing synthetic data within the training process is beneficial too. First, it can be used for performance tracking as another source of information for validating model checkpoints. Secondly, simulating new data during training and minimizing the error on the synthetic samples leads to a larger and richer dataset. In a real image dataset, there can only be 1 or 2 (if both past and future captures are considered) cloudy samples associated with each cloud-free image. In contrast, the same cloud-free sample can be used to create a large number of simulated cloudy samples. This motivates the second important context being addressed in this chapter, namely, the use of simulated data for training.

In summary, the contributions of this chapter include the definition and implementation of SatelliteCloudGenerator, a versatile PyTorch-compatible tool for generating clouds in optical satellite imagery and subsequent demonstration and comparison between deep neural network models trained on real and synthetic data.

5.1 Two Sources of Paired Cloudy Image Data

There are two distinct paths to obtain pairs of cloudy and cloud-free samples that have been explored in the literature, one relying exclusively on real data and the other on

simulated images. Both are motivated by a certain limitation of the real physical world. The limitation is the fact that it is (most likely) not physically possible to acquire a cloud-free observation and a corresponding cloudy observation, where all factors (such as lighting, exact time and conditions on the planet's surface), apart from the presence of the cloud, are preserved.

The first approach to generating pair cloudy data relaxes this requirement for constant factors and it links cloudy and cloud-free images that are proximate in time. This occurs under the assumption that all other factors remain similar, rather than constant, within some unknown margin. However, the assumption may be too optimistic, and the conditions may vary enough to make the cloud-free sample a rather inaccurate approximation of the cloudy image with clouds removed. An example of that was shown earlier in Figure 5.1.

This approach has been adopted in works such as [35, 64, 72, 74, 169, 207]. In [74] and [35], Landsat image pairs are gathered with a time gap of 16 or 32 days. In [207] this gap is up to 15 days apart, while in [64] it could be up to 35 days. These lengths between acquired paired samples mean that the changes in the ground surface view may be profound even without any clouds present in either image. In SEN12MS-CR dataset [72], it is ensured that the optical cloudy and cloud-free images are captured within the same meteorological season, which appears to be a rather loose constraint, as illustrated in Figure 5.1. In the related work on SEN12MS-CR-TS dataset [169], 30 samples evenly spaced in time are captured for each ROI across the full year, yielding temporal gaps of at least 12 days.

The alternative approach maintains the constant-factor requirement, by simulating the cloud component and adding it to a source cloud-free image. In this case, the compromise is that it is not guaranteed how accurate the simulated clouds are; they could be a better or worse representation of the real phenomenon of clouds, depending on the quality of the simulation engine.

A limitation shared by both approaches is that they both rely on the presence of clear-sky data. This in itself enforces a very strong bias on the resulting datasets, since only the subset of all ground data is processed. The correlation between the cloud

cover and the state of the Earth's surface is likely not strong enough to completely hide away some features of the ground surface data. However, in the practical context, only a finite number of image samples is acquired over a finite temporal scope, which can contribute to substantial sampling bias. In the presence of this noisy sampling, the bias of the cloud-free samples could be quite strong and it is not yet clear how that can be mitigated apart from aiming for larger datasets.

The phenomenon of clouds in the Earth's atmosphere is a complex process and it may require a respectively complex and expensive computational simulation. Yet, the majority of literature to date has focused on borrowing from the fields of computer graphics, where the generation of random shapes that resemble structures encountered in nature has been of interest for many years [208]. In the seminal paper, Perlin noise has been introduced [208] as a relatively lightweight method for generating naturally looking random structures. Almost three decades later, the approaches of applying procedural noise for the simulation of clouds received some attention in the literature, starting with a rather brief description in [209], and eventually including some of the more developed use cases [58, 65].

Other than that, many hybrid approaches were proposed, seeking a trade-off between the disadvantages of real and simulated data. In [75], cloud masks are extracted from real images using either layer separation methods or channel threshold and then used to synthesise a cloudy image. Some similar approaches have also been previously applied to the problem of dehazing [73, 210], and later revisited for the thin cloud removal problem [78]. In the case of [73, 78], the transparency is adjusted by channel wavelength. In [211] a framework of cloudy image arithmetic is proposed, which relies on extracting real clouds (rather than masks) from images and then the addition of those clouds to new scenes.

In this work, a novel technique is proposed, where the simplicity of the Perlin noise is combined with a flexible and versatile open-source framework for simulating realistic clouds. It enables previously unexplored features such as control over the scale of the synthesised clouds, the thickness of the clouds, the influence of the ground image over the perceived colour of the cloud, spatial misalignment of the cloud layer between im-

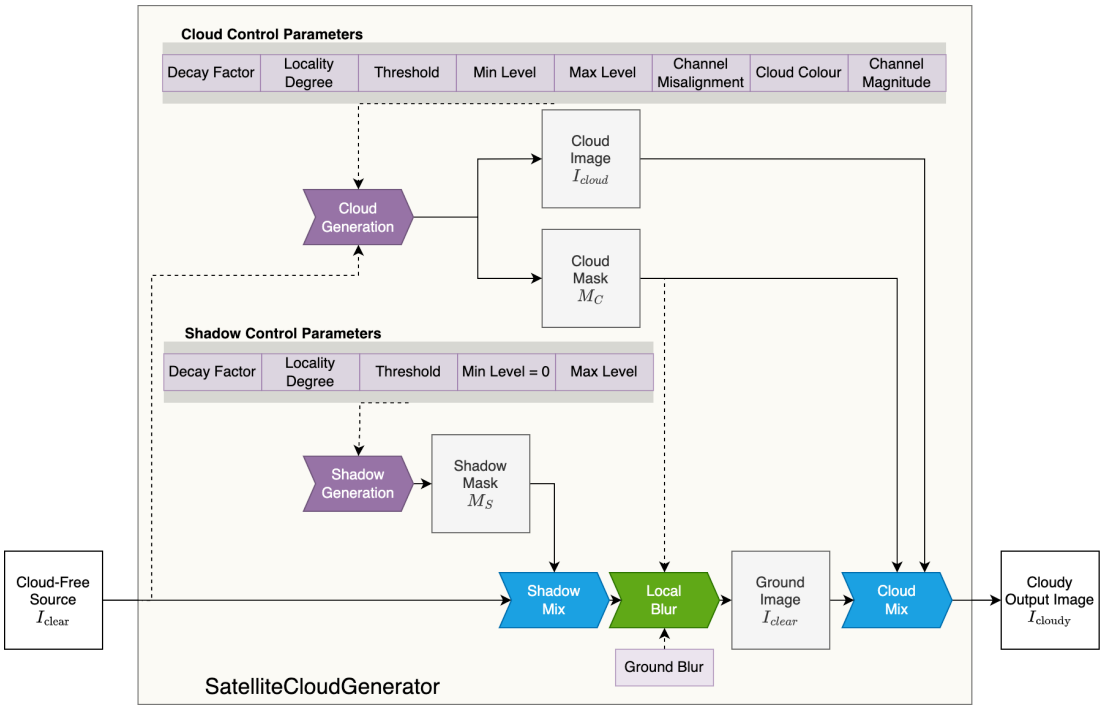


Figure 5.2: Diagram of the SatelliteCloudGenerator pipeline for generating an image with cloud and shadow presence from a cloud-free source image.

age channels, blurring of the cloud, simulation of cloud shadows, and channel-specific magnitude. It is demonstrated how these settings are managed by adjusting configuration objects and how the generated cloud masks and shadows can be converted to segmentation masks if required.

In summary, the SatelliteCloudGenerator[1] framework provides a high level of flexibility for generating an unlimited number of cloudy-clear image data pairs.

5.2 SatelliteCloudGenerator Framework

What follows is a more in-depth description of SatelliteCloudGenerator, starting from the method of noise generation and then progressing to the analysis of specific parameters that control features of the generated data. An outline flow diagram of the internal operation of SatelliteCloudGenerator is shown in Figure 5.2, where a cloud-free source image I_{clear} is supplied as input and a cloudy image I_{cloudy} is returned as output.

[1] https://github.com/cidcom/SatelliteCloudGenerator

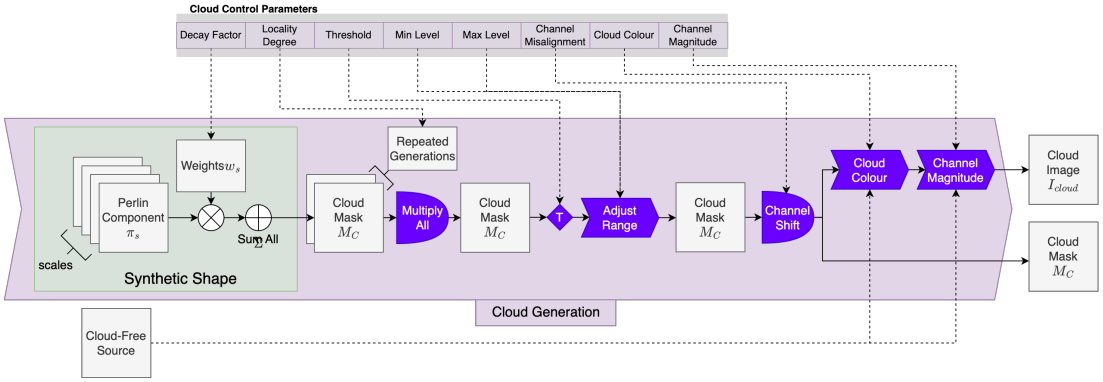


Figure 5.3: Diagram of the SatelliteCloudGenerator cloud generation component.

Furthermore, the core component responsible for generating a cloud mask and a cloud image is shown in more detail in Figure 5.3 and a similar (but not the same) functionality responsible for shadow generation is shown later in Figure 5.17. The following paragraphs provide detail on the pipeline represented by these diagrams.

5.2.1 Synthetic Shape

The key structure of the generated clouds is derived using a function based on Perlin noise [208], as indicated by the green region of the diagram in Figure 5.3. The use of Perlin Noise has been explored in earlier literature [58, 65, 209], but little detail is provided about the generation process. This work explicitly reports on that and defines a set of parameters to simulate a diverse range of cloud transparency maps.

The first stage of the process involves generating the base shape of the cloud transparency map. This can be done using procedural synthetic noise generation methods, and here, a Perlin noise π_s generated at several harmonic scales s is used to generate the resulting cloud shape mask M_C. As shown in Figure 5.4(a)-(c), Perlin noise can be generated at various scales, resulting in different frequencies present in the spectrum. These different scales can be weighted with scalar factor w_s and summed in order to generate more complex-looking noise structures, as shown in Figure 5.4(d).

$$M_C = \sum_{s}^{N} w_s \pi_s \tag{5.1}$$

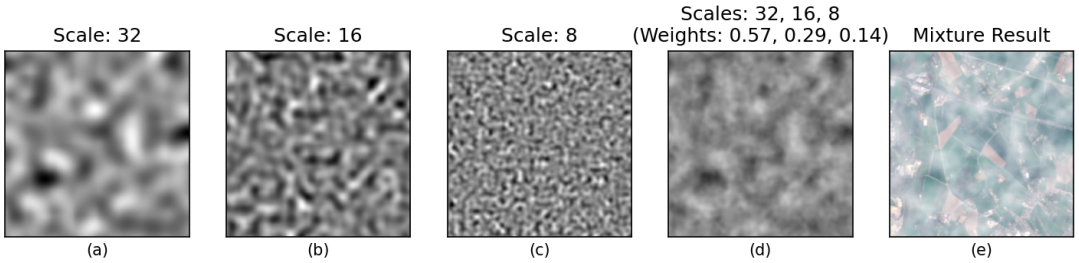


Figure 5.4: Demonstration of the Perlin noise generated at several scales (32, 16, 8) (a)-(c) and the result of their weighted sum (d). An example of the resulting cloud mask mixed with a real image is shown in (e).

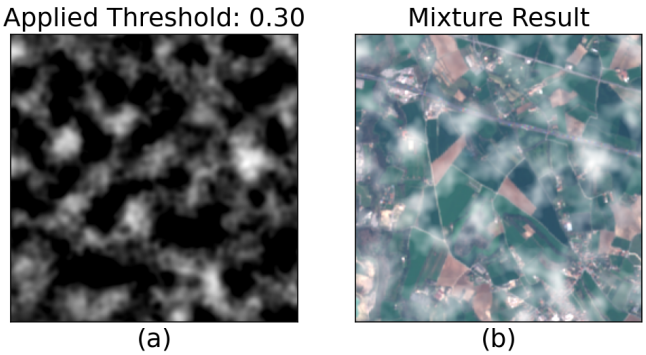


Figure 5.5: Example of a threshold of 0.30 applied to the cloud mask from Figure 5.4.

The weights w_s applied to individual shapes at each scale s control the spectral content of the image (spatial frequency domain). Hence, they can be adjusted to obtain shapes that are smoother by applying lower weights for finer scales, or sharper by increasing the weight of those scales. This is controlled by the `decay_factor` parameter. By default, this factor is set to 1, and higher values will result in smoother shapes. The decay_factor d_f parameter corresponds to the exponent, to which the base of each weight w_s is raised. This base corresponds directly to the scale number s:

$$w_s = s^{(d_f)} \tag{5.2}$$

The resulting shape computed using the Perlin noise method is likely to have a few sparsely distributed global minimum points, instead of a larger region of floor values, which would be required to produce larger areas in the image with no clouds present. To produce such an effect, a `clear_threshold` can be applied, which will assign a value

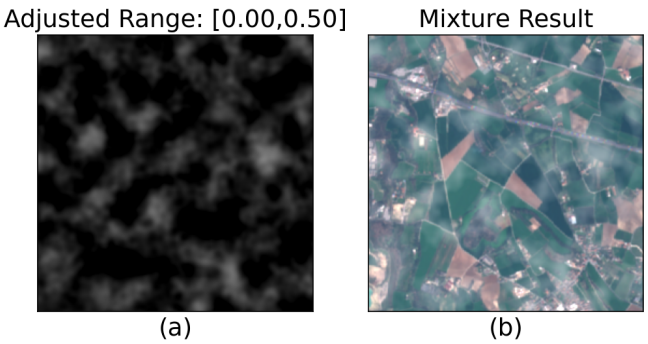


Figure 5.6: Example of range adjusted to [0.0, 0.5] from the original shape

of 0.0 to all values of M_C below that threshold, as illustrated in Figure 5.5. For a noise shape M_C scaled to the range of [0,1], the threshold τ_c is applied as:

$$M_C \leftarrow \frac{\text{ReLU}(M_C - \tau_c)}{\max(M_C) - \tau_c} \tag{5.3}$$

The denominator of $(\max(M_C) - \tau_c)$ is used to rescale the range of the noise shape back to [0,1].

Once the shape passes through the threshold operation, the value range can be adjusted by setting the `min_lvl` and `max_lvl` parameters, which shift the minimum and maximum value of the shape to these two levels, correspondingly. For example, a `min_lvl` value of 0.0 will indicate that the most transparent pixels will have no cloud cover at all. By increasing the `min_lvl` it can be ensured that all pixels have cloud presence at least at that level. An example of range adjustment is shown in Figure 5.6. This gives the properly scaled transparency mask M_C for the cloud:

$$M_C \leftarrow \text{min\_lvl} + M_C \odot \text{max\_lvl} \tag{5.4}$$

The shape mask M_C adjusted to the range of [`min_lvl, max_lvl`] is treated as the final transparency map of the simulated cloud. The last step of the process involves using this mask in a mixing operation with the clear-sky source sample. The mixing operation for the clouds is defined as

$$I_{\text{cloudy}} = I_{\text{clear}} \odot (1 - M_C) + M_C \odot I_{\text{cloud}}, \tag{5.5}$$

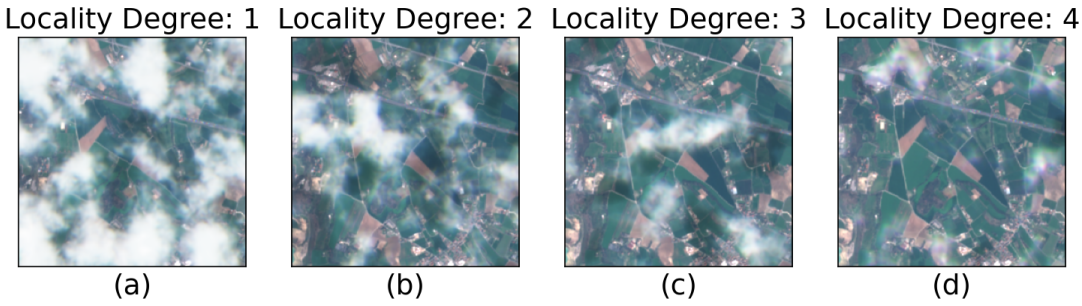


Figure 5.7: Varying levels of locality degree obtained for a range of `locality_degree` parameter values.

where the output is the cloudy image I_{cloudy}, based on a source clear-sky image I_{clear} and a cloud-component image I_{cloud}. In the simplest setting, the cloud-component image I_{cloud} could be equal to a constant colour of the ambient cloud, however, the subsequently introduced feature attempts to make this aspect more realistic.

5.2.2 Cloud Locality Degree

Another desirable feature is to be able to make the generated clouds more local. In many cases, the clouds will only occupy a limited area of the image, as opposed to an approximately uniform spread generated by the Perlin noise method. The approach proposed in this work is to multiply several generated cloud mask shapes to increase the sparsity of the cloud shape, which is performed directly before the thresholding operation, as shown in the diagram in Figure 5.3. Each shape in multiplication decreases the likelihood of a high value of cloud thickness being preserved in the resulting product. After multiplication, the cloud shape is rescaled to [0,1], and a similar process of applying threshold and then scaling to the range between `min_lvl` and `max_lvl` is performed. An example is shown in Figure 5.7, where the parameter of `locality_degree` indicates the number of noise shapes multiplied by each other. As shown, the clouds become sparser with the increasing value of this parameter.

Notably, a changed locality of the clouds can also be achieved by increasing the `clear_threshold` value, as shown in Figure 5.8. This, however, brings another effect of sharper cloud edges compared to the example with changed locality degree (Figure 5.7).

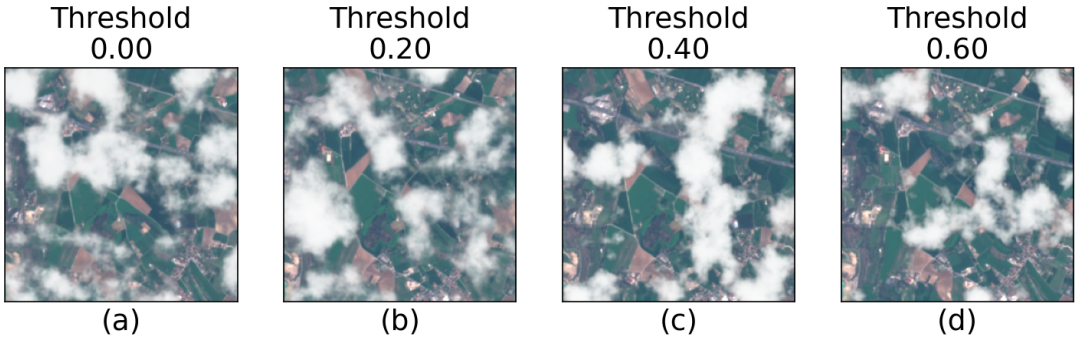


Figure 5.8: Different degrees of locality can also be achieved by adjusting the `clear_threshold` parameter value, but the cloud edges tend to become sharper.

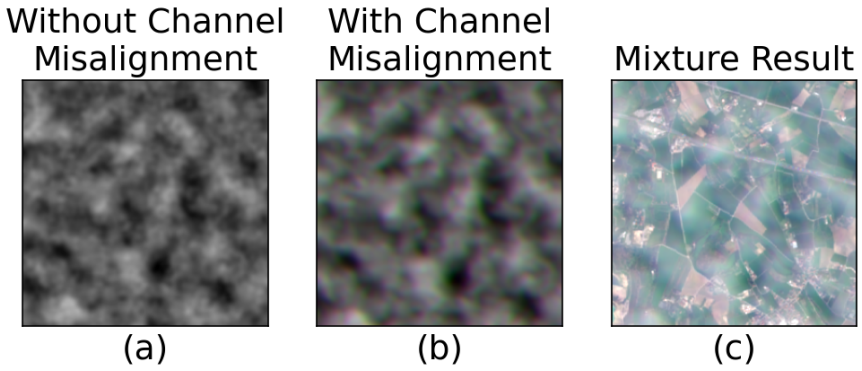


Figure 5.9: An example of channel misalignment (b) applied to a cloud shape (a). The result of mixing the cloud-free image with the channel shifted cloud shape is shown in (c).

Effectively, the proposed tool provides two distinct ways of producing sparser clouds with distinct visual effects as illustrated by Figures 5.7 and 5.8.

5.2.3 Channel Misalignment

The real cloud data will often exhibit an effect of channel misalignment, where individual channels of the cloud object are spatially misaligned due to the velocity of the acquiring sensor (if the individual channels are sensed at slightly different time instants) [75]. This effect can be simulated using the `channel_offset` parameter, which determines the maximum possible offset between two consecutive channels in either the x or y spatial dimension of the image, in terms of the number of pixels. In the generator, the exact value of shift in each dimension will be sampled uniformly from

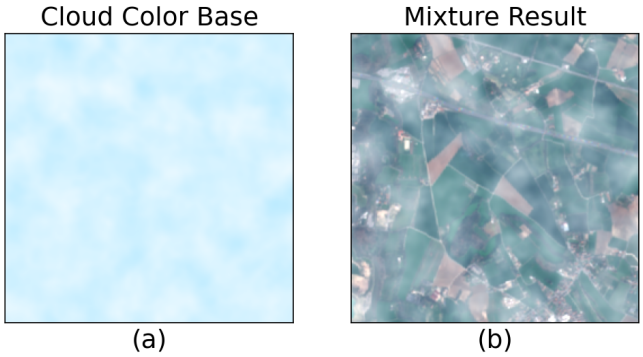


Figure 5.10: An example of a cloud with colour adjusted by the ground reflectance. I_{cloud} based on the ground colour (a) and the result of mixing using such a base cloud colour (b).

[-`channel_offset`, +`channel_offset`], resulting in a range of potential discrete offsets. Subpixel values are not considered. An example of the feature is shown in Figure 5.9.

5.2.4 Cloud colour

As described in an earlier work of [75], the clouds present in satellite imagery do not generally resemble a purely white component, but rather, are coloured by the ambient light reflected from the ground.

The cloud colour will tend to be similar to the mean colour reflected from the ground surface in that area. This colour can be computed by averaging all pixels (mean value per channel) in the source clear-sky image. Furthermore, this effect is partially dependent on the cloud thickness, meaning that the thicker cloud will let through less light from the ground and therefore, the influence of the ground colour is weaker.

To simulate this feature, the colour cloud component I_{cloud} is adjusted to a value between pure white and the average normalised colour γ of the clear ground image I_{clear}, which is performed after the channel misalignment operation. The ambience colour γ is normalised by scaling the maximum value to 1. In effect, the cloud colour component I_{cloud} receives the following assignment:

$$I_{\text{cloud}} \leftarrow 1 \odot (1 - M_C) + \gamma \odot M_C \tag{5.6}$$

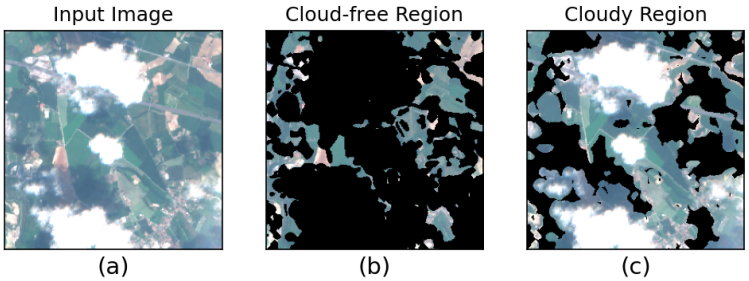


Figure 5.11: Example of cloud-free and cloudy regions masked from a real cloudy sample.

Figure 5.10 shows an example of I_{cloud} (left) and the final result I_{cloudy} of mixing with the colour cloud base (right).

5.2.5 Channel-Specific Magnitude

In several works [78, 212] it has been noted that the exact magnitude of the cloud transparency mask will depend on the wavelength of the specific image channel. This means that the phenomenon of cloud presence in satellite images does not only vary across space but also across sensor channels. In the context of evaluating the techniques for detection and removal of clouds, this aspect is particularly important, in order to make the simulated data as similar as possible to the real data. During training, this could also be important for many tasks, since optimizing the loss only on simulated clouds could make the real clouds be perceived as out of domain objects and as a consequence, prevent successful detection or removal.

The intensity of the cloud component can be adjusted by applying a set of channel-specific weights to I_{cloud} before mixing with the cloud-free input image, as shown in Figure 5.3. However, it is not immediately clear what the values of those weights should be.

In this work, the channel magnitude weights is extracted from real cloudy images accounting for the ratio ρ between a selected statistic feature c_{clear} in the cloud-free region and another statistic feature c_{cloud} in the cloud-affected region of the image. Figure 5.11 demonstrates how a real cloudy sample can be used for sampling cloud-free and cloudy regions using the cloud detection technique of s2cloudless [21].

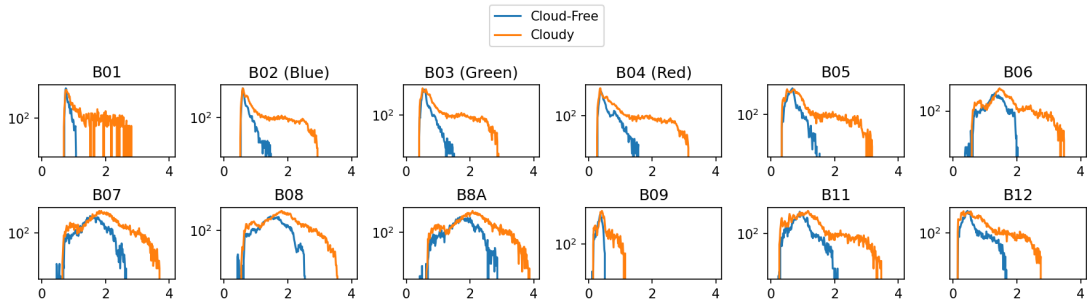


Figure 5.12: Histogram curves for the example image. In each case, it is shown that the cloudy region (orange plot) contains higher values, but also exhibits some resemblence to the cloud-free histogram (blue) due to the leakage.

$$\rho = \frac{c_{\text{cloud}}}{c_{\text{clear}}} \tag{5.7}$$

For example, the statistic feature can be chosen as the mean reflected colour. However, in some cases, the cloudy region may be heavily influenced by the non-cloudy due to the likely presence of nearly-cloud-free pixels in the cloud mask, as illustrated in Figure 5.11. This interference can lead to an underestimated reflection statistic from the cloudy region. To reduce this effect, the statistic c_{cloud} is instead selected as the 95% quantile of the distribution observed in the cloudy region. This value can be expected to correspond to be close to the maximum reflected value in the true cloudy region, with more stability than the maximum value (100% quantile). This approach should work well for the vast majority of scenarios, as long as more than 5% of the cloud mask coverage does, in fact, contain a cloud.

A further illustration of the relationship between the distribution of values in the cloudy and cloud-free region is shown in Figure 5.12, where histogram curves are shown for the cloud-free (blue) and cloudy (orange) region of the image, individually for each band. It is apparent that the cloudy region tends to contain higher values for each channel. The leakage of cloud-free pixels to the cloud mask can also be observed, manifested by similar histogram shapes for the lower values. It may be worth comparing these curves to the histogram curves of a real completely cloud-free image from the same location and proximate time, as shown in Figure 5.13. It appears that the histogram curves extracted from the cloud-free region of a cloudy image, and the curves extracted

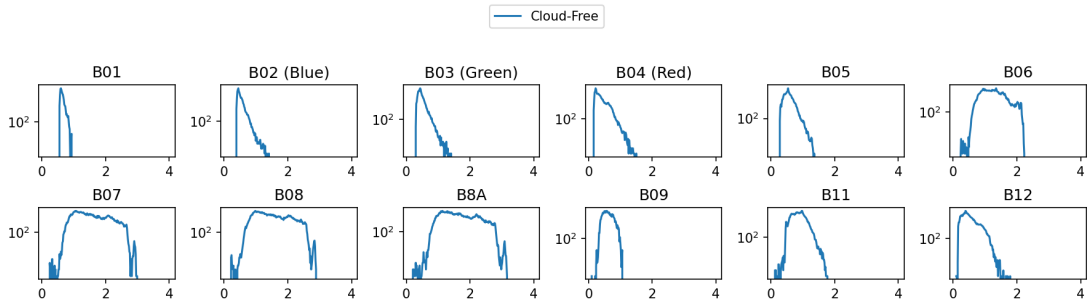


Figure 5.13: Histogram curves for the example cloud-free image. The histogram curves exhibit some resemblance to the cloud-free histogram curves extracted from the cloud-free region of a cloudy image.

from the cloud-free image are very similar.

Since the cloud-free mask is generally unlikely to contain any clouds, the statistic c_{clear} extracted from that region can be closer to the centre and is indeed selected as the central 50% quantile (median) of the distribution.

$$\rho = \frac{c_{\text{cloud}}^{95\%}}{c_{\text{clear}}^{50\%}} \tag{5.8}$$

The ratio ρ can then be multiplied with the statistic $\hat{c}_{\text{clear}}$ extracted from a new cloud-free image and give a predicted channel weight vector $\hat{c}_{\text{cloud}}$ for the cloud strength:

$$\hat{c}_{\text{cloud}} = \hat{c}_{\text{clear}}\rho \tag{5.9}$$

The cloud component I_{cloud} is then multiplied by $\hat{c}_{\text{cloud}}$ to give a magnitude-adjustment cloud component:

$$I_{\text{cloud}} \leftarrow \hat{c}_{\text{cloud}} \odot I_{\text{cloud}} \tag{5.10}$$

Figure 5.14 shows the effect of applying this approach, showing that the application of the CSM scaling results in an image more visually similar to a real reference (in both cases, the values go well above 1.0 and are hence saturated in this figure).

More insight can be obtained by exploring the histogram curves within each band for the three images in Figure 5.14. This is shown in Figure 5.15, where each row shows the cloud-free and cloudy histograms for each image. It is apparent that a simulated

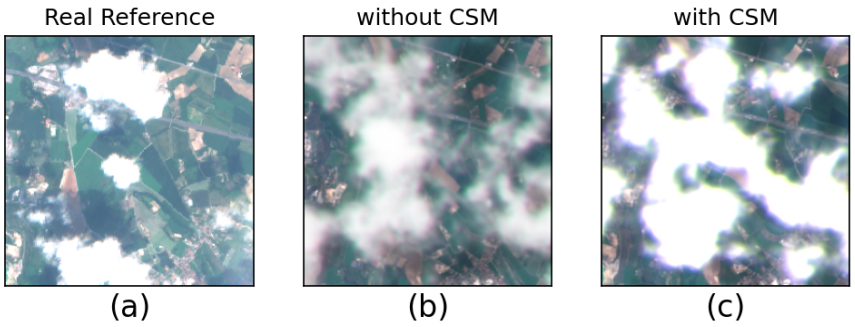


Figure 5.14: Example of a sample simulated with (c) and without (b) channel-specific magnitude (CSM) scaling. The real image reference used for magnitude scaling is shown in (a).

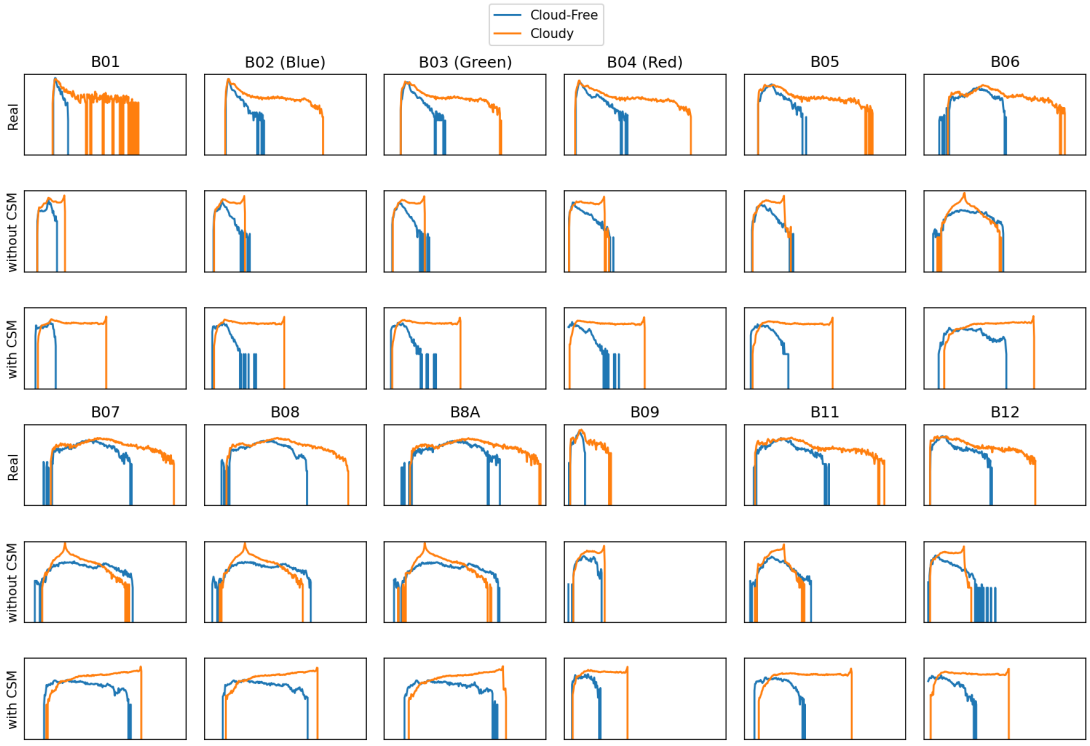


Figure 5.15: Comparison between the cloud-free and cloudy histogram curves between a real image (top), a simulated image with naive magnitude scaling (middle), and channel-specific magnitude scaling (bottom). It can be observed that the channel-specific magnitude histogram curves cover value ranges more similar to those in the real cloudy image. Plot axis are the same as in Figure 5.12 and 5.13, but are omitted for clarity.

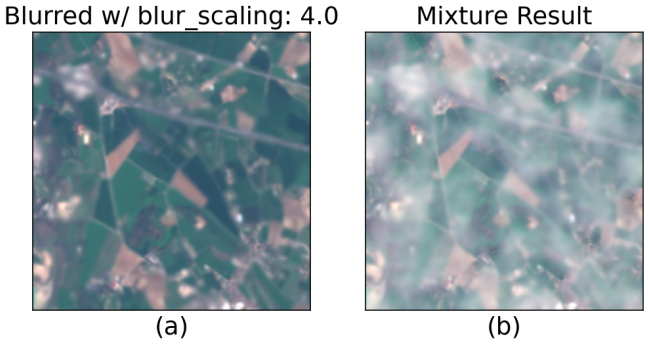


Figure 5.16: Example of the ground blurring effect. Locally varying Gaussian blur kernel is applied to the image to yield a locally blurred ground image (a), which can then be used as the representation of the ground in the mixture (b).

image (middle row) results in an unnatural peak around the value of 1.0, and the cloudy colour distribution (orange) rarely exceeds the range of the cloud-free colour distribution (blue), as it should in a real image (top row). However, if channel-specific magnitude (CSM) feature is applied, the range of intensities is more similar to a real cloudy sample and consistently higher than the cloud-free region range.

5.2.6 Ground Blurring

Another effect of the through-cloud scattering, besides changed cloud colour, is the blurring of the underlying ground image, as identified in [75]. There, the source clear-sky image is transformed into a mixture of the original image and a blurred version (blur with a constant Gaussian kernel), performed based on an alpha mask dependent on the cloud thickness.

Here, a more precise approach is developed by applying convolution to the source clear-sky image I_{clear} with a locally changing Gaussian blur kernel $h()$, as opposed to a static one. The variance of the used kernel is proportional to the cloud thickness M_C, and can be adjusted by multiplying the modulating signal M_C by the `blur_scaling` factor β.

$$I_{\text{clear}} \leftarrow I_{\text{clear}} * h(\beta M_C) \tag{5.11}$$

By default, this value is equal to 1.0, when the ground blurring effect is applied. It

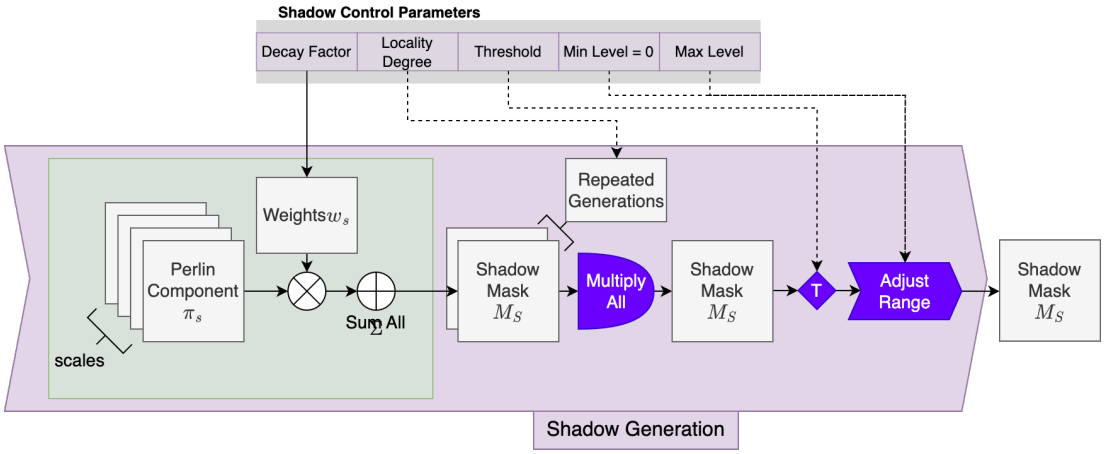


Figure 5.17: Diagram of the SatelliteCloudGenerator shadow generation component.

can also be disabled by assigning 0 to the `blur_scaling` factor β.

Figure 5.16 illustrates the output of the blurring operation (a) as well as the final mixture output (b).

5.2.7 Ground Shadow

Satellite images with cloud presence will often include shadows cast on the ground surface. Depending on the sun's angle and other conditions, the shadows present in the image could be a result of clouds that are outside of the view. For that reason, the shadows generated in an uncorrelated fashion could be a plausible representation of a possible event. An example is shown in Figure 5.18. The mixture process for a shadow is similar to the cloud mixing operation, in a manner analogous to Equation 5.5:

$$I_{\text{clear}} \leftarrow I_{\text{clear}} \odot (1 - M_S) + M_S \odot I_{\text{clear}} \tag{5.12}$$

but since the shadow component image I_{shadow} can be approximated by a zero constant, the operation will simply be:

$$I_{\text{clear}} \leftarrow I_{\text{clear}} \odot (1 - M_S) \tag{5.13}$$

Similarly to the cloud example, a simple mask, such as one shown in Figure 5.18(a),

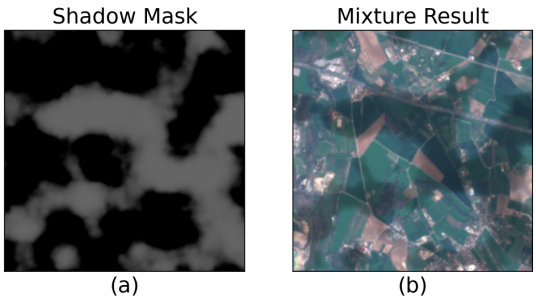


Figure 5.18: An example of a shadow generation feature.

can be mixed with a clear image to achieve a simulated shadow in the image as in Figure 5.18(b):

5.2.8 Configuring Cloud Generators

The use of synthetic noise source allows to generate effectively an unlimited number of cloudy samples for every single cloud-free source image. Consequently, it is possible to sample from a very wide distribution of samples, much larger than what can be stored and packaged into a single dataset. To model this source of data as a sampler, this work introduces modules known as Cloud Generators.

The introduction of Cloud Generators allows to encapsulate a specific simulation configuration (corresponding to a type of generated clouds) with a sampling function. A Cloud Generator module behaves in a manner similar to torchvision image augmentation modules and inherits from `torch.nn.module`. That way, new samples of specific type can be generated by simply passing through this module.

Four predefined configurations are provided in the software release, and new ones can be created as a Python dictionary. Table 5.1 contains the parameter levels for each of these four configurations. The first one uses a wide range between `min_lvl` and `max_lvl` values to simulate large thick clouds in the image. The other three generators focus on more specific types of clouds, namely, local (thick clouds covering a smaller portion of the image), thin (local semi-transparent clouds), and fog (semi-transparent layer over the entire image). Thick clouds are achieved by setting the `max_lvl` parameter to 1.0, meaning that portions of the image will contain pixels completely dominated

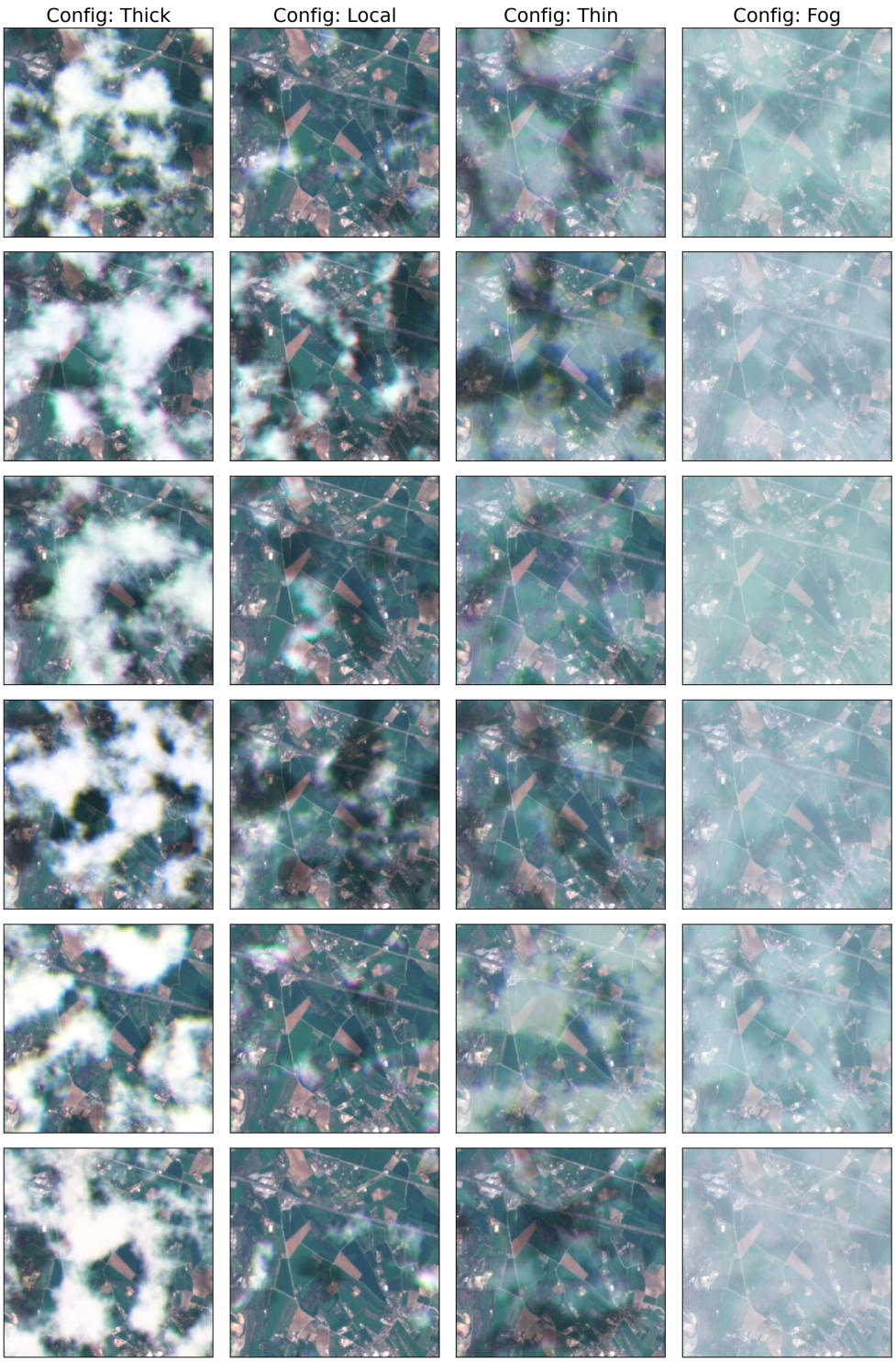


Figure 5.19: Random samples synthesised using 4 different Cloud Generator configurations used for this work.

by the cloud component. The thick and local configurations differ only by value of the `locality_degree` parameter, where thick has that set 1 (large clouds), and local has it set to a [2,4][2] range (various degrees of more local clouds). The thin configuration limits the `max_lvl` parameter to [0.4,0.7] range, to ensure semi-transparency. Finally, Fog lifts the `min_lvl` parameter to [0.3,0.6] range, resulting in the full image containing a semi-transparent cloud.

Table 5.1: Configuration parameters for four types of clouds.

Parameter	Config: Thick	Config: Local	Config: Thin	Config: Fog
`min_lvl`	0.0	0.0	[0.0,0.1]	[0.3,0.6]
`max_lvl`	1.0	1.0	[0.4,0.7]	[0.6,0.7]
`threshold`	[0.0,0.2]	[0.0,0.2]	0.0	0.0
`locality_degree`	1	[2,4]	[1,3]	1
`decay_factor`	1.0	1.0	1.0	1.0
`cloud_colour`	True	True	True	True
`channel_offset`	2	2	2	2
`blur_scaling`	2.0	2.0	2.0	2.0

Example samples from each configuration are shown in Figure 5.19.

5.2.9 Generation of Segmentation Masks

For many applications, especially that of cloud detection, segmentation labels are required to train the models. Since the cloud simulation tool described herein has direct access to the cloud mixing mask, it can be used to generate discrete segmentation data.

The process of transforming the exact cloud and shadow mixing masks M_C and M_S to discrete segmentation-like labels is as follows. The format of the segmentation labels in this example will follow the approach in CloudSEN12 [11], but could be easily adapted to other formats. In this case, the segmentation map is composed of 4 classes, 0 for clear sky, 1 for thick cloud, 2 for thin cloud, and 3 for cloud shadow. This way, the label output M_{seg} can be based on 3 binary masks:

$$M_{\text{seg}} = 1 \odot B_{\text{thick}} + 2 \odot B_{\text{thin}} + 3 \odot B_{\text{shadow}} \tag{5.14}$$

[2]For any part of the configuration expressed as range, the used value is extracted by sampling from a uniform distribution (discretised, if necessary).

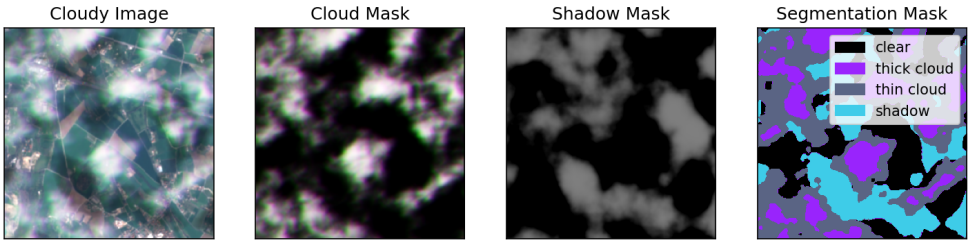


Figure 5.20: Example of a precise segmentation mask derived from the simulation tool.

where B_{thick}, B_{thin}, and B_{shadow} are the binary segmentation masks for thick clouds, thin clouds, and shadows, respectively. The values of these binary masks can be derived directly from the cloud and shadow transparency masks based on a range of values that should result in a positive binary value. An example of the process of translating a cloud mask and a shadow mask into a segmentation mask is shown in Figure 5.20.

In the cloud detection training example that follows in the next paragraphs, both thick and thin cloud classes are merged in to a single class representing a cloud. In this case, any value above 0.1 for the cloud transparency mask or shadow transparency mask results in a positive binary value for that pixel. For pixels with both cloud and shadow presence, the cloud label is assigned.

5.3 Comparison to Real Data: Cloud Detection

The first experiment to show the use of synthetic data for training and evaluation is focused on the task of cloud detection. For this task, the architecture of MobileNetV2 [213] (which achieved the highest performance in the CloudSEN12 paper [11]) is trained from scratch on Sentinel Level-2A images. The networks are trained from scratch (no pretraining) in order to match the exact optimisation conditions for the explored data variants. The baseline variant (a) is optimised on the manually annotated data of real clouds sourced from the official high-quality subset of CloudSEN12 dataset (the first released version) [11]. The alternative variants (b)-(d) use the clear images from that subset and simulate the clouds using the simulation method proposed in this work. The variants involving the use of simulator data include two variants that use samples simulated without channel-specific magnitude (b) and (c), and two variants (c)

and (d) where the channel-specific magnitude is used. In each case, a fully synthetic approach is tested as in (b) and (d), as well as a hybrid approach that mixes 50% of real data with 50% of simulated data as in (c) and (e).

For this experiment, the network operates on the bands contained in the Sentinel-2 L2A product. The networks are trained with the standard cross entropy loss for 3 classes (clear, cloud, shadow), until a point where the validation loss does not decrease for 240 epochs. Each batch contains 32 clear images and 32 cloudy images, but the loss on the clear images is multiplied by a factor of 0.1 so that the cloudy images are prioritised during learning. The weights parameters are optimised using an AdamW [214] optimiser with the initial learning rate of 10^{-3}, which is scaled down by a factor of 0.1 whenever the validation loss does not decrease for 128 epochs.

As a result, each network has been trained for about 20,000 optimisation steps until the validation loss ceased to improve. Although this process could be tuned further to optimise various learning hyperparameters and yield a lower validation loss, the motivation for the experiments conducted here is to compare the effect of the real or simulated training data for these models.

The metrics reported for the performance were selected based on the CloudSEN12 work [11], where Producer's Accuracy (PA), User's Accuracy (UA) and Balanced Overall Accuracy (BOA) are reported. The first two metrics are more widely known in the field of object classification as Recall (Producer's Accuracy) and Precision (User's Accuracy). Finally, the false positive rate is also reported in addition to the metrics used in CloudSEN12.

The Producer's Accuracy (PA), or Recall, is computed as the fraction of positives that are correctly detected. For an example of the cloud class, it corresponds to the number of correctly detected cloud pixels divided by the number of all true cloud pixels. It can be interpreted as an approximate probability of a pixel containing a cloud being assigned cloud class by the model. A high level of PA means that a large portion of the cloud present in the image is contained in the cloud mask.

$$\text{PA} = \frac{TP}{TP + FN} \tag{5.15}$$

User's Accuracy (UA), or Precision, corresponds to the fraction of all positive detection that are correct detections. For the example of the cloud class, it is computed as the number of correctly detected cloud pixels divided by the number of all pixels detected as cloud. It can be interpreted as an approximate probability of a pixel detected as cloud containing, in fact, cloud. A high level of UA means that a large portion of the cloud mask produced in the model contains cloud pixels, with minimal leakage of non-cloudy pixels into the mask.

$$\text{UA} = \frac{TP}{TP + FP} \tag{5.16}$$

The Balanced Overall Accuracy (BOA) is the average of True Positive Rate (Producer's Accuracy or recall) and True Negative Rate. This is particularly helpful for non-balanced datasets, where there is a significant imbalance between positives and negatives in the ground truth. The true negative rate corresponds to the ratio of all negative instances correctly labeled as negatives.

$$\text{BOA} = \frac{\frac{TP}{TP+FN} + \frac{TN}{TN+FP}}{2} \tag{5.17}$$

Finally, False Positive Rate (FPR) is provided as the rate of falsely rejected positive instances. For the example of cloud detection, it can be interpreted as the number of pixels incorrectly detected as cloud divided by the total number of cloud-free pixels.

$$\text{FPR} = \frac{FP}{FP + TN} \tag{5.18}$$

Table 5.2 contains the metrics computed on the cloudy images of the test dataset containing the original real cloudy samples. In terms of balanced overall accuracy for the cloud class, the performance is quite comparable across variants, with the model trained on real data performing best (0.79). Yet, the models (b) and (d) trained exclusively on simulated data can achieve non-trivial performance of 0.75 and 0.78, respectively. This also demonstrates the improvement achieved with realistic channel-specific magnitude (CSM) feature of the simulator. This improvement is also observed for the hybrid approaches (c) and (e), where the BOA increases from 0.73 to 0.76 for

the clear class, and from 0.71 to 0.74 for the shadow class.

These observations motivate two important conclusions. First, it is possible to train cloud detection models exclusively on simulated data and achieve good performance when tested on real samples. Second, the channel-specific magnitude appears to consistently lead to improved accuracy on real test data.

An important aspect of this test that should be acknowledged is that the real data labels have been produced by humans, who have inevitably instilled some bias into the ground truth. This bias is the net effect of many factors and may be difficult to determine precisely, however, it can be understood that any type of error consistently produced by humans leads to a certain bias in both real training and test data. Consequently, the models trained only on simulated data (b) and (d) had no access to observe this type of bias, yet are expected to reproduce it when tested on real data. Hence, they may be put at an unavoidable disadvantage. To understand this effect better, the models (c) and (e) can be inspected. In terms of BOA, these models perform marginally lower for the cloud class (both scoring 0.78 compared to 0.79 achieved with (a)), meaning that the presence of simulated data in the training samples makes it more difficult to learn the biases present in the real data.

The other metrics beyond BOA provide more insight into the results. Producer's Accuracy (PA), as discussed earlier, measures the amount of coverage for each class, which can be understood as how much of the present class is actually contained in the detected region. In this case, all of the models trained on simulated data (b)-(e) strongly outperform the real data model (a) for the cloud class. It means that those models are more likely to contain the complete set of cloudy pixels in the cloud masks they produce, which could often be considered beneficial for the purpose of masking out the cloud-affected regions. Conversely, they consistently achieve lower User's Accuracy (UA), which means that the cloud masks they produce will often contain a higher portion of cloud-free pixels.

This leads to one more conclusion, which is that the models trained with simulated data are less conservative in the process of producing cloud masks, meaning that they tend to overestimate the cloud-affected region compared to the model trained on real

data. Depending on the application, this behavior could be considered more or less beneficial (this depends on whether precise cloud-free region masks with no presence of cloud are prioritised or not).

Table 5.2: Evaluation on the real cloudy images for the cloud detection task.

Model	Trained on	Label	Test on Real: Cloudy Subset			
			BOA	PA	UA	FPR
(a)	Real	Cloud	0.79	0.68	0.85	0.11
		Clear	0.78	0.86	0.66	0.31
		Shadow	0.72	0.47	0.65	0.03
(b)	Simulated	Cloud	0.75	0.84	0.70	0.34
		Clear	0.68	0.63	0.61	0.28
		Shadow	0.50	0.01	0.67	0.00
(c)	Hybrid	Cloud	0.78	0.87	0.72	0.32
		Clear	0.73	0.60	0.74	0.14
		Shadow	0.71	0.45	0.60	0.04
(d)	Simulated with CSM	Cloud	0.78	0.75	0.78	0.20
		Clear	0.75	0.76	0.67	0.25
		Shadow	0.72	0.47	0.66	0.03
(e)	Hybrid with CSM	Cloud	0.78	0.75	0.79	0.19
		Clear	0.76	0.74	0.69	0.23
		Shadow	0.74	0.53	0.59	0.05

To support this quantitative analysis, a set of visual results produced for each model is shown in Figure 5.21, containing both thin and thick type of clouds. It can also be observed (and further confirmed by the performance achieved for the shadow class) that the model (b) appears to fail at detecting any kind of shadow. Yet, model (d), which follows the exact same training scenario but with channel-specific magnitude, achieves BOA of 0.72 for the shadow class, on par with the model (a) trained on real data, which suggests that the channel-specific magnitude feature could have a wider positive effect on the model apart from more accurate learning of the cloud class representation.

So far, testing on real data has confirmed that it is possible to train models exclusively on simulated data and achieve competitive performance on real data, as in the case of the model (d). This indicates a degree of alignment between the simulated and real data, and demonstrates a considerable value that simulated data can bring in the context of training.

However, the use of simulated data can be helpful beyond the training stage. Specif-

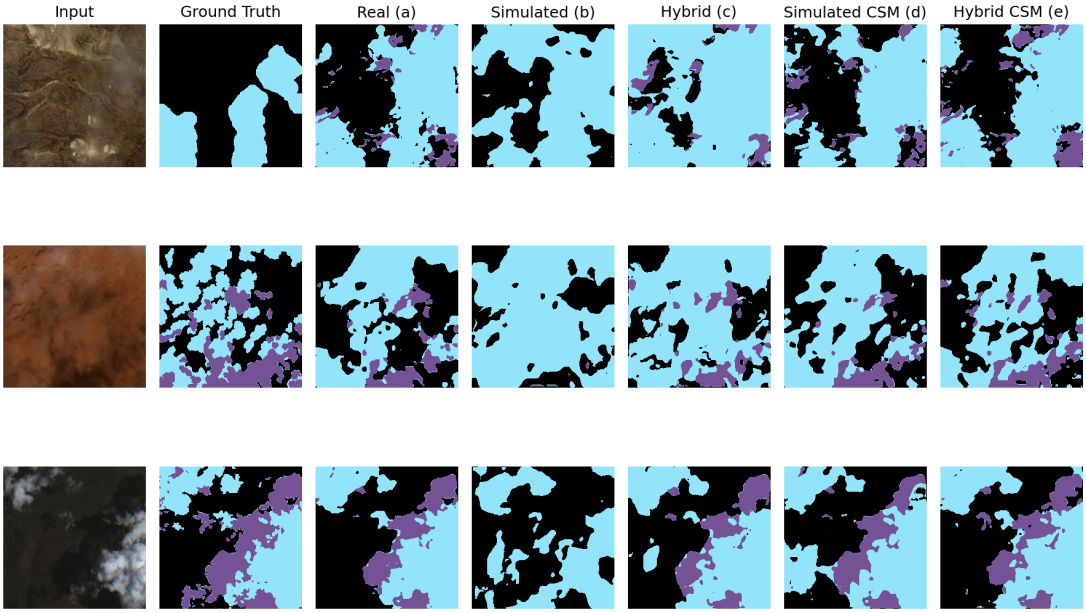


Figure 5.21: Detection models applied to three samples of real cloudy data.

ically, the cloudy images produced by the simulator can guarantee precise ground truth labels, unlike the real data annotated by humans. This is due to the access to the exact cloud transparency map during the synthesis process, which can be used to produce exact ground truth labels, as demonstrated earlier in this chapter. The models have been trained using four different configurations described earlier for several types of clouds (thick, local, thin, fog). The correspondence between the configurations and the images they produce allows to produce datasets of test images that only contain a specific type of clouds, which cannot be easily done with real data. Tables 5.3-5.6 contain metrics recorded for the four types of cloud.

The results in the tables prove that all models trained on simulated data (b)-(e) outperform the model (a) trained on real data only. This could be attributed to two effects - first, the simulated test data does not suffer from noisy ground truth like the human annotations, and second, the simulated clouds are not equivalent to the real clouds, so in this case, the models trained on simulated data had more access to adjust to that domain.

For the thick clouds test images, as shown in Table 5.3, the BOA achieved by the models (b)-(e) is higher compared to the real data, which could indicate that the

Table 5.3: Evaluation on the simulated thick cloud for cloud detection.

Model	Trained on	Label	Test on Simulated: Thick			
			BOA	PA	UA	FPR
(a)	Real	Cloud	0.78	0.58	0.99	0.02
		Clear	0.72	0.83	0.32	0.40
		Shadow	0.67	0.40	0.29	0.06
(b)	Simulated	Cloud	0.80	0.95	0.90	0.34
		Clear	0.77	0.64	0.61	0.10
		Shadow	0.51	0.02	0.37	0.00
(c)	Hybrid	Cloud	0.85	0.93	0.93	0.23
		Clear	0.76	0.60	0.62	0.08
		Shadow	0.70	0.45	0.39	0.04
(d)	Simulated with CSM	Cloud	0.81	0.66	0.98	0.04
		Clear	0.75	0.82	0.36	0.33
		Shadow	0.67	0.38	0.30	0.05
(e)	Hybrid with CSM	Cloud	0.83	0.70	0.98	0.05
		Clear	0.74	0.74	0.39	0.27
		Shadow	0.71	0.51	0.28	0.08

ground truth labels between the training and test images are more consistent. This is consistent with the fact that simulated data allows for precise ground truth extracted from the synthesis process. Another interesting observation is that the models (b) and (c) trained without channel-specific magnitude achieve the best trade-off between PA and UA, meaning that they achieve values above 0.90 for both. This means that these models can mask out most of the cloudy area and include few non-cloudy pixels in the resulting cloud mask. On the other hand, the models (a), (d), and (e) consistently achieve much higher UA than PA, which means that they tend to produce masks that mostly contain cloudy pixels, but not all of the cloudy pixels in the image do get contained in that mask. Finally, the issue with the model (b) not being able to detect shadows is still present and appears to be minimised when channel-specific magnitude is used for the clouds, as in model (d).

For local clouds, the model trained on real data still achieves superior performance of BOA at 0.80, compared to 0.77 achieved by channel-specific magnitude with only simulated data, as shown in Table 5.4. However, the models (b)-(d) trained on simulated images achieve higher PA, meaning that they are capable of including larger portions of the cloud in the resulting cloud mask.

Table 5.4: Evaluation on the simulated local cloud for cloud detection.

Model	Trained on	Label	Test on Simulated: Local			
			BOA	PA	UA	FPR
(a)	Real	Cloud	0.80	0.75	0.66	0.14
		Clear	0.74	0.80	0.82	0.32
		Shadow	0.61	0.26	0.35	0.05
(b)	Simulated	Cloud	0.76	0.89	0.47	0.37
		Clear	0.69	0.63	0.82	0.24
		Shadow	0.50	0.00	0.45	0.00
(c)	Hybrid	Cloud	0.77	0.90	0.48	0.36
		Clear	0.71	0.55	0.88	0.14
		Shadow	0.64	0.35	0.32	0.07
(d)	Simulated with CSM	Cloud	0.77	0.78	0.55	0.23
		Clear	0.71	0.71	0.81	0.29
		Shadow	0.60	0.25	0.35	0.04
(e)	Hybrid with CSM	Cloud	0.79	0.83	0.55	0.25
		Clear	0.72	0.66	0.84	0.23
		Shadow	0.63	0.32	0.31	0.07

For the thin clouds in Table 5.5, the simulated data models (b)-(d) achieve higher accuracy than the real model (a), and their PA is considerably higher, which again, suggests that they can extract a larger portion of the cloud in their cloud masks.

Visual results of the models applied to the three types of simulated data are shown in Figure 5.22.

Finally, two more subsets are tested in Table 5.6 and Table 5.7, which contain foggy image examples and cloud-free image examples, respectively. In the case of foggy images (where the entire image is covered by cloud), only User's Accuracy is reported, which corresponds to the fraction of pixels correctly classified as a cloud. In this case, all models achieve maximum performance, meaning that they assign the correct label to all examples in the test dataset.

Similarly, for the cloud-free examples, User's Accuracy is reported for the clear class and again, all models achieve maximum performance. This suggests that the foggy and cloud-free images are universally less challenging compared to the thick, local, and thin cloud types.

This concludes the analysis of the cloud detection task, which demonstrates that

Table 5.5: Evaluation on the simulated thin clouds for cloud detection.

Model	Trained on	Label	Test on Simulated: Thin			
			BOA	PA	UA	FPR
(a)	Real	Cloud	0.70	0.49	0.91	0.09
		Clear	0.67	0.83	0.41	0.49
		Shadow	0.62	0.28	0.27	0.05
(b)	Simulated	Cloud	0.72	0.87	0.79	0.44
		Clear	0.68	0.54	0.56	0.17
		Shadow	0.51	0.02	0.37	0.00
(c)	Hybrid	Cloud	0.74	0.85	0.81	0.36
		Clear	0.68	0.51	0.59	0.14
		Shadow	0.67	0.40	0.32	0.06
(d)	Simulated with CSM	Cloud	0.71	0.58	0.87	0.16
		Clear	0.69	0.77	0.44	0.40
		Shadow	0.61	0.27	0.28	0.05
(e)	Hybrid with CSM	Cloud	0.72	0.63	0.86	0.19
		Clear	0.68	0.68	0.45	0.33
		Shadow	0.65	0.38	0.25	0.08

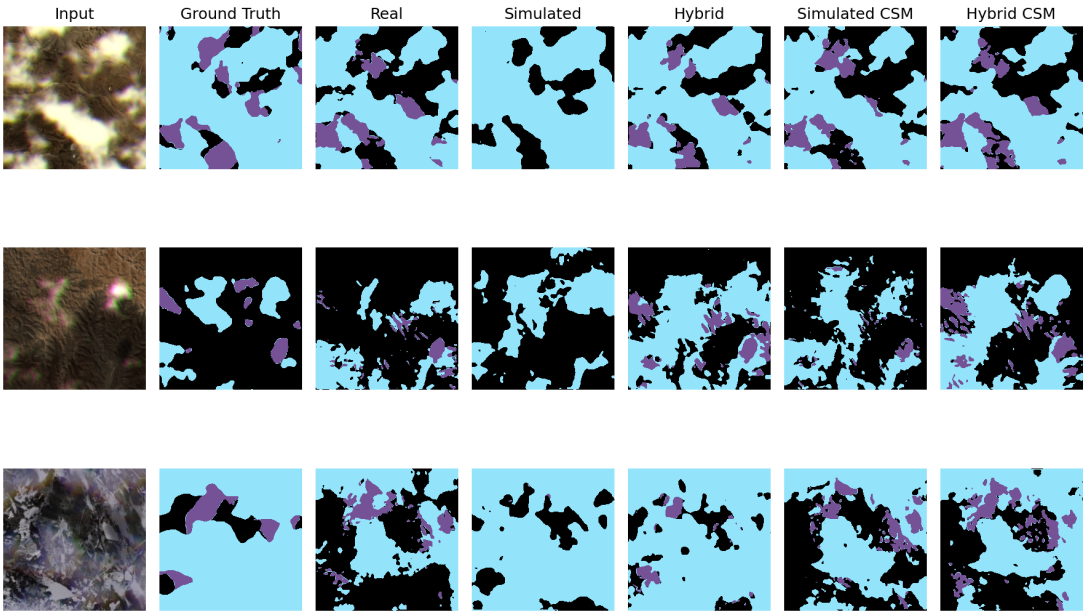


Figure 5.22: Detection models applied to three types of simulated cloudy data.

Table 5.6: Evaluation on the detection task for fog images. Producer Accuracy (PA), User Accuracy (UA) Balanced Accuracy (BOA), and False Positive Rate.

Model	Trained on	Test on Simulated: Fog UA
(a)	Real	1.0
(b)	Simulated	1.0
(c)	Hybrid	1.0
(d)	Simulated with CSM	1.0
(e)	Hybrid with CSM	1.0

Table 5.7: Evaluation on the cloud-free images for the detection task. Producer Accuracy (PA), User Accuracy (UA) Balanced Accuracy (BOA), and False Positive Rate.

Model	Trained on	Test on Real: Cloud-Free Subset UA
(a)	Real	1.0
(b)	Simulated	1.0
(c)	Hybrid	1.0
(d)	Simulated with CSM	1.0
(e)	Hybrid with CSM	1.0

cloud detection models can be trained exclusively on simulated cloudy data and achieve performance comparable to the models trained on real data. Furthermore, the realistic magnitude of the cloud component in each channel of multi-spectral data has been found beneficial for the performance on real clouds when learning from simulated data.

In the next section, a similar experiment focused on the cloud removal task is presented.

5.4 Comparison to Real Data: Cloud Removal

For the task of cloud removal, the dataset of SEN12MS-CR is used, containing real pairs of cloudy and non-cloudy Sentinel-2 images, along with corresponding Sentinel-1 samples. The dataset contains Sentinel-2 Level-1C product, which consists of 13 bands of multispectral data. Furthermore, Band 10 is excluded from the experiment, since it primarily responds to the top of atmosphere reflections of cirrus clouds [215], which has often a different effect than in other bands, as can be observed in Figures 5.23 and 5.24. This effect is not currently modelled by the cloud simulator and hence the

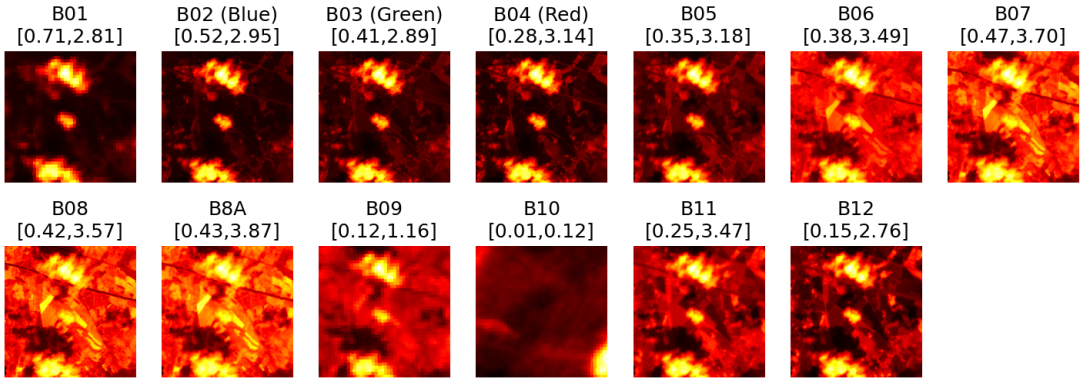


Figure 5.23: Example of content of individual multi-spectral bands in each channel for a cloudy Sentinel-2 L1C image. Intensity range for each band is listed in square brackets.

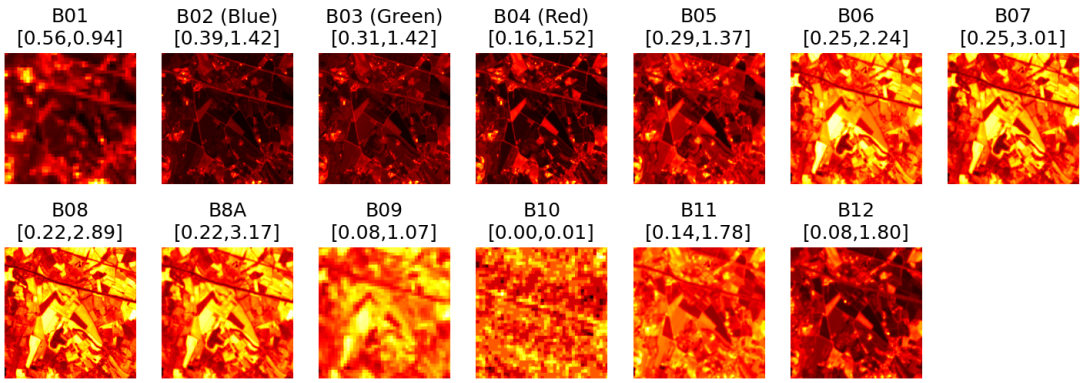


Figure 5.24: Example of content of individual multi-spectral bands in each channel for a clear Sentinel-2 L1C image.

exclusion. The two figures contain visualisations of individual bands from a Sentinel-2 L1C product for a cloudy and clear sample. In both cases, all bands except for Band-10 (short wave infrared (SWIR) – Cirrus) appear to be highly correlated. In the cloudy image, the bands tend to contain a similar presence of the cloud, while in Band-10 this object appears absent. Similarly, Band-10 in the clear image appears to detect a fairly different structure than the other bands.

The baseline architecture used for the experiments on cloud removal is DSen2-CR [77], a simple residual-based architecture consisting primarily of convolutional operations. In each case, it is trained from scratch to a point where no improvement in the validation loss occurs for 30 epochs. The networks are trained using AdamW

Table 5.8: Evaluation on the cloud removal task - (↑) SSIM Metric

Trained on	Real	Thick	Local	Thin	Fog
(a) Real	0.623/0.561	0.619/0.541	0.858/0.668	0.842/0.803	0.740/0.739
(b) Simulated	0.444/0.323	0.474/0.343	0.837/0.530	0.846/0.790	0.669/0.669
(c) Hybrid	0.603/0.538	0.619/0.531	0.873/0.666	0.859/0.814	0.746/0.745
(d) Sim-CSM	0.544/0.462	0.682/0.604	0.899/0.737	0.882/0.846	0.755/0.753
(e) Hyb-CSM	0.567/0.485	0.670/0.595	0.889/0.728	0.872/0.839	0.749/0.748

optimiser [214] with a starting learning rate of 10^{-4} and the same decay strategy as the cloud detection model above. Each batch of data contained 4 clear and 4 cloudy images, and, similarly to the cloud detection scheme, the loss on the clear images is multiplied by a factor of 0.1. Due to the large dataset size of SEN12MS-CR, during each epoch, the loss is optimised on 1000 random samples from the training dataset, and the validation loss is computed on 500 random samples from the validation dataset.

Similarly to the previous example with cloud detection, the models are tested on 5 different test datasets, one with real clouds and another four with simulated-only data of different cloud types. The commonly used metrics of SSIM (Table 5.8) and RMSE (Table 5.9) are used to evaluate the models. Furthermore, each metric is reported for the whole image (the first listed value) as well as the isolated cloud-affected region (the second listed value).

The results in Table 5.8 indicate that while the model trained on the real data (a) performs best on that type of data (SSIM of 0.623 for the whole image and 0.561 for the inpainted region), the model (d) trained exclusively on simulated data with channel-specific magnitude achieves SSIM 0.544/0.462. On the other hand, for any type of simulated test data, model (d) outperforms model (a). Similarly to the detection task, model (b) trained on simulated cloudy images without channel-specific magnitude consistently produces results of the lowest quality. In Figures 5.25 and 5.26, it can be observed that the model (b) does not really apply any visible changes to the input image, indicating that it does not recognise the cloud objects as something that should be removed. It achieves a SSIM of 0.444/0.323, compared to 0.544/0.462 the equivalent model trained with channel-specific magnitude.

In the case of the cloud removal task, the conclusions are similar to the earlier

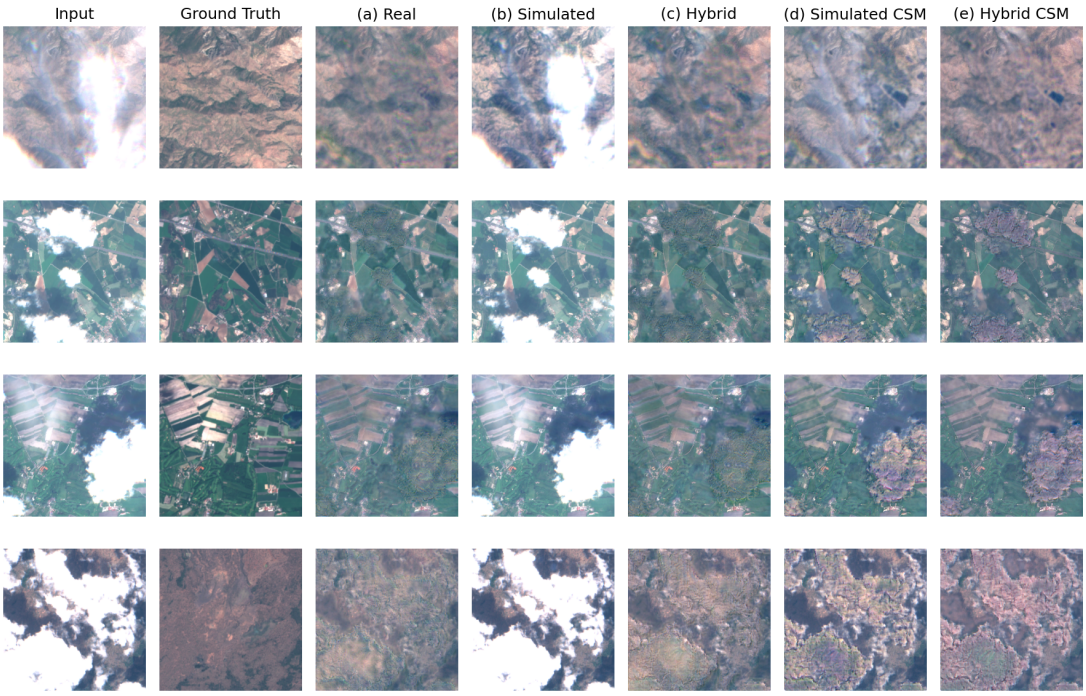


Figure 5.25: Examples of model output on real images.

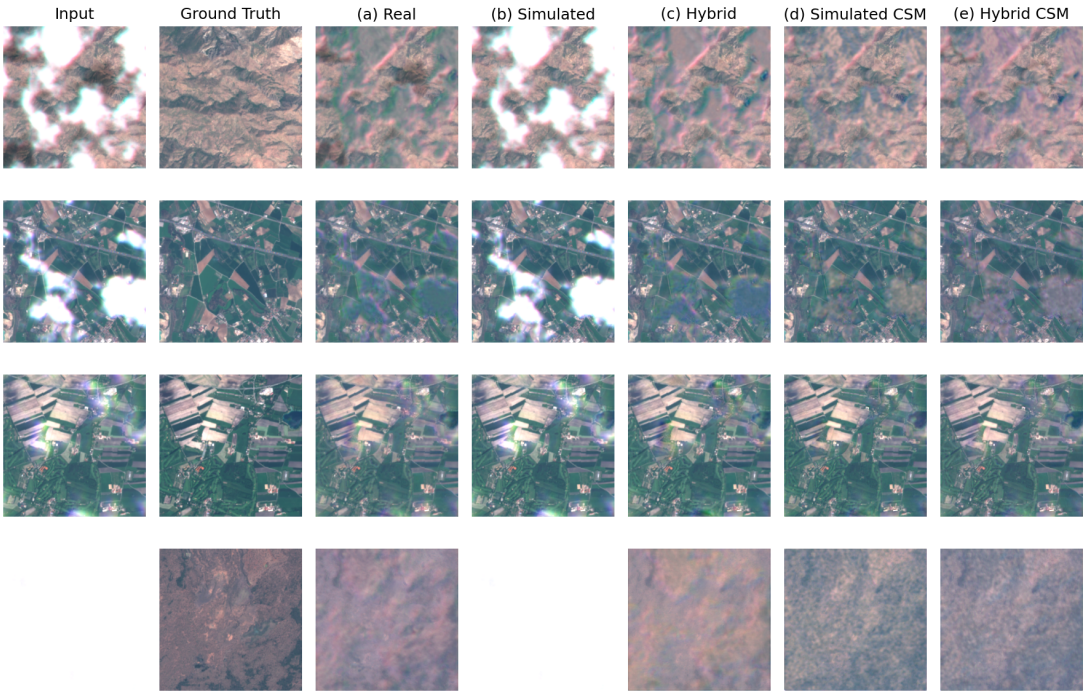


Figure 5.26: Examples of model output on the four types of simulated cloudy images.

Table 5.9: Evaluation on the cloud removal task - (↓) RMSE Metric

Trained on	Real	Thick	Local	Thin	Fog
(a) Real	0.217/0.229	0.289/0.323	0.142/0.233	0.165/0.187	0.266/0.266
(b) Simulated	0.884/1.042	1.108/1.271	0.330/0.628	0.300/0.364	0.802/0.802
(c) Hybrid	0.227/0.236	0.303/0.344	0.136/0.247	0.162/0.193	0.276/0.276
(d) Sim-CSM	0.261/0.264	0.198/0.223	0.099/0.170	0.126/0.146	0.196/0.196
(e) Hyb-CSM	0.238/0.251	0.209/0.235	0.107/0.180	0.134/0.154	0.209/0.209

example of cloud detection. It is possible to train a model to remove clouds from satellite imagery, provided that channel-specific magnitude is applied during synthesis. There remains a gap between the real and simulated data, meaning that models trained exclusively on the real data do not perform as well on simulated data, and vice versa.

5.5 Summary

The issue with the quality of the cloud-free ground truth in the real data has motivated the introduction of SatelliteCloudGenerator. It has been designed for fast computation on GPU and for high control over the cloud appearance. The quality of the simulated data has been tested by training cloud detection and cloud removal models, demonstrating that the models trained exclusively on simulated data can perform the same task on real data, especially if the channel-specific magnitude effect is employed. The performance of these simulation-based models never reached the same level of performance on real data as models trained on real data, which could be due to two effects. First, the models trained on real data have the advantage of accessing real input cloudy samples, unlike the models trained only on the simulated images, which could indicate a gap between the appearance of real and simulated clouds. The second effect is the bias of human annotators for cloud detection or the bias of changes occurring on the ground for the cloud removal data. This means that the model trained on real data could have had the opportunity to learn the accurate features in the real data, but also the biases in it. The latter could lead to good performance on the real test data, but be harmful when generalising beyond the curated dataset. On simulated test data, the model trained with the simulated data (again, channel-specific magnitude is crucial)

performs better than the model trained on the real data. This again, could be an effect of the biases in the simulated data (due to the nature of the simulation framework) or the result of the more precise ground truth in simulated data, which should influence both during the training and evaluation stage.

Ultimately, neither real nor simulated data appears to be universally more advantageous. The ideal is an abundant source of real images with precise ground truth. The real image sources are not abundant and are likely to contain distorted ground truth. The simulated sources are abundant and with precise ground truth, but a certain gap between real and simulated images can be expected. Given this state of matters, a good way to achieve a trade-off is to perform both training and evaluation on both data sources, which is also what the next chapter describes.

Chapter 6

Comparison of Different Learning Levels

The Chapters 3 and 4 have proposed two novel learning modes for satellite image inpainting; internal learning and learning from language. These two approaches are different from the most often employed type of deep learning, which is based on optimizing the network parameters on a large set of data tailored for that specific task. This mode of learning is referred to as external learning.

The primary aim of this chapter is to compare the explored learning modes and based on that, draw some conclusions about the strengths and weaknesses of each variant. To facilitate such a comparison, a common task pipeline must be defined. In the chapters on the topics of internal learning and language-based learning methods, the focus has been put on the inpainting part of the process. Hence, in order to remove clouds, these techniques are now combined with cloud and shadow detection to act as the inpainting mask. In the context of the more general cloud removal task, a cloud mask must be obtained, which could be computed using an existing cloud detection technique.

Furthermore, apart from different learning regimes, this work also explores the potential of using both multi-source (radar) and multi-temporal (historical) data to increase the quality of cloud-free prediction. The complete comparison is made on a common test dataset.

The chapter begins with a description of the design of the dataset, taking into consideration both testing and training since the external network will require a separate set of data for training. The dataset takes advantage of the simulation framework proposed in Chapter 5.

This relates to the second contribution of this chapter, which is the externally trained network. Despite the presence of several well-performing cloud removal models, there is no available model that can flexibly accept multi-source and multi-temporal guidance input at inference. In order to fill that gap and enable comparison with the other two variants of learning, a new type of externally trained deep neural network model is defined.

These two elements, the common test dataset and an externally trained network lead to the final set of results in this work, which contains a performance evaluation of each learning type; internal, external, and learning from language.

6.1 Dataset

The dataset is built based on the openly available SEN12MS-CR-TS [169]. In order to test the cloud removal performance in a setting with access to multi-source and multi-temporal guidance, a dataset containing this type of matched data is necessary. The majority of publicly available datasets do not satisfy this condition. The majority of datasets will contain only cloudy images paired with cloud-free (e.g. RICE1 [207] or STGAN [64]), and a few with accompanied radar data (e.g. SEN12MS-CR [67], SEN12MS-CR-TS [169], CloudSEN12 [11]). However, only SEN12MS-CR-TS contains both radar data and historical optical data. This is delivered as 30 samples for each location containing both Sentinel-1 and Sentinel-2 acquisitions. The acquisitions cover 53 regions of interest (ROIs), corresponding to a total of 80,000 square kilometres of global coverage. This work follows the same splitting approach as in the original dataset, where 40 of the ROIs are used for training purposes, while 13 are kept out as a test set. Each sampled ROI yields 30 images evenly distributed across the year 2018. Finally, each scene is split into patches (with no overlap) of 256 by 256 pixels.

An example collage for a single patch with 30 captures of both Sentinel-1 and

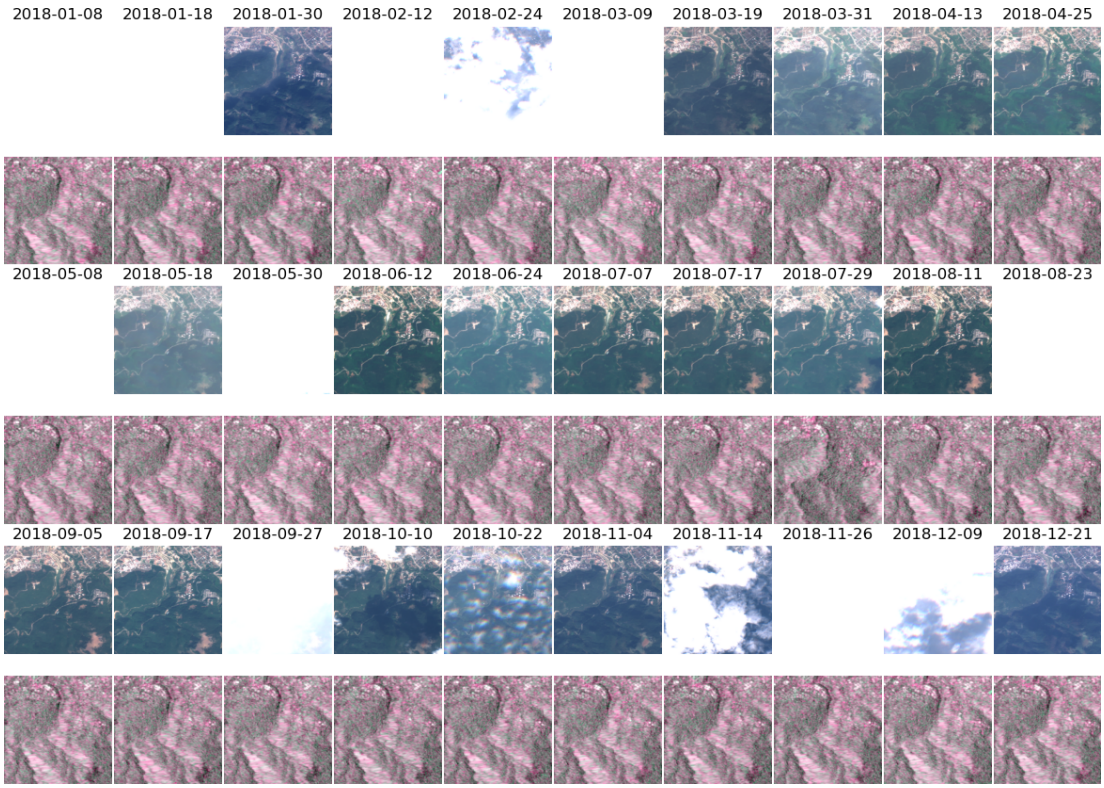


Figure 6.1: A patch from the test dataset containing 30 acquisitions throughout the year. The white patches are oversaturated due to cloud presence. The dates correspond to the Sentinel-2 samples. The bottom row contains a false-colour image of the Sentinel-1 data.

Sentinel-2 images from a single year is shown in Figure 6.1. Sentinel-1 data is displayed as false colour with VV, VH, and the mean of the two assigned to the RGB display channels. Sentinel-2 images are scaled in the same way as described in the dataset manuscript [169].

Following the conclusions of the previous chapter on simulation, both real and simulated data pairs have unique disadvantages. Real cloudy data does not have accessible clear-sky ground truth, while the simulated data will always be an approximation of what real clouds should look like. For that reason, both types of cloudy samples are considered in training and testing. The simulated cloudy samples are generated with the available cloud-free data as input.

Test Subset

For the test subset, the main objective is to obtain a static set of samples suitable for evaluation. As described below, the initial set of 3,716 test patches undergoes a filtering stage, followed by a manual inspection.

Step 1: Selection of Valid Patches

First, the patches (each containing the record of 30 optical and radar samples) are filtered based on the following conditions:

- A single cloudy sample with cloud coverage between 0.1 and 0.9, followed or preceded by a cloud-free sample (treated as ground truth) exists for that patch.
- Another cloud-free sample has been captured before the ground truth cloud-free image that can be used as a historical sample, if more than one historical image exists, the oldest one is selected.
- SAR samples are extracted for all three optical samples.

This reduces the initial number of 3,716 patches to 1,222. Each of the 1,222 samples contains a pair of cloudy and cloud-free images as well as a historical cloud-free image from the past, all accompanied by temporally proximate SAR acquisitions. This allows for testing the capability of the cloud removal methods to transfer different types of knowledge (historical and cross-sensor).

Step 2: Manual Output Filtering

To ensure the high quality of the testing samples, each image from the filtered set of 1,222 patches is manually inspected. It can be observed, that some of the cloud-free images contain very little variance and are almost trivial, therefore, not particularly useful for evaluation. These images most often contained water, green areas, or were outside of the acquired tile and hence black, as shown in Figure 6.2.

After filtering out the cloud-free images with low-variance, a similar approach was applied to the historical samples, and the samples in Figure 6.3 were further excluded.

Figure 6.2: Manually removed cloud-free samples due to conditions such as water, cloud haze, invalid black pixels.

Figure 6.3: Manually removed historical cloud-free samples due to conditions or cloud haze or invalid black pixels.

Figure 6.4 shows cloudy samples excluded from the dataset due to the lack of apparent clouds or tile margin effects.

The manual inspection has reduced the number of patches from 1,222 to 1,031. Examples of samples included in the dataset are shown in Figure 6.5 (cloud-free), Figure 6.7 (cloudy), and Figure 6.6 (historical).

The resulting triplets (of matched Sentinel-2 and Sentinel-1 captures) are shown in Figure 6.8. Depending on the region, the acquisition dates, and the type of scene present on the ground, the historical Sentinel-2 image is more (like in the fourth row) or less similar to the cloud-free ground truth (like in the second row).

Simulation

For each triplet in the resulting test dataset, another image with simulated clouds is generated using the cloud-free sample. This is done using the simulation tool described in the previous chapter, with the channel-specific magnitudes extracted from the real cloudy image of the triplet. This allows to obtain a set of cloudy samples with realistic

Figure 6.4: Manually removed cloudy samples due to conditions such as absent cloud or invalid black pixels.

Table 6.1: Parameters used for the simulated test clouds

Parameter	Value
Locality Degree	[1,2,3]*
`max_lvl`	*
`min_lvl`	0.0
`const_scale`	True
`cloud_color`	True
`clear_threshold`	0.1
`channel_offset`	2
`blur_scaling`	2

features and comparable conditions to those observed for real samples.

The settings used to simulate the clouds are listed in Table 6.1. The locality degree ranges between 1 and 3, depending on the percentage of the cloud cover in the real sample (to achieve a more representative set of simulated clouds). If the real cloud percentage in the triplet is above 60%, the degree is selected as 1. For values larger than 40% but no higher than 60%, the value of 2 is used. For the remaining range, locality degree of 3 is used. Note that this assignment has been selected intuitively through experimentation since the locality degree does not exhibit a precise relationship to the percentage of coverage due to the complexity and randomness of the synthesis process. Examples of clouds generated for the test set using the described process are shown in Figure 6.9.

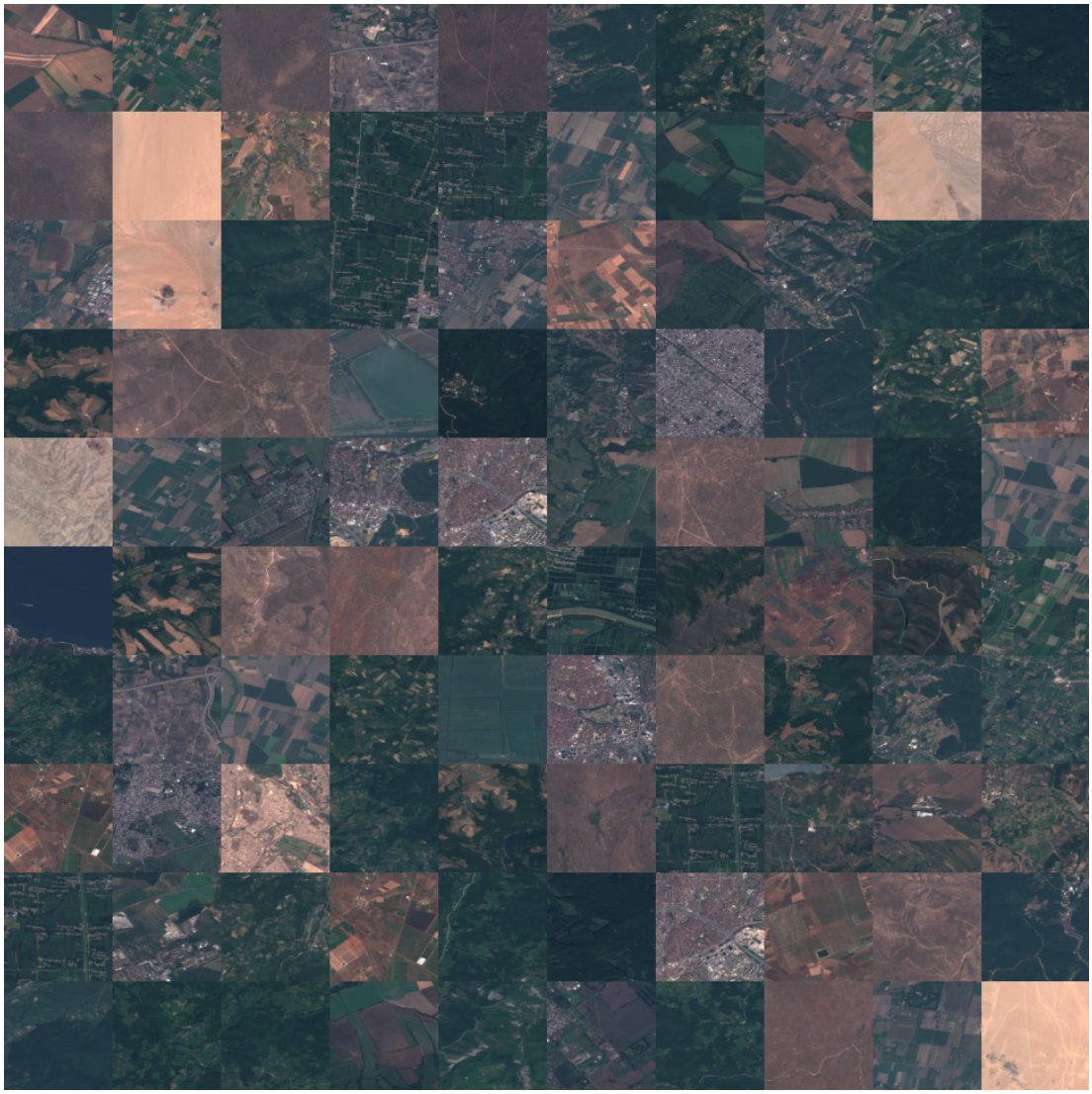


Figure 6.5: Subset of included cloud-free samples.

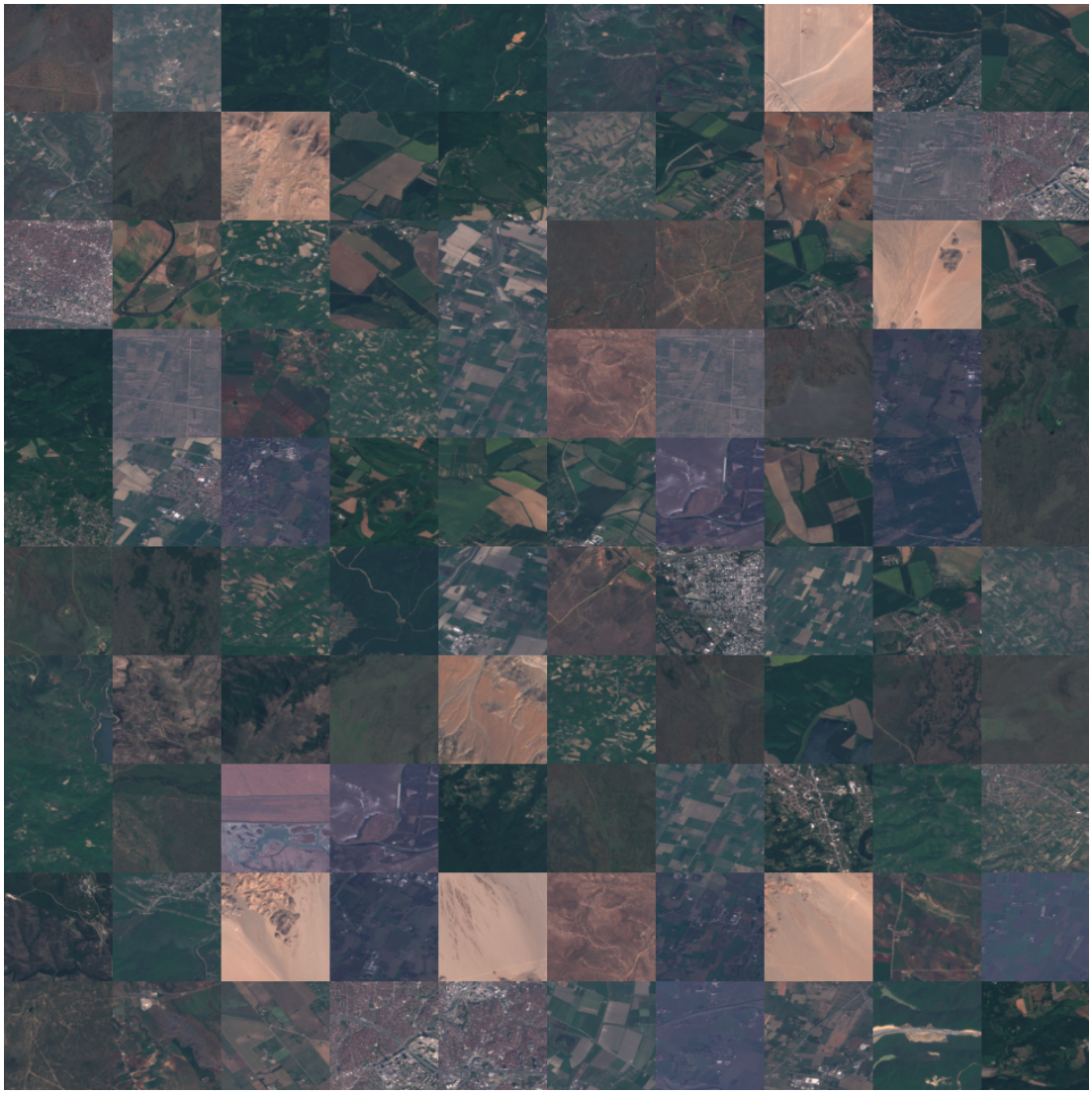


Figure 6.6: Subset of included historical cloud-free samples.

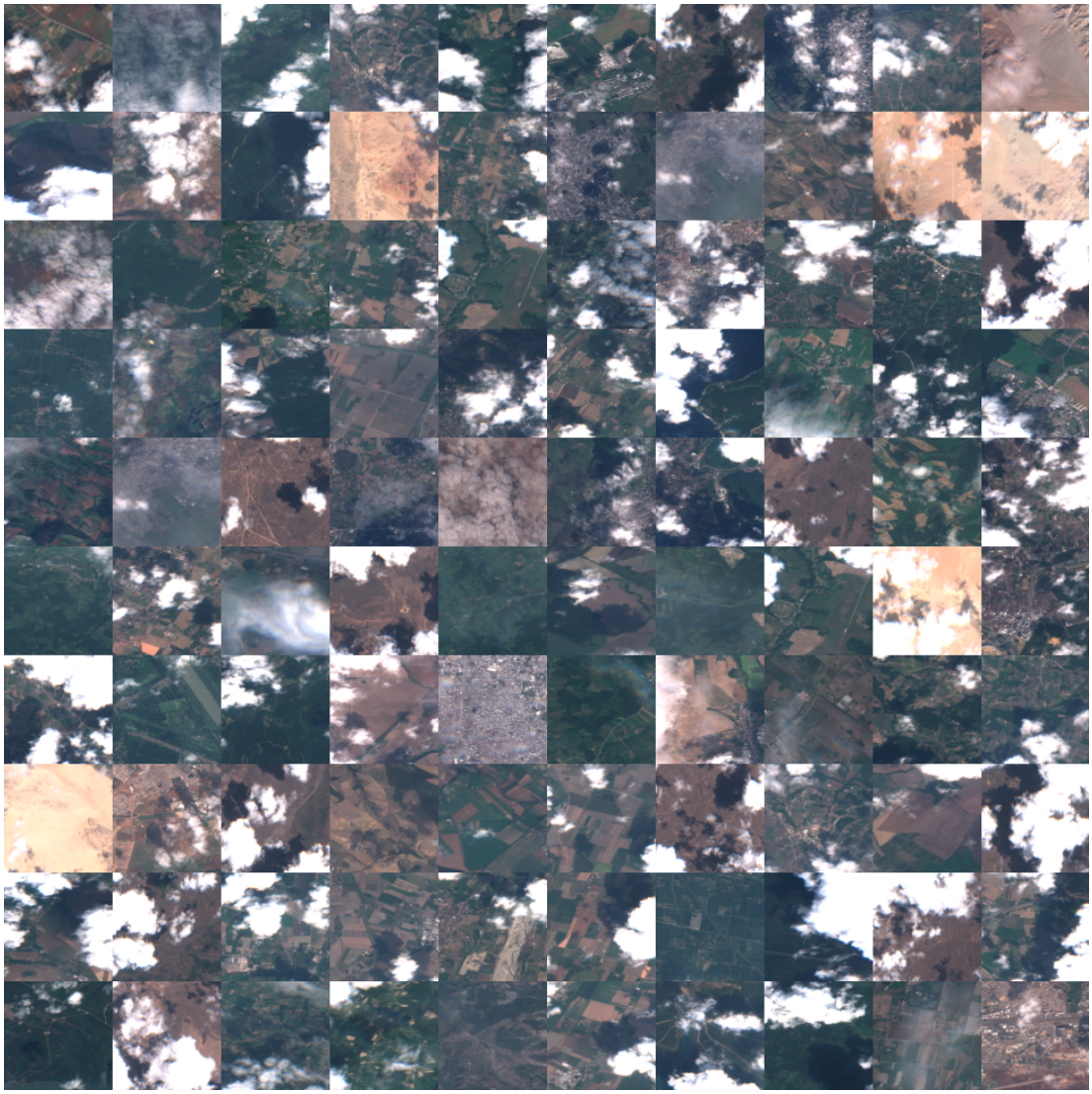


Figure 6.7: Subset of included cloudy samples.

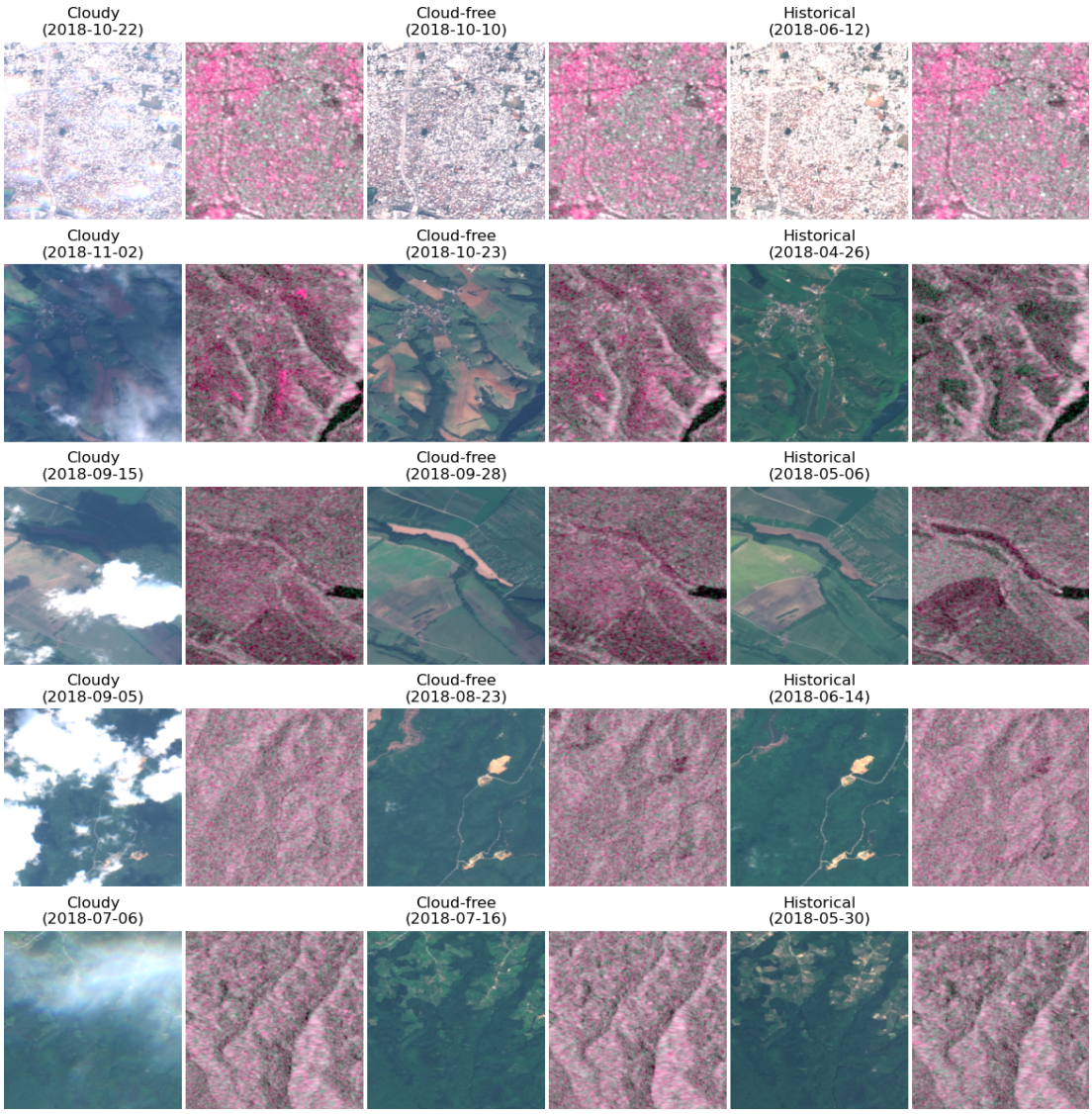


Figure 6.8: Examples of cloudy, cloud-free, historical triplets present in the filtered test dataset.

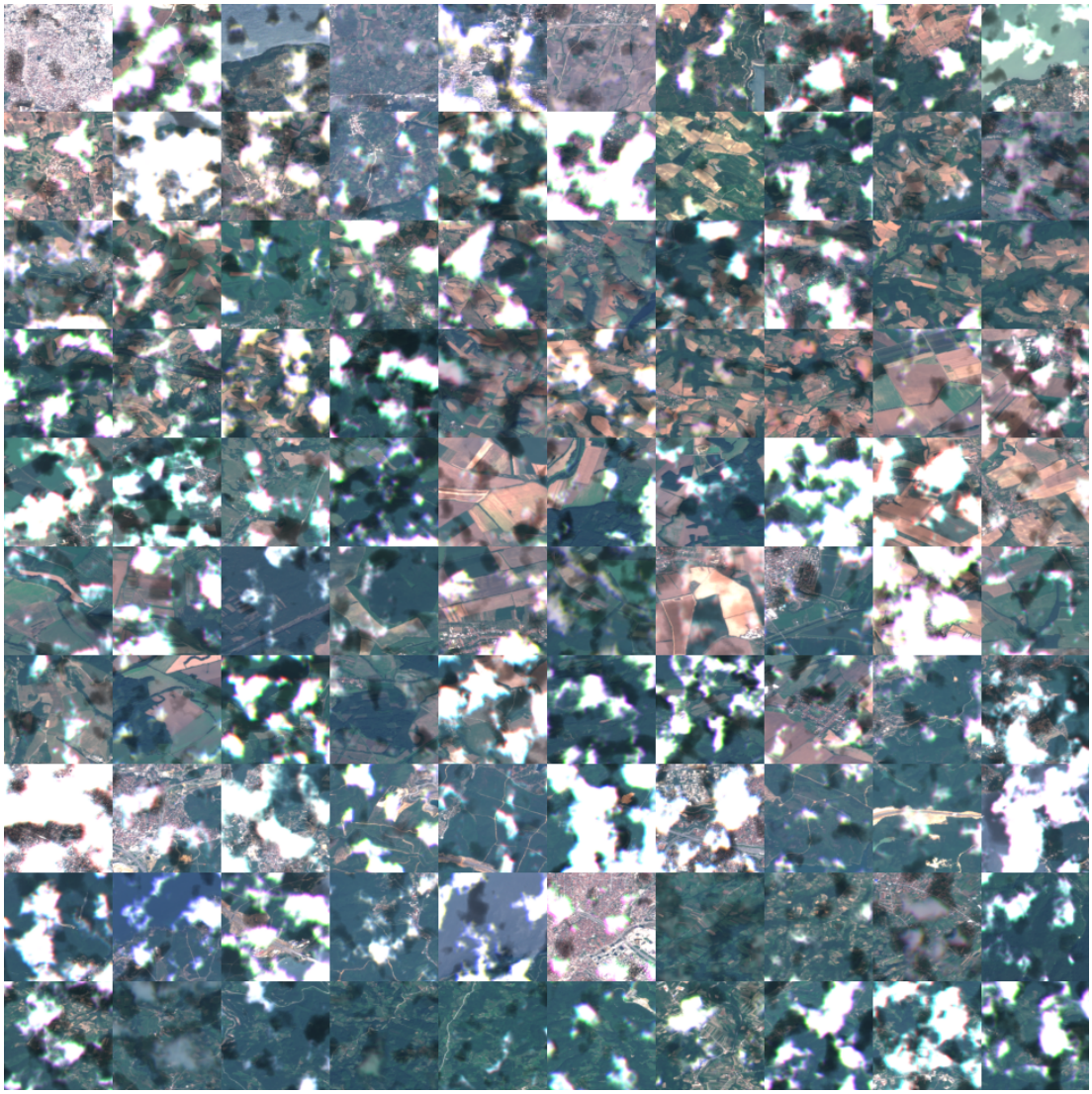


Figure 6.9: Subset of simulated samples.

Training Subset

For training, the simulation tool can be used by generating new clouds during training and that is the approach followed in this work. However, the challenge of converting the 30 samples from 2018 in each patch to a set of triplets still remains.

In the case of test subset, the historical sample was selected to maximise the distance in time to the cloud-free ground truth image (and hence increase the difficulty of the task). For training, the volume of data is prioritised over the difficulty of the samples, and hence, if more than one triplet is available from a single series of 30 images, it will be extracted.

The possible triplets are sampled in the following way. First, there must exist at least two cloud-free samples, where the latter one has to be neighbouring with a cloudy sample. If this condition is not met, the patch is not a suitable source. Otherwise, the indices of all cloud-free samples are saved as potential historical sources. Then, the indices of all neighbouring pairs of cloudy and cloud-free are saved too. With those two lists, a triplet can be sampled by first i) selecting a cloudy-clear pair from the second list, and then ii) selecting a historical sample from the indices in the historical list that precede the indices selected in the first step.

Given the size of the training subset, manual filtering has not been performed under the expectation that noisy samples in the training dataset will not prevent successful learning of the task with a sufficiently large batch size and good training practice.

6.2 External Learning Model

To compare the internal learning and learning with language approaches to a more standard approach, a new type of model must be designed that will accept the same type of data (historical and radar guidance images) since there have been no such models proposed in the literature.

The starting point for the proposed model is the convolutional residual network of DSen2-CR [77], which is trained to predict the residual between the cloudy Sentinel-2 image and the cloud-free Sentinel-2 image. As shown in Figure 6.11, the regular DSen2-

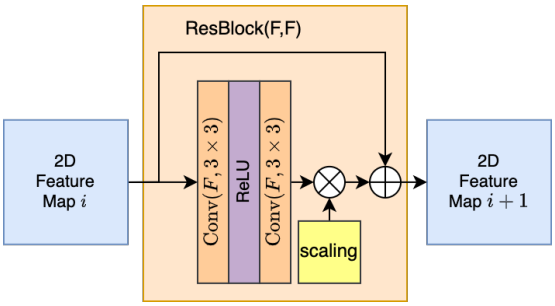


Figure 6.10: Diagram of a residual block used for the DSen2-CR architecture.

CR network takes the concatenated Sentinel-2 and Sentinel-1 image from the cloudy sample as input, and returns the approximated residual of Sentinel-2 as output. The key element of the architecture is the residual block, shown in Figure 6.10.

The residual block takes a 2D feature map with F channels, and feeds it through a convolution layer, followed by a ReLU activation, and another convolution (all of which operate on F-dimensional representations). The output of the second convolution layer is multiplied by a scaling factor (used for training stabilisation [216], equal to 0.1 by default) and added to the input feature map. Hence, the convolutions are responsible for predicting a residual representation.

The complete flow inside the network includes an initial mapping from the concatenated Sentinel-2 and Sentinel-1 channels[1] to 256 dimensions, followed by a ReLU activation, and then 16 residual blocks. Finally, another convolutional layer transforms 256-dimensional output feature to 12 channels of Sentinel-2, which is added to the input (and potentially cloudy) Sentinel-2 image.

Naturally, this approach is constrained to accept Sentinel-1 as the only source of additional guidance. To allow both Sentinel-1 radar data as well as historical Sentinel-2 input, two lightweight encoding modules are added to the DSen2-CR, as shown in Figure 6.12, in order to populate a guidance image with 2-channels with appropriate data. This approach allows for flexible use of either Sentinel-1 or Sentinel-2 guidance data, depending on the type of supporting data available during inference. This way, the network can be used like the regular DSen2-CR network that only takes Sentinel-1

[1]Note that in this work, 12 channels of the Sentinel-2 representation are used, instead of 13 like in the DSen2-CR work.

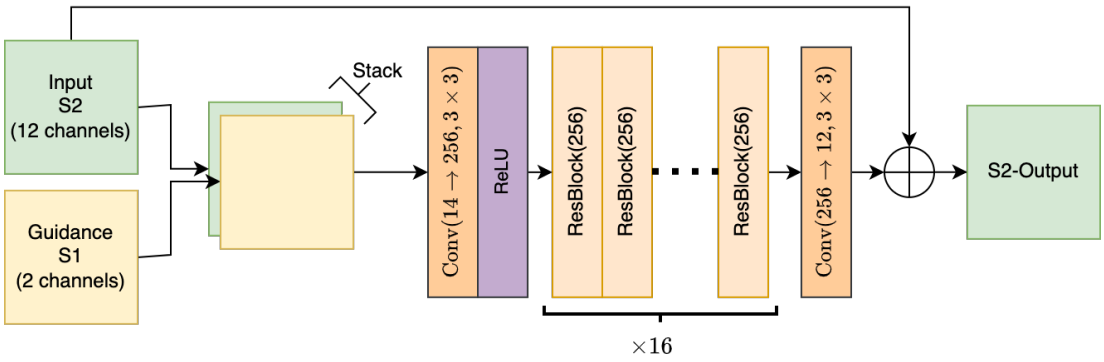


Figure 6.11: Architecture of DSen2-CR, capable of accepting both Sentinel-1 and Sentinel-2 guidance for the cloud removal task.

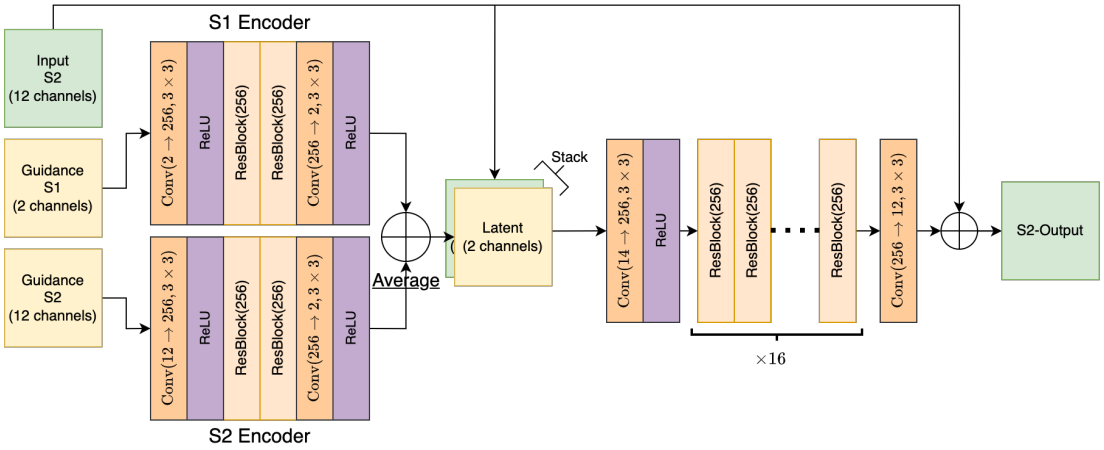


Figure 6.12: Architecture of Multi-DSen2-CR, capable of accepting both Sentinel-1 and Sentinel-2 guidance for the cloud removal task.

data, but just as well, it could also take Sentinel-2 historical data (or even multiple historical samples). This is because both Sentinel-1 and Sentinel-2 encoders used for pre-processing compute output with the same number of channels and can therefore be combined by averaging.

The resulting averaged 2-channel representation is concatenated with the input Sentinel-2 data subject to cloud removal. The rest of the architecture is the same as in DSen2-CR with 16 residual blocks of 256 channels, and another final convolution layer for channel conversion.

The network is trained with both real and simulated cloudy data (each used ran-

domly with 50% likelihood) and 4 types of simulator configurations used in the simulation chapter. Furthermore, to support different combinations of guidance data, the network is fed with either i) radar-only guidance, ii) historical-only guidance, or iii) both types of guidance, each with equal probability.

The network is trained using an AdamW optimiser with a learning rate of 10^{-4} and a batch size of 16. A learning rate scheduler is employed to divide the learning rate value by a factor of 10 if the training loss does not decrease for 2 epochs. Finally, the checkpoint with the lowest validation loss is used as the final model, which in this case occurred after 626 epochs.

6.3 Comparison

With the external cloud removal model trained, it is now possible to identify 3 types of solutions:

- The **internal** learning approach tested is the MCPN network (Emergent type) taking historical and radar data as supporting signals and optimizing with the standard settings for 4,000 steps. Cloud detection of s2cloudless is used to generate the masks for the real data, while exact cloud-shadow masks are available for the simulated data.

- The **external** network is precisely what has been described in the previous paragraphs.

- The **language** model follows the main inpainting approach from the language chapter, which involves edge-guided inpainting of RGB channels, followed by DIP-based multi-channel filling with historical and radar data stacked on top of each other. Similarly to the internal approach, cloud detection of s2cloudless is used to generate the masks for the real data, while exact cloud-shadow masks are available for the simulated data.

Before comparing the performance, it is important to identify the technical differences between the three tested approaches, from the usage perspective. Table 6.2

Table 6.2: Key Features of Compared Cloud Removal Solutions

Method	Internal	External	Language
Mask-Free	✗	✓	✗
Modality-Agnostic	✓	✗	✗
Inference Speed	Low	High	Low

contains a summary of the key technical aspects of the cloud removal techniques. First, the use of the externally trained network makes it easy to remove the requirement for a mask, unlike the other two methods focused on inpainting. However, the internal approach has the advantage of freely allowing any type of input data, for both cloudy input and guidance images, while the externally trained network is constrained to only operate on the type of data it has been trained on. The language-based pipeline incorporates a multi-channel fill that has the flexibility of the internal learning regime, but it still requires the RGB bands to be present in the main representation so that edge-guided inpainting with StableDiffusion can be performed. Finally, the inference speed for the external network is considerably higher than in the case of internally optimised solutions, which could often take 2-3 orders of magnitude longer to compute.

The first comparison is performed on the test subset with real cloudy images. As discussed in the simulation chapter, it must be acknowledged that the quality of the ground truth could be lower in this case since the cloud-free reference comes from about 10 days before or after the cloudy image. However, the advantage of this subset is that the models are evaluated on the removal of real instances of clouds.

Table 6.3 contains the metrics achieved for all 12 MSI channels contained in the representation (mean and standard deviation are reported). In this case, the externally trained model achieves considerably higher performance (mean inpainting SSIM of 0.743) compared to the two alternative methods (0.567 and 0.563). Among those two, the internal learning approach achieves slightly better performance for the inpainting region (0.567 as opposed to 0.563), yet lower whole image SSIM, indicating distortion of the known region of the image.

Interestingly, the performance for the RGB is generally higher for the internal and external techniques, but not for the language. The latter is attributed to two effects.

Table 6.3: Evaluation on the cloud removal task - Real test subset, 12 MSI channels

Method	SSIM-Whole	SSIM-Inpainting	RMSE-Whole	RMSE-Inpainting
Internal	0.650±0.107	0.567±0.106	0.301±0.177	0.384±0.307
External	**0.786**±0.064	**0.743**±0.066	**0.135**±0.046	**0.142**±0.046
Language	0.665±0.114	0.563±0.094	0.245±0.114	0.276±0.120

Table 6.4: Evaluation on the cloud removal task - Real test subset, RGB channels

Method	SSIM-Whole	SSIM-Inpainting	RMSE-Whole	RMSE-Inpainting
Internal	0.714±0.119	0.657±0.126	0.168±0.091	0.197±0.149
External	**0.834**±0.064	**0.798**±0.070	**0.079**±0.032	**0.083**±0.033
Language	0.664±0.135	0.561±0.129	0.146±0.047	0.165±0.044

First, the language model has already been identified as prone to adding a lot of new features to the images, often resembling satellite images of Earth taken from a larger distance, instead of cropped patches, leading to distortions in the RGB bands. After the RGB inpainting, the MSI DIP-based filling is used, which tends to first reconstruct the lower frequency content of the image, which could act as a filter that removes some of the distortions introduced in the RGB band and hence, increasing the performance in the non-RGB bands.

For the models performing better in the RGB region, it could be that those bands are generally easier to process and hence, the performance is considerably higher with the internal method achieving 0.657 SSIM of the inpainting as opposed to 0.567 for all bands, and the external method reaching 0.798 instead of 0.743.

These results are illustrated in Figure 6.13, with several randomly picked samples from the test dataset. The externally trained network introduces less distorted output. Internal learning will often converge to distorted representations of the image, like in the first and last rows. Furthermore, it is more prone to produce distortions in some types of signals, such as dark spots in the cloudy region. The same issue can be observed in the output of the language-based technique but to a lesser degree.

For the simulated clouds, similar effects are observed, as shown in Table 6.5 (12 channels) and Table 6.6 (RGB), but the margin between the best performing external approach and the internal learning approach is smaller, which could likely be attributed to the better quality of the inpainting masks. This is because the simulated subset

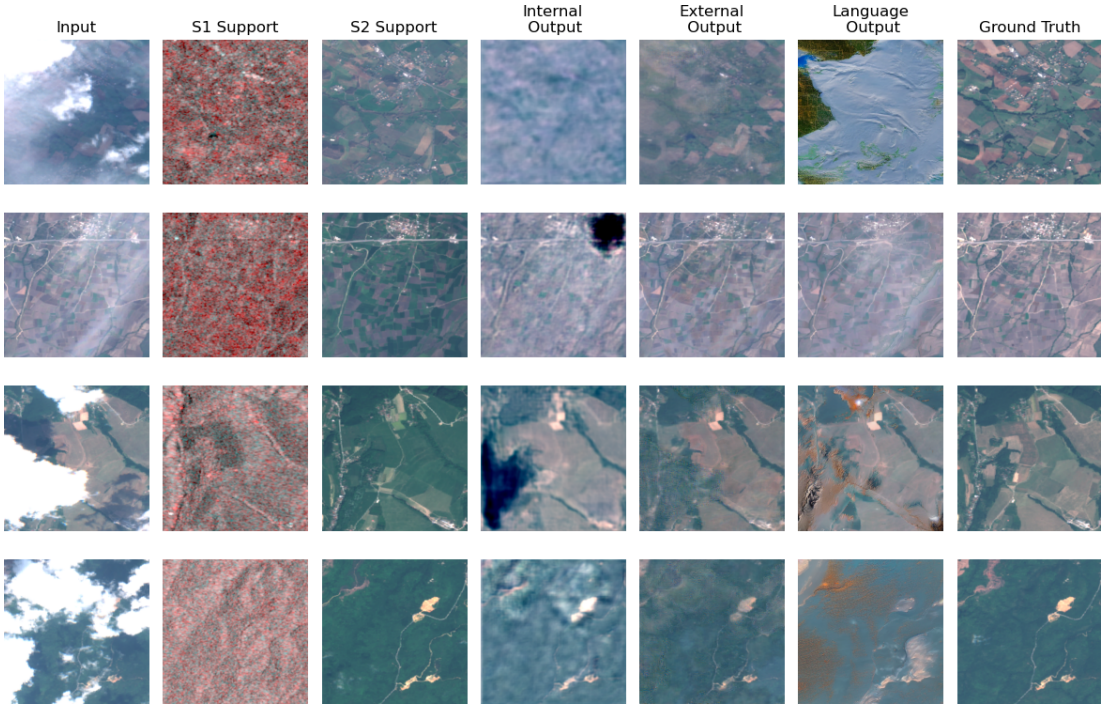


Figure 6.13: Output of the tested techniques for real cloudy images.

contains the exact cloud and shadow masks, which allows to eliminate the distortions in the cloud-free region that can occur for real cloudy samples. The external model does not rely on the provision of masks, so this effect will not apply in that case. Finally, the RMSE metric appears to be more favourable for the Internal approach here than for the External learning method. This, again could be an effect of the precise cloud mask provision, which prevents leakage of the cloudy component to the output of the Internal approach (this generally occurs in the case of real data). This way, simple solutions that are close to the mean colour but lack precise details could outperform the External technique. However, for the SSIM metric, which is dependent on the local correlations of the two images, the External learning approach is still optimal of the three. This demonstrates a certain bias of the commonly used metrics and the need for structure-based evaluation measures, such as SSIM.

Again, the performance for the RGB bands is consistently higher for all three methods (in this case, also for the language-based approach). This could be the result of the more precise masking in the case of the two methods relying on the provision of

Table 6.5: Evaluation on the cloud removal task - Simulated test subset, 12 MSI channels

Method	SSIM-Whole	SSIM-Inpainting	RMSE-Whole	RMSE-Inpainting
Internal	0.775±0.081	0.775±0.083	**0.151**±0.095	**0.148**±0.097
External	**0.805**±0.104	**0.805**±0.104	0.187±0.087	0.186±0.089
Language	0.707±0.101	0.642±0.087	0.235±0.112	0.261±0.120

Table 6.6: Evaluation on the cloud removal task - Simulated test subset, RGB chaannels

Method	SSIM-Whole	SSIM-Inpainting	RMSE-Whole	RMSE-Inpainting
Internal	0.813±0.084	0.815±0.086	**0.079**±0.059	**0.076**±0.052
External	**0.847**±0.093	**0.848**±0.094	0.106±0.069	0.104±0.070
Language	0.776±0.123	0.777±0.123	0.092±0.039	0.090±0.039

the mask, as well as the simulated cloud removal being a potentially easier task.

These observations can be further supported by the visual results shown in Figure 6.14. It appears that the external model does not perform as well with simulated clouds, while the internal and the language methods produce quite consistent output owing to the precise masking of the cloud-affected region. The convergence issues for MCPN are still present like in the sample in the second row.

6.4 Summary

This concludes the comparison experiments, with the externally trained model being a clear winner in terms of performance. However, there is still more to the problem than mere performance on a single dataset. The two methods based on internal learning and language-based learning achieved worse (yet non-trivial) performance, but most importantly, both have been used to perform the task without ever training on it. This means that these two approaches can be used for a variety of different modalities, while the external model accepts only 12-channel data with guidance from 12-channel optical and 2-channel radar signals, and cannot remove clouds from different representations of satellite images.

The context of using two types of paired cloudy data, real and simulated, is also important. There is a trade-off associated with not only training but also evaluation, which ultimately determines the conclusions that can be drawn about the models'

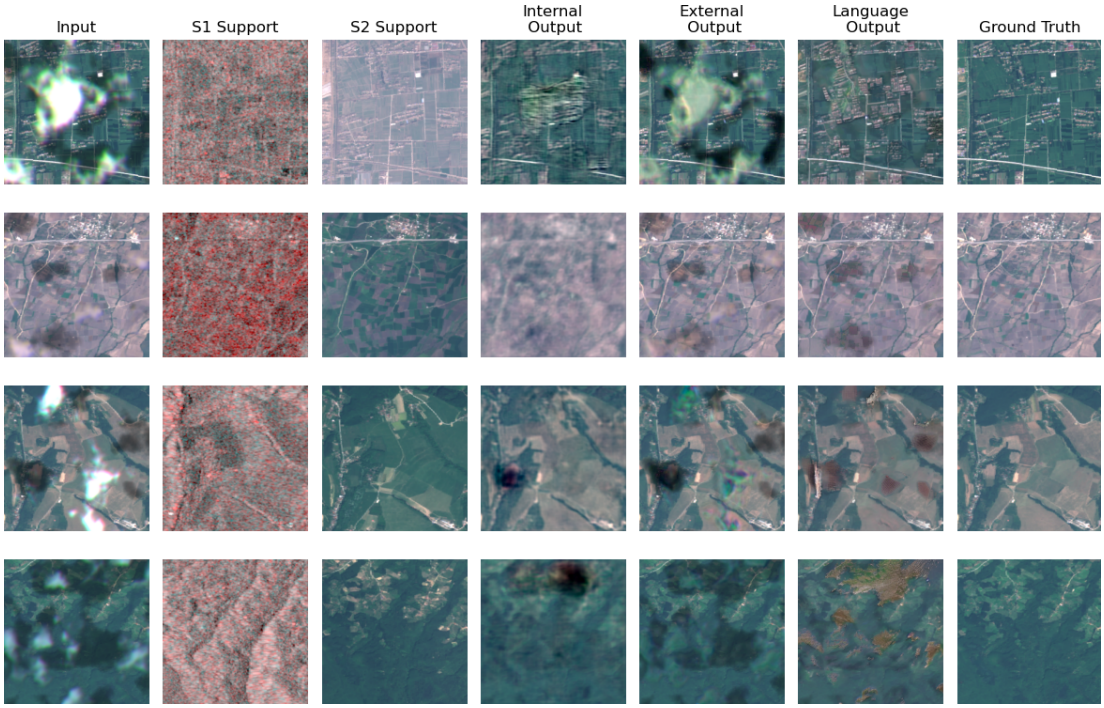


Figure 6.14: Examples of output computed for simulated cloud input.

performance. In the case of real data, the ground truth cloud-free image is a capture from several days before or after the cloudy sample, which can lead to significant changes having occurred on the ground between the two acquisitions. This means that the quality of ground truth for real images could be non-ideal and hence, a data-induced error could influence the measured model performance. On the other hand, the simulated data source guarantees that the ground truth will be accurate, however, the appearance of the clouds is only an approximation of what real clouds look like and in this case, the input cloudy image could contain some inaccuracies.

For that reason, it may be crucial to always perform the evaluation with both real and simulated data and consider the limitations outlined above. An ideal cloud removal model should be able to remove all clouds perfectly, and in that case, it should achieve zero error on the simulated images. However, for real data, given the drift of the ground truth image, the same method will yield a non-zero error whenever there is any inaccuracy in the ground truth.

Arriving at a solution that represents this ideal model with perfect performance

is a considerable challenge, as demonstrated by this work. The results indicate that pre-training on large-scale datasets leads to the highest gains in performance. However, the internal or language-based methods exhibit non-trivial capabilities while offering a high degree of flexibility. These observations motivate further progress in this direction, where the advantages of high performance and flexibility are combined into improved techniques for cloud removal. Based on the results presented in this work, this is a likely venue for getting close to the ideal cloud removal system.

Chapter 7

Conclusions

Cloud removal is a challenging problem that remains an open area of research. The problem of clouds obscuring the optical images of the Earth is relevant in many contexts and hence, high-quality solutions to the problem will be highly beneficial in a wide range of applications.

This work has explored the technical challenges behind cloud removal and proposed several unconventional techniques applicable to this task. Another important challenge of data availability was addressed by proposing a cloud simulator tool and demonstrating its applicability for the training and evaluation of cloud-removing deep neural networks. Finally, a comparison chapter introduced a shared dataset for evaluation and compared different types of solutions. The externally trained network has proven to perform best on all types of data, however, the other approaches achieved non-trivial performance while offering more flexibility than the externally trained model.

7.1 Contributions

The relevance of the cloud removal problem and the motivation for better solutions to this technical challenge was outlined in the introductory Chapter 1, which was followed by an overview of the existing remote sensing solutions, the phenomenon of clouds, and the literature on cloud detection, removal and image synthesis, contained in Chapter 2. The chapter also provided a formulation of the cloud removal problem as well as the

metrics used to evaluate the agreement with a ground truth reference.

In Chapter 3, the techniques of internal learning for cloud removal were proposed. These techniques were of interest because they can be flexible in regards to the input representation (the shape of processed data and the domain they come from). Several solutions inspired by the work on Deep Image Prior were proposed, all relying on the parameterisation of the images using convolutional neural network architecture, hence name Multi-modal Convolutional Parameterisation Networks (MCPN). It was shown that these networks can achieve good performance for the inpainting and super-resolution tasks, and freely accept guidance data, such as SAR images or historical optical references.

In Chapter 4, another novel path was explored, where the use of models pre-trained on the combination of text and visual data was explored. First, it was demonstrated that such models can capture useful representations for processing clouds in satellite imagery, despite never having been trained on the task specifically. It was shown that a CLIP pre-trained model can be effectively used to detect the presence of clouds in satellite imagery and can identify cloud-affected images from multiple data sources (Sentinel-2 and Landsat) as well as with various channel mappings (RGB and false-colour mappings from different bands). This analysis was followed by another proposal of applying a pre-trained StableDiffusion text-to-image model with ControlNet guidance to inpaint parts of the satellite images. These models are constrained to RGB channels, so a technique of transferring the information synthesised in RGB to other multi-spectral channels was proposed, based on Deep Image Prior. However, the general text-to-image models from the StableDiffusion family have been found prone to hallucinating a lot of undesired features in the inpainted regions, making them less suitable for inpainting, especially for large masks.

A reliable evaluation of cloud removal solutions requires reliable data. In Chapter 5, a novel technique for simulating the presence of clouds in multi-spectral imagery was proposed. This was a result of combining multiple methods for generating realistic clouds, such as cloud colour, channel misalignment, or channel-specific magnitude. It was shown how the controllable parameters of the simulation tool could be used to

achieve different types of clouds. To demonstrate the utility of the simulated data, it was shown that deep neural networks can be trained to perform two relevant tasks of cloud-shadow detection and cloud removal. The results indicate that the channel-specific magnitude is a particularly important feature of the simulated data and that networks trained purely on synthetic clouds can perform well on real ones. Finally, the data drift in real pairs of cloudy and cloud-free data means that the ground truth could be an inaccurate approximation of the correct output. In this case, simulated data does not suffer from this drift and it was shown how it can be used to evaluate models and get more insight into the performance for different types of clouds.

In Chapter 6, a shared test dataset for evaluating different types of solutions was proposed. The dataset contains pairs of cloud-free and cloudy samples (both real and simulated), a historical optical sample (usually from months before), and matched temporally proximate SAR images for each. It was described how the dataset was filtered to only contain high-quality data. Consequently, a deep neural network architecture capable of using both historical and radar guidance data for the cloud removal process was proposed and trained for comparison. The comparison was made to the two types of solutions explored in Chapter 3 (internal learning) and Chapter 4 (language). The results show that the externally trained model was capable of achieving higher performance and that the internal learning and the language-based techniques suffered from poor convergence, or undesired artefacts. That said, it is recognised that, unlike the external model, the alternative proposed approaches exhibit greater freedom and transferability across different domains and representations.

This work makes contributions in several key aspects related to cloud removal technologies, which can be summarised as:

- Proposes an internal satellite image inpainting technique termed MCPN with multi-source and multi-temporal guidance
- Demonstrates use and evaluation of CLIP, a general vision-language model, for the task of cloud presence detection
- Proposes an edge-guided image inpainting using StableDiffusion and evaluates

text-to-image generators for the task of satellite image inpainting

- Proposes an RGB-to-MSI channel completion technique based on the Deep Image Prior
- Proposes and evaluates a framework designed for the simulation of clouds and shadows in multi-spectral satellite images
- Demonstrates that models trained exclusively on simulated data are capable of processing real cloudy images
- Proposes a novel external architecture capable of fusing multi-source and multi-temporal guidance for cloud removal
- Designs a common evaluation dataset consisting of both real and simulated clouds as well as multi-source and multi-temporal guiding signals
- Evaluates and contrasts the proposed solutions on a common test dataset

7.2 Future Work

Several solutions were proposed as a potential replacement to the established approach of acquiring a (hopefully large) set of images and training a deep neural network to minimise the error in the output. This was motivated by certain problems associated with this approach, such as the risk of overfitting and limited adaptation to different sensing modalities.

However, the performance of these alternative methods was generally lower than the externally trained equivalent. In some sense, it is understandable given the fact that the externally trained model has the advantageous opportunity to learn from a wide set of examples precisely relevant to the performed task. Yet, it was demonstrated that a considerable amount of useful information for the removal task can be found in the input sample (in the cloud-free regions, the radar data, or the historical reference) or that a lot of powerful priors can be extracted from other tasks, such as vision-language joint learning.

A natural direction would be to try to bridge this gap and either redesign the external learning architectures to offer more flexibility (some effort in this direction was already done in this work, where any number of radar guidance images and optical guidance images can potentially be used) or inject more information into the internal learning methods while stabilising the internal optimisation process.

Finally, the cloud simulation framework proposed herein is an effective tool for the training and evaluation of models designed for cloud detection or cloud removal. However, further research could focus on potential extensions to further increase the realism of the generated clouds. This could follow a physics-informed approach with additional expert knowledge manually incorporated into the process, or a data-based approach where some of the advancements in generative modelling could be employed.

The findings of this work could provide foundations for such efforts and eventually lead to techniques that can adapt to different types of new data and make robust predictions about the view underneath the clouds in any image of the Earth, regardless of the sensing type, channels, season, or location.

Bibliography

[1] R. A. Schowengerdt, *Remote Sensing*, 2006.

[2] R. W. Orloff, *Apollo : the definitive sourcebook*, 1st ed., ser. Springer-Praxis books in space exploration. Berlin: Springer, 2006.

[3] J. J. McCarthy, "Reflections on: Our planet and its life, origins, and futures," *Science*, vol. 326, no. 5960, pp. 1646–1655, 2009.

[4] NASA Image and Video Library, "ARC-1972-AC78-1116," 1972, [Online; accessed April 29, 2023]. [Online]. Available: https://images.nasa.gov/details-ARC-1972-AC78-1116

[5] W. Emery and A. Camps, *The History of Satellite Remote Sensing*, 2017.

[6] H. P. Noordung, *The Problem of Space Travel: The Rocket Motor*, 2010.

[7] Y. Zhao, T. Celik, N. Liu, and H. C. Li, "A Comparative Analysis of GAN-Based Methods for SAR-to-Optical Image Translation," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, 2022.

[8] Committee on Earth Observation Satellites, "The CEOS Database," 2023, [Online; accessed April 29, 2023]. [Online]. Available: http://database.eohandbook.com

[9] A. Nugent, D. DeCou, and S. Russell, "Atmospheric Science: ATMO 200 Companion Text," *Atmospheric Science: ATMO 200 Companion Text*, pp. 167–218, 2019.

[10] G. L. Stephens, M. Christensen, T. Andrews, J. Haywood, F. F. Malavelle, K. Suzuki, X. Jing, M. Lebsock, J. L. F. Li, H. Takahashi, and O. Sy, "Cloud physics from space," *Quarterly Journal of the Royal Meteorological Society*, vol. 145, no. 724, pp. 2854–2875, 2019.

[11] C. Aybar, L. Ysuhuaylas, K. Gonzales, J. Loja, F. Herrera, L. Bautista, A. Flores, R. Yali, L. Diaz, N. Cuenca, F. Prudencio, D. Montero, M. Sudmanns, D. Tiede, G. Mateo-garc, and G. Luis, "CloudSEN12 - a global dataset for semantic understanding of cloud and cloud shadow in satellite imagery," *EarthArXiv*, 2022.

[12] R. W. Saunders and K. T. Kriebel, "An improved method for detecting clear sky and cloudy radiances from AVHRR data," *International Journal of Remote Sensing*, vol. 9, no. 1, pp. 123–150, 1988.

[13] S. A. Ackerman, K. I. Strabala, W. P. Menzel, R. A. Frey, C. C. Moeller, and L. E. Gumley, "Discriminating clear sky from clouds with MODIS," *Journal of Geophysical Research Atmospheres*, vol. 103, no. D24, pp. 32 141–32 157, 1998.

[14] F. M. Bréon and S. Colzy, "Cloud detection from the spaceborne POLDER instrument and validation against surface synoptic observations," *Journal of Applied Meteorology*, vol. 38, no. 6, pp. 777–785, 1999.

[15] G. Lissens, P. Kempeneers, F. Fierens, and J. Van Rensbergen, "Development of cloud, snow, and shadow masking algorithms for VEGETATION imagery," *International Geoscience and Remote Sensing Symposium (IGARSS)*, vol. 2, pp. 834–836, 2000.

[16] O. Hagolle, M. Huc, D. V. Pascual, and G. Dedieu, "A multi-temporal method for cloud detection, applied to FORMOSAT-2, VENμS, LANDSAT and SENTINEL-2 images," *Remote Sensing of Environment*, vol. 114, no. 8, pp. 1747–1755, 2010. [Online]. Available: http://dx.doi.org/10.1016/j.rse.2010.03.002

[17] P. L. Scaramuzza, M. A. Bouchard, and J. L. Dwyer, "Development of the landsat data continuity mission cloud-cover assessment algorithms," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 4, pp. 1140–1154, 2012.

[18] X. Zhu, F. Gao, D. Liu, and J. Chen, "A modified neighborhood similar pixel interpolator approach for removing thick clouds in landsat images," *IEEE Geoscience and Remote Sensing Letters*, vol. 9, no. 3, pp. 521–525, 2012.

[19] J. Louis, V. Debaecker, B. Pflug, M. Main-Knorn, J. Bieniarz, U. Mueller-Wilm, E. Cadau, and F. Gascon, "Sentinel-2 SEN2COR: L2A processor for users," *European Space Agency, (Special Publication) ESA SP*, vol. SP-740, no. August, pp. 9–13, 2016.

[20] D. Frantz, E. Haß, A. Uhl, J. Stoffels, and J. Hill, "Improvement of the Fmask algorithm for Sentinel-2 images: Separating clouds from bright surfaces based on parallax effects," *Remote Sensing of Environment*, vol. 215, no. April 2017, pp. 471–481, 2018.

[21] Sentinel Hub Team, "Sentinel hub's cloud detector for sentinel-2 imagery," 2017, [Online; accessed April 29, 2023]. [Online]. Available: https://github.com/sentinel-hub/sentinel2-cloud-detector

[22] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998.

[23] T. Johnston, S. R. Young, D. Hughes, R. M. Patton, and D. White, "Optimizing Convolutional Neural Networks for Cloud Detection," 2017.

[24] M. Shi, F. Xie, Y. Zi, and J. Yin, "Cloud detection of remote sensing images by deep learning," *International Geoscience and Remote Sensing Symposium (IGARSS)*, vol. 2016-November, pp. 701–704, 2016.

[25] G. Mateo-Garcia, L. Gomez-Chova, and G. Camps-Valls, "Convolutional neural networks for multispectral image cloud masking," in *International Geoscience and Remote Sensing Symposium (IGARSS)*, vol. 2017-July, 2017, pp. 2255–2258.

[26] S. Ozkan, M. Efendioglu, and C. Demirpolat, "Cloud detection from RGB color remote sensing images with deep pyramid networks," *International Geoscience and Remote Sensing Symposium (IGARSS)*, vol. 2018-July, pp. 6939–6942, 2018.

[27] Y. Zi, F. Xie, and Z. Jiang, "A cloud detection method for Landsat 8 images based on PCANet," *Remote Sensing*, vol. 10, no. 6, pp. 1–21, 2018.

[28] Z. Li, H. Shen, Q. Cheng, Y. Liu, S. You, and Z. He, "Deep learning based cloud detection for medium and high resolution remote sensing images of different sensors," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 150, no. February, pp. 197–212, 2019. [Online]. Available: https://doi.org/10.1016/j.isprsjprs.2019.02.017

[29] J. H. Jeppesen, R. H. Jacobsen, F. Inceoglu, and T. S. Toftegaard, "A cloud detection algorithm for satellite imagery based on deep learning," *Remote Sensing of Environment*, vol. 229, no. May, pp. 247–259, 2019. [Online]. Available: https://doi.org/10.1016/j.rse.2019.03.039

[30] M. Domnich, I. Sünter, H. Trofimov, O. Wold, F. Harun, A. Kostiukhin, M. Järveoja, M. Veske, T. Tamm, K. Voormansik, A. Olesk, V. Boccia, N. Longepe, and E. G. Cadau, "KappaMask: Ai-based cloudmask processor for sentinel-2," *Remote Sensing*, vol. 13, no. 20, 2021.

[31] G. Mateo-García, V. Laparra, D. López-Puigdollers, and L. Gómez-Chova, "Transferring deep learning models for cloud detection between Landsat-8 and Proba-V," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 160, no. November 2019, pp. 1–17, 2020. [Online]. Available: https://doi.org/10.1016/j.isprsjprs.2019.11.024

[32] G. Mateo-Garcia, V. Laparra, D. Lopez-Puigdollers, and L. Gomez-Chova, "Cross-Sensor Adversarial Domain Adaptation of Landsat-8 and Proba-V Images for Cloud Detection," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 747–761, 2021.

[33] L. Baetens, C. Desjardins, and O. Hagolle, "Validation of copernicus Sentinel-2 cloud masks obtained from MAJA, Sen2Cor, and FMask processors using reference cloud masks generated with a supervised active learning procedure," *Remote Sensing*, vol. 11, no. 4, pp. 1–25, 2019.

[34] V. Zekoll, M. Main-Knorn, J. Louis, D. Frantz, R. Richter, and B. Pflug, "Comparison of masking algorithms for sentinel-2 imagery," *Remote Sensing*, vol. 13, no. 1, pp. 1–21, 2021.

[35] J. Li, Z. Wu, Z. Hu, Z. Li, Y. Wang, and M. Molinier, "Deep learning based thin cloud removal fusing vegetation red edge and short wave infrared spectral information for sentinel-2A imagery," *Remote Sensing*, vol. 13, no. 1, pp. 1–31, 2021.

[36] D. López-Puigdollers, G. Mateo-García, and L. Gómez-Chova, "Benchmarking deep learning models for cloud detection in landsat-8 and sentinel-2 images," *Remote Sensing*, vol. 13, no. 5, pp. 1–20, 2021.

[37] S. Skakun, J. Wevers, C. Brockmann, G. Doxani, M. Aleksandrov, M. Batič, D. Frantz, F. Gascon, L. Gómez-Chova, O. Hagolle, D. López-Puigdollers, J. Louis, M. Lubej, G. Mateo-García, J. Osman, D. Peressutti, B. Pflug, J. Puc, R. Richter, J. C. Roger, P. Scaramuzza, E. Vermote, N. Vesel, A. Zupanc, and L. Žust, "Cloud Mask Intercomparison eXercise (CMIX): An evaluation of cloud masking algorithms for Landsat 8 and Sentinel-2," *Remote Sensing of Environment*, vol. 274, no. September 2021, 2022.

[38] O. R. Mitchell, E. J. Delp, and P. L. Chen, "Filtering To Remove Cloud Cover in Satellite Imagery." *IEEE Trans Geosci Electron*, vol. GE-15, no. 3, pp. 137–141, 1977.

[39] J. Cihlar and J. Howarth, "Detection and Removal of Cloud Contamination from AVHRR Images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 32, no. 3, pp. 583–589, 1994.

[40] B. Wang, A. Ono, K. Muramatsu, and N. Fujiwarattt, "Automated detection and removal of clouds and their shadows from landsat TM images," *IEICE Transactions on Information and Systems*, vol. E82-D, no. 2, pp. 453–460, 1999.

[41] M. Li, S. C. Liew, and L. K. Kwoh, "Generating "cloud free" and "cloud-shadow free" mosaic for spot panchromatic images," *International Geoscience and Remote Sensing Symposium (IGARSS)*, vol. 4, no. C, pp. 2480–2482, 2002.

[42] Z. Wang, J. Jin, J. Liang, K. Yan, and Q. Peng, "A new cloud removal algorithm for multi-spectral images," *MIPPR 2005: SAR and Multispectral Image Processing*, vol. 6043, p. 60430W, 2005.

[43] E. H. Helmer and B. Ruefenacht, "Cloud-free satellite image mosaics with regression trees and histogram matching," *Photogrammetric Engineering and Remote Sensing*, vol. 71, no. 9, pp. 1079–1089, 2005.

[44] F. Melgani, "Contextual reconstruction of cloud-contaminated multitemporal multispectral images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 2, pp. 442–455, 2006.

[45] S. Gabarda and G. Cristóbal, "Cloud covering denoising through image fusion," *Image and Vision Computing*, vol. 25, no. 5, pp. 523–530, 2007.

[46] A. Maalouf, P. Carré, B. Augereau, and C. Fernandez-Maloigne, "A bandelet-based inpainting technique for clouds removal from remotely sensed images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 7, pp. 2363–2371, 2009.

[47] L. Poggio, A. Gimona, and I. Brown, "Spatio-temporal MODIS EVI gap filling under cloud cover: An example in Scotland," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 72, pp. 56–72, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.isprsjprs.2012.06.003

[48] J. Chen, X. Zhu, J. E. Vogelmann, F. Gao, and S. Jin, "A simple and effective method for filling gaps in Landsat ETM+ SLC-off images," *Remote Sensing*

of Environment, vol. 115, no. 4, pp. 1053–1064, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.rse.2010.12.010

[49] C. Zeng, H. Shen, and L. Zhang, "Recovering missing pixels for Landsat ETM+ SLC-off imagery using multi-temporal regression analysis and a regularization method," *Remote Sensing of Environment*, vol. 131, pp. 182–194, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.rse.2012.12.012

[50] D. C. Tseng and C. L. Chien, "A cloud removal approach for aerial image visualization," *International Journal of Innovative Computing, Information and Control*, vol. 9, no. 6, pp. 2421–2440, 2013.

[51] C. H. Lin, P. H. Tsai, K. H. Lai, and J. Y. Chen, "Cloud removal from multitemporal satellite images using information cloning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 1, pp. 232–241, 2013.

[52] X. Li, H. Shen, S. Member, L. Zhang, and S. Member, "Recovering Quantitative Remote Sensing Products Contaminated by Thick Clouds and Shadows Using Multitemporal Dictionary Learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 11, pp. 7086–7098, 2014.

[53] H. Shen, H. Li, Y. Qian, L. Zhang, and Q. Yuan, "An effective thin cloud removal procedure for visible remote sensing images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 96, pp. 224–235, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.isprsjprs.2014.06.011

[54] B. Huang, Y. Li, X. Han, Y. Cui, W. Li, and R. Li, "Cloud removal from optical satellite imagery with SAR imagery using sparse representation," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 5, pp. 1046–1050, 2015.

[55] M. Xu, M. Pickering, A. J. Plaza, and X. Jia, "Thin cloud removal based on signal transmission principles and spectral mixture analysis," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 3, pp. 1659–1669, 2016.

[56] T. Sandhan and J. Y. Choi, "Simultaneous Detection and Removal of High Altitude Clouds from an Image," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017-Octob, pp. 4789–4798, 2017.

[57] J. Wang, P. A. Olsen, A. R. Conn, and A. C. Lozano, "Removing Clouds and Recovering Ground Observations in Satellite Image Sequences via Temporally Contiguous Robust Matrix Completion," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 2754–2763, 2016.

[58] K. Enomoto, K. Sakurada, W. Wang, H. Fukui, M. Matsuoka, R. Nakamura, and N. Kawaguchi, "Filmy Cloud Removal on Satellite Imagery with Multispectral Conditional Generative Adversarial Nets," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 2017-July, pp. 1533–1541, 2017.

[59] P. Singh and N. Komodakis, "Cloud-GAN: Cloud removal for sentinel-2 imagery using a cyclic consistent generative adversarial networks," *International Geoscience and Remote Sensing Symposium (IGARSS)*, vol. 2018-July, pp. 1772–1775, 2018.

[60] C. Grohnfeldt, M. Schmitt, and X. Zhu, "A conditional generative adversarial network to fuse SAR and multispectral optical data for cloud removal from Sentinel-2 images," *International Geoscience and Remote Sensing Symposium (IGARSS)*, vol. 2018-July, pp. 1726–1729, 2018.

[61] J. Gao, Q. Yuan, J. Li, H. Zhang, and X. Su, "Cloud removal with fusion of high resolution optical and SAR images using generative adversarial networks," *Remote Sensing*, vol. 12, no. 1, 2020.

[62] J. D. Bermudez, P. N. Happ, D. A. Oliveira, and R. Q. Feitosa, "SAR to Optical Image Synthesis for Cloud Removal with Generative Adversarial Networks," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 4, no. 1, pp. 5–11, 2018.

[63] Z. Xu, K. Wu, L. Huang, Q. Wang, and P. Ren, "Cloudy Image Arithmetic: A Cloudy Scene Synthesis Paradigm With an Application to Deep-Learning-Based Thin Cloud Removal," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, 2022.

[64] V. Sarukkai, A. Jain, B. Uzkent, and S. Ermon, "Cloud removal in satellite images using spatiotemporal generative networks," *Proceedings - 2020 IEEE Winter Conference on Applications of Computer Vision, WACV 2020*, pp. 1785–1794, 2020.

[65] J. Li, Z. Wu, Z. Hu, J. Zhang, M. Li, L. Mo, and M. Molinier, "Thin cloud removal in optical remote sensing images based on generative adversarial networks and physical model of cloud distortion," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 166, no. March, pp. 373–389, 2020.

[66] H. Pan, "Cloud Removal for Remote Sensing Imagery via Spatial Attention Generative Adversarial Network," 2020. [Online]. Available: http://arxiv.org/abs/2009.13015

[67] P. Ebel, A. Meraner, M. Schmitt, and X. X. Zhu, "Multisensor Data Fusion for Cloud Removal in Global and All-Season Sentinel-2 Imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 7, pp. 5866–5878, 2021.

[68] M. Xu, F. Deng, S. Jia, X. Jia, and A. J. Plaza, "Attention mechanism-based generative adversarial networks for cloud removal in Landsat images," *Remote Sensing of Environment*, vol. 271, no. August 2021, p. 112902, 2022. [Online]. Available: https://doi.org/10.1016/j.rse.2022.112902

[69] F. Xu, Y. Shi, P. Ebel, L. Yu, G. S. Xia, W. Yang, and X. X. Zhu, "GLF-CR: SAR-enhanced cloud removal with global–local fusion," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 192, no. July, pp. 268–278, 2022. [Online]. Available: https://doi.org/10.1016/j.isprsjprs.2022.08.002

[70] Z. Xu, K. Wu, W. Wang, X. Lyu, and P. Ren, "Semi-supervised thin cloud removal with mutually beneficial guides," *ISPRS Journal of Photogrammetry*

and Remote Sensing, vol. 192, no. August, pp. 327–343, 2022. [Online]. Available: https://doi.org/10.1016/j.isprsjprs.2022.08.026

[71] A. Sebastianelli, E. Puglisi, M. P. D. Rosso, J. Mifdal, A. Nowakowski, P. P. Mathieu, F. Pirri, and S. L. Ullo, "PLFM: Pixel-Level Merging of Intermediate Feature Maps by disentangling and fusing spatial and temporal data for Cloud Removal," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–1, 2022.

[72] P. Ebel, M. Schmitt, and X. X. Zhu, "Internal Learning for Sequence-to-Sequence Cloud Removal via Synthetic Aperture Radar Prior Information," *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, pp. 2691–2694, 2021.

[73] M. Qin, F. Xie, W. Li, Z. Shi, and H. Zhang, "Dehazing for Multispectral Remote Sensing Images Based on a Convolutional Neural Network with the Residual Architecture," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 5, pp. 1645–1655, 2018.

[74] W. Li, Y. Li, D. Chen, and J. C.-w. Chan, "Thin Cloud Removal with Residual Symmetrical Concatenation Network," in *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2019, pp. 1974–1977.

[75] K. Y. Lee and J. Y. Sim, "Cloud Removal of Satellite Images Using Convolutional Neural Network with Reliable Cloudy Image Synthesis Model," *Proceedings - International Conference on Image Processing, ICIP*, vol. 2019-Septe, pp. 3581–3585, 2019.

[76] Y. Chen, L. Tang, X. Yang, R. Fan, M. Bilal, and Q. Li, "Thick Clouds Removal from Multitemporal ZY-3 Satellite Images Using Deep Learning," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 143–153, 2020.

[77] A. Meraner, P. Ebel, X. X. Zhu, and M. Schmitt, "Cloud removal in Sentinel-2 imagery using a deep residual neural network and SAR-

optical data fusion," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 166, no. January, pp. 333–346, 2020. [Online]. Available: https://doi.org/10.1016/j.isprsjprs.2020.05.013

[78] Y. Zi, F. Xie, N. Zhang, Z. Jiang, W. Zhu, and H. Zhang, "Thin Cloud Removal for Multispectral Remote Sensing Images Using Convolutional Neural Networks Combined with an Imaging Model," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 3811–3823, 2021.

[79] Q. Zhang, Q. Yuan, Z. Li, F. Sun, and L. Zhang, "Combined deep prior with low-rank tensor SVD for thick cloud removal in multitemporal images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 177, no. April, pp. 161–173, 2021. [Online]. Available: https://doi.org/10.1016/j.isprsjprs.2021.04.021

[80] P. Ebel, M. Schmitt, and X. X. Zhu, "Internal learning for sequence-to-sequence cloud removal via synthetic aperture radar prior information," in *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, 2021, pp. 2691–2694.

[81] S. Han, J. Wang, and S. Zhang, "Former-CR : A Transformer-Based Thick Cloud Removal Method with Optical and SAR Imagery," 2023.

[82] S. Chen, X. Chen, X. Chen, J. Chen, X. Cao, M. Shen, W. Yang, and X. Cui, "A novel cloud removal method based on IHOT and the cloud trajectories for landsat imagery," *Remote Sensing*, vol. 10, no. 7, pp. 1–17, 2018.

[83] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. Burges, L. Bottou, and K. Q. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[84] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2, no. September, pp. 1033–1038, 1999.

[85] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2001*, pp. 341–346, 2001.

[86] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, pp. 417–424, 2000.

[87] I. Drori, D. Cohen-Or, and H. Yeshurun, "Fragment-based image completion," *ACM SIGGRAPH 2003 Papers, SIGGRAPH '03*, pp. 303–312, 2003.

[88] J. Sun, L. Yuan, J. Jia, and H. Y. Shum, "Image completion with structure propagation," *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 861–868, 2005.

[89] N. Diakopoulos, I. Essa, R. Jain, G. Goos, and J. Hartmanis, "Content Based Image Synthesis," in *Image and Video Retrieval*, P. Enser, Y. Kompatsiaris, N. E. O'Connor, A. F. Smeaton, and A. W. M. Smeulders, Eds., vol. 106. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 299–307.

[90] M. Johnson, G. J. Brostow, J. Shotton, O. Arandjelovic, V. Kwatra, and R. Cipolla, "Semantic photo synthesis," *Computer Graphics Forum*, vol. 25, no. 3, pp. 407–413, 2006.

[91] J. Hays and A. A. Efros, "Scene completion using millions of photographs," *ACM Transactions on Graphics*, vol. 26, no. 3, pp. 1–7, 2007.

[92] A. Oliva and A. Torralba, "Building the gist of a scene: the role of global image features in recognition," in *Progress in Brain Research*, ser. Progress in Brain Research, S. Martinez-Conde, S. L. Macknik, L. M. Martinez, J.-M. Alonso, and P. U. Tse, Eds. Elsevier, 2006, vol. 155 B, pp. 23–36. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0079612306550022

[93] G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[94] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 10 684–10 695.

[95] Jia Deng, Wei Dong, R. Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," 2009, pp. 248–255.

[96] Y. Bengio, G. Mesnil, Y. Dauphin, and S. Rifai, "Better mixing via deep representations," in *Proceedings of the 30th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, S. Dasgupta and D. McAllester, Eds., vol. 28, no. 1. Atlanta, Georgia, USA: PMLR, 17–19 Jun 2013, pp. 552–560. [Online]. Available: https://proceedings.mlr.press/v28/bengio13.html

[97] Y. Bengio, É. Thibodeau-Laufer, G. Alain, and J. Yosinski, "Deep generative stochastic networks trainable by backprop," *31st International Conference on Machine Learning, ICML 2014*, vol. 2, pp. 1470–1485, 2014.

[98] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, no. Ml, pp. 1–14, 2014.

[99] 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. 3, no. January, pp. 2672–2680, 2014.

[100] A. Krizhevsky, "Learning multiple layers of features from tiny images," 2009.

[101] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *32nd International Conference on Machine Learning, ICML 2015*, vol. 3, 2015, pp. 2246–2255.

[102] J. Song, C. Meng, and S. Ermon, "Denoising Diffusion Implicit Models," pp. 1–22, 2020. [Online]. Available: http://arxiv.org/abs/2010.02502

[103] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, vol. 2020-December, no. NeurIPS 2020, 2020, pp. 1–25.

[104] A. Nichol and P. Dhariwal, "Improved Denoising Diffusion Probabilistic Models," 2021.

[105] P. Dhariwal and A. Nichol, "Diffusion Models Beat GANs on Image Synthesis," *Advances in Neural Information Processing Systems*, vol. 11, pp. 8780–8794, 2021.

[106] E. L. Denton, S. Chintala, arthur Szlam, and R. Fergus, "Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015. [Online]. Available: https://proceedings.neurips.cc/paper{_}files/paper/2015/file/aa169b49b583a2b5af89203c2b78c67c-Paper.pdf

[107] A. Dosovitskiy, J. T. Springenberg, and T. Brox, "Learning to generate chairs with convolutional neural networks," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June-2015, pp. 1538–1546, 2015.

[108] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, pp. 1–16, 2016.

[109] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 5967–5976, 2017.

[110] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," *Proceedings of the IEEE*

International Conference on Computer Vision, vol. 2017-Octob, pp. 2242–2251, 2017.

[111] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 4396–4405, 2019.

[112] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, no. NeurIPS, pp. 8107–8116, 2020. [Online]. Available: https://blog.faradars.org/generative-adversarial-networks/

[113] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila, "Alias-Free Generative Adversarial Networks," *Advances in Neural Information Processing Systems*, vol. 2, no. NeurIPS, pp. 852–863, 2021. [Online]. Available: http://arxiv.org/abs/2106.12423

[114] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 12 868–12 878, 2021.

[115] A. Nguyen, J. Clune, Y. Bengio, A. Dosovitskiy, and J. Yosinski, "Plug & play generative networks: Conditional iterative generation of images in latent space," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2017.

[116] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 8821–8831. [Online]. Available: https://proceedings.mlr.press/v139/ramesh21a.html

[117] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper{_}files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[118] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical Text-Conditional Image Generation with CLIP Latents," no. Figure 3, 2022. [Online]. Available: http://arxiv.org/abs/2204.06125

[119] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S. K. S. Ghasemipour, R. Gontijo-Lopes, B. K. Ayan, T. Salimans, J. Ho, D. J. Fleet, and M. Norouzi, "Photorealistic text-to-image diffusion models with deep language understanding," in *Advances in Neural Information Processing Systems*, A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, Eds., 2022. [Online]. Available: https://openreview.net/forum?id=08Yk-n5l2Al

[120] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context Encoders: Feature Learning by Inpainting," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December, 2016, pp. 2536–2544. [Online]. Available: https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Pathak_Context_Encoders_Feature_CVPR_2016_paper.pdf%0Apapers3://publication/uuid/9E05080B-9457-4DFE-B5DA-C42DC2CFEE40

[121] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Transactions on Graphics*, vol. 36, no. 4, 2017.

[122] G. Liu, F. A. Reda, K. J. Shih, T. C. Wang, A. Tao, and B. Catanzaro, "Image Inpainting for Irregular Holes Using Partial Convolutions," in *The European Conference on Computer Vision (ECCV)*, 2018.

[123] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative Image Inpainting with Contextual Attention," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 5505–5514, 2018.

[124] Z. Yi, Q. Tang, S. Azizi, D. Jang, and Z. Xu, "Contextual Residual Aggregation for Ultra High-Resolution Image Inpainting," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 7505–7514, 2020.

[125] S. Zhao, J. Cui, Y. Sheng, Y. Dong, X. Liang, E. Chang, and Y. Xu, "Large scale image compeletion via comodulated GAN," in *International Conference on Learning Representations*, 2021.

[126] R. Suvorov, E. Logacheva, A. Mashikhin, A. Remizova, A. Ashukha, A. Silvestrov, N. Kong, H. Goka, K. Park, and V. Lempitsky, "Resolution-robust Large Mask Inpainting with Fourier Convolutions," *Proceedings - 2022 IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2022*, pp. 3172–3182, 2022. [Online]. Available: http://arxiv.org/abs/2109.07161

[127] Y. Ma, X. Liu, S. Bai, L. Wang, A. Liu, D. Tao, and E. R. Hancock, "Region-wise Generative Adversarial Image Inpainting for Large Missing Areas," *IEEE Transactions on Cybernetics*, vol. 1, no. c, pp. 1–14, 2022.

[128] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. Van Gool, "RePaint: Inpainting using Denoising Diffusion Probabilistic Models," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2022-June, pp. 11 451–11 461, 2022. [Online]. Available: http://arxiv.org/abs/2201.09865

[129] C. Saharia, W. Chan, H. Chang, C. Lee, J. Ho, T. Salimans, D. Fleet, and M. Norouzi, "Palette: Image-to-Image Diffusion Models," *Proceedings of ACM SIGGRAPH*, vol. 1, no. 1, pp. 1–10, 2022. [Online]. Available: http://arxiv.org/abs/2111.05826

[130] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-January, 2017, pp. 105–114. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2017/papers/Ledig_Photo-Realistic_Single_Image_CVPR_2017_paper.pdf

[131] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced Deep Residual Networks for Single Image Super-Resolution," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 2017-July, 2017, pp. 1132–1140.

[132] R. Dahl, M. Norouzi, and J. Shlens, "Pixel Recursive Super Resolution," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017-October, pp. 5449–5458, 2017.

[133] Y. Chen, Y. Tai, X. Liu, C. Shen, and J. Yang, "FSRNet: End-to-End Learning Face Super-Resolution with Facial Priors," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2492–2501, 2018.

[134] S. Menon, A. Damian, S. Hu, N. Ravi, and C. Rudin, "PULSE: Self-Supervised Photo Upsampling via Latent Space Exploration of Generative Models," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2434–2442, 2020.

[135] 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*, 2022. [Online]. Available: http://arxiv.org/abs/2104.07636

[136] A. Shocher, N. Cohen, and M. Irani, "Zero-Shot Super-Resolution Using Deep Internal Learning," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3118–3126, 2018.

[137] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 349–356, 2009.

[138] M. Zontak and M. Irani, "Internal statistics of a single natural image," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 977–984, 2011.

[139] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep Image Prior," *International Journal of Computer Vision*, vol. 128, no. 7, pp. 1867–1888, 2020.

[140] Y. Gandelsman, A. Shocher, and M. Irani, "'Double-dip': Unsupervised image decomposition via coupled deep-image-priors," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 11 018–11 027, 2019.

[141] H. Zhang, L. Mai, H. Jin, Z. Wang, N. Xu, and J. Collomosse, "An internal learning approach to video inpainting," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2019-Octob, pp. 2720–2729, 2019.

[142] L. P. Zuckerman, E. Naor, G. Pisha, S. Bagon, and M. Irani, "Across Scales and Across Dimensions: Temporal Super-Resolution Using Deep Internal Learning," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12352 LNCS, pp. 52–68, 2020.

[143] A. Shocher, S. Bagon, P. Isola, and M. Irani, "InGAN: Capturing and retargeting the 'DNA' of a natural image," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2019-Octob, no. i, pp. 4491–4500, 2019.

[144] S. Bell-Kligler, A. Shocher, and M. Irani, "Blind super-resolution kernel estimation using an internal-GAN," *Advances in Neural Information Processing Systems*, vol. 32, no. 788535, pp. 1–10, 2019. [Online]. Available: http://arxiv.org/abs/1909.06581

[145] T. R. Shaham, T. Dekel, and T. Michaeli, "SinGAN: Learning a generative model from a single natural image," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2019-Octob, pp. 4569–4579, 2019.

[146] J. Lin, Y. Pang, Y. Xia, Z. Chen, and J. Luo, "TuiGAN: Learning Versatile Image-to-Image Translation with Two Unpaired Images," in *ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., vol. 12349 LNCS. Cham: Springer International Publishing, 2020, pp. 18–35. [Online]. Available: 10.1007/978-3-030-58548-8_2

[147] I. D. Mastan and S. Raman, "DeepCFL: Deep contextual features learning from a single image," *Proceedings - 2021 IEEE Winter Conference on Applications of Computer Vision, WACV 2021*, pp. 2896–2905, 2021.

[148] I. D. Mastan, S. Raman, and P. Singh, "DILIE: Deep Internal Learning for Image Enhancement," *Proceedings - 2022 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops, WACVW 2022*, vol. 2, pp. 24–33, 2022.

[149] N. Granot, B. Feinstein, A. Shocher, S. Bagon, and M. Irani, "Drop the GAN: In Defense of Patches Nearest Neighbors as Single Image Generative Models," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2022-June, pp. 13 450–13 459, 2022. [Online]. Available: http://arxiv.org/abs/2103.15545

[150] L. Gatys, A. Ecker, and M. Bethge, "A Neural Algorithm of Artistic Style," *Journal of Vision*, vol. 16, no. 12, p. 326, 2016.

[151] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," *Lecture Notes in Computer Science*

(including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 9906 LNCS, pp. 694–711, mar 2016. [Online]. Available: http://arxiv.org/abs/1603.08155

[152] X. Huang and S. Belongie, "Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017-October, pp. 1510–1519, 2017.

[153] D. Y. Park and K. H. Lee, "Arbitrary style transfer with style-attentional networks," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 5873–5881, 2019.

[154] Y. Deng, F. Tang, W. Dong, W. Sun, F. Huang, and C. Xu, "Arbitrary Style Transfer via Multi-Adaptation Network," *MM 2020 - Proceedings of the 28th ACM International Conference on Multimedia*, pp. 2719–2727, 2020.

[155] Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu, and M. Song, "Neural Style Transfer: A Review," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 11, pp. 3365–3385, 2020.

[156] H. Chen, L. Zhao, Z. Wang, H. Zhang, Z. Zuo, A. Li, W. Xing, and D. Lu, "Artistic Style Transfer with Internal-external Learning and Contrastive Learning," *Advances in Neural Information Processing Systems*, vol. 32, no. NeurIPS, pp. 26 561–26 573, 2021.

[157] S. Liu, T. Lin, D. He, F. Li, M. Wang, X. Li, Z. Sun, Q. Li, and E. Ding, "AdaAttN: Revisit Attention Mechanism in Arbitrary Neural Style Transfer," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 6629–6638, 2021. [Online]. Available: http://arxiv.org/abs/2108.03647

[158] M. M. Cheng, X. C. Liu, J. Wang, S. P. Lu, Y. K. Lai, and P. L. Rosin, "Structure-Preserving Neural Style Transfer," *IEEE Transactions on Image Processing*, vol. 29, pp. 909–920, 2020.

[159] J. Huo, S. Jin, W. Li, J. Wu, Y. K. Lai, Y. Shi, and Y. Gao, "Manifold Alignment for Semantically Aligned Style Transfer," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 14 841–14 849, 2021. [Online]. Available: http://arxiv.org/abs/2005.10777

[160] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[161] Q. Zhang, Q. Yuan, C. Zeng, X. Li, and Y. Wei, "Missing data reconstruction in remote sensing image with a unified spatial-temporal-spectral deep convolutional neural network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 8, pp. 4274–4288, 2018.

[162] X. Luo, Z. Han, and L. Yang, "Progressive attentional manifold alignment for arbitrary style transfer," in *Proceedings of the Asian Conference on Computer Vision (ACCV)*, December 2022, pp. 3206–3222.

[163] G. Ongie, A. Jalal, C. A. Metzler, R. G. Baraniuk, A. G. Dimakis, and R. Willett, "Deep Learning Techniques for Inverse Problems in Imaging," *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 1, pp. 39–56, 2020.

[164] A. Horé and D. Ziou, "Image quality metrics: Psnr vs. ssim," in *2010 20th International Conference on Pattern Recognition*, 2010, pp. 2366–2369.

[165] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[166] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf

[167] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, no. 1, pp. 586–595, 2018.

[168] K. Ding, K. Ma, S. Wang, and E. P. Simoncelli, "Image quality assessment: Unifying structure and texture similarity," *CoRR*, vol. abs/2004.07728, 2020. [Online]. Available: https://arxiv.org/abs/2004.07728

[169] P. Ebel, Y. Xu, M. Schmitt, and X. X. Zhu, "SEN12MS-CR-TS: A Remote Sensing Data Set for Multi-modal Multi-temporal Cloud Removal," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–1, 2022.

[170] F. A. Kruse, A. B. Lefkoff, J. W. Boardman, K. B. Heidebrecht, A. T. Shapiro, P. J. Barloon, and A. F. Goetz, "The spectral image processing system (SIPS)-interactive visualization and analysis of imaging spectrometer data," *Remote Sensing of Environment*, vol. 44, no. 2-3, pp. 145–163, 1993.

[171] M. I. Assaf Shocher, Nadav Cohen, ""zero-shot" super-resolution using deep internal learning," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[172] M. Czerkawski, P. Upadhyay, C. Davison, A. Werkmeister, J. Cardona, R. Atkinson, C. Michie, I. Andonovic, M. Macdonald, and C. Tachtatzis, "Paired sentinel-1 and sentinel-2 images for 2 locations in scotland and india for 2019 and 2020," 2022. [Online]. Available: https://zenodo.org/record/5903334

[173] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.

[174] P. Upadhyay, M. Czerkawski, C. Davison, J. Cardona, M. Macdonald, I. Andonovic, C. Michie, R. Atkinson, N. Papadopoulou, K. Nikas, and C. Tachtatzis, "A flexible multi-temporal and multi-modal framework for sentinel-1 and sentinel-2 analysis ready data," *Remote Sensing*, vol. 14, no. 5, 2022. [Online]. Available: https://www.mdpi.com/2072-4292/14/5/1120

[175] K. He, J. Sun, and X. Tang, "Guided image filtering," in *Computer Vision – ECCV 2010*, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 1–14.

[176] J. T. Barron and B. Poole, "The fast bilateral solver," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 617–632.

[177] T.-W. Hui, C. C. Loy, and X. Tang, "Depth map super-resolution by deep multi-scale guidance," in *Proceedings of European Conference on Computer Vision (ECCV)*, 2016.

[178] B. Ham, M. Cho, and J. Ponce, "Robust guided image filtering using nonconvex potentials," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 1, pp. 192–207, 2018.

[179] R. D. Lutio, S. D'Aronco, J. D. Wegner, and K. Schindler, "Guided super-resolution as pixel-to-pixel transformation," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 8828–8836, 2019.

[180] R. de Lutio, A. Becker, S. D'Aronco, S. Russo, J. D. Wegner, and K. Schindler, "Learning graph regularisation for guided super-resolution," in *CVPR*, 2022.

[181] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[182] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *CVPR*, 2017.

[183] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A Comprehensive Survey on Transfer Learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2021.

[184] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: An astounding baseline for recognition," *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 512–519, 2014.

[185] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *Proceedings of the 31st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, E. P. Xing and T. Jebara, Eds., vol. 32, no. 1. Bejing, China: PMLR, 22–24 Jun 2014, pp. 647–655. [Online]. Available: https://proceedings.mlr.press/v32/donahue14.html

[186] S. Kornblith, J. Shlens, and Q. V. Le, "Do better imagenet models transfer better?" *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 2656–2666, 2019.

[187] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning Transferable Visual Models From Natural Language Supervision," 2021. [Online]. Available: http://arxiv.org/abs/2103.00020

[188] S. Lobry, D. Marcos, J. Murray, and D. Tuia, "RSVQA: Visual Question Answering for Remote Sensing Data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 12, pp. 8555–8566, 2020.

[189] R. Felix, B. Repasky, S. Hodge, R. Zolfaghari, E. Abbasnejad, and J. Sherrah, "Cross-Modal Visual Question Answering for Remote Sensing Data: The International Conference on Digital Image Computing: Techniques and Applications (DICTA 2021)," *DICTA 2021 - 2021 International Conference on Digital Image Computing: Techniques and Applications*, pp. 1–9, 2021.

[190] "Fine tuning clip with remote sensing (satellite) images and captions," https://huggingface.co/blog/fine-tune-clip-rsicd, accessed: 2023-03-19.

[191] OpenAI, "Clip," https://github.com/openai/CLIP, 2021.

[192] K. Zhou, J. Yang, C. C. Loy, and Z. Liu, "Learning to Prompt for Vision-Language Models," *International Journal of Computer Vision*, vol. 130, no. 9, pp. 2337–2348, 2022. [Online]. Available: https://doi.org/10.1007/s11263-022-01653-1

[193] M. J. Hughes and R. Kennedy, "High-quality cloud masking of landsat 8 imagery using convolutional neural networks," *Remote Sensing*, vol. 11, no. 21, 2019.

[194] M. Ding, Z. Yang, W. Hong, W. Zheng, C. Zhou, D. Yin, J. Lin, X. Zou, Z. Shao, H. Yang, and J. Tang, "CogView: Mastering Text-to-Image Generation via Transformers," *Advances in Neural Information Processing Systems*, vol. 24, no. NeurIPS, pp. 19 822–19 835, 2021.

[195] J. Yu, Y. Xu, J. Y. Koh, T. Luong, G. Baid, Z. Wang, V. Vasudevan, A. Ku, Y. Yang, B. K. Ayan, B. Hutchinson, W. Han, Z. Parekh, X. Li, H. Zhang, J. Baldridge, and Y. Wu, "Scaling Autoregressive Models for Content-Rich Text-to-Image Generation," 2022. [Online]. Available: http://arxiv.org/abs/2206.10789

[196] "Compvis/stable-diffusion-v1-1 model card at huggingface." [Online]. Available: https://huggingface.co/CompVis/stable-diffusion-v1-1#training

[197] E. Mostaque, "Cost of construction." [Online]. Available: https://twitter.com/emostaque/status/1563870674111832066

[198] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, P. Schramowski, S. Kundurthy, K. Crowson, L. Schmidt, R. Kaczmarczyk, and J. Jitsev, "LAION-5B: An open large-scale dataset for training next generation image-text models," pp. 1–50, 2022. [Online]. Available: http://arxiv.org/abs/2210.08402

[199] L. Zhang and M. Agrawala, "Adding Conditional Control to Text-to-Image Diffusion Models," 2023. [Online]. Available: http://arxiv.org/abs/2302.05543

[200] A. Bansal, E. Borgnia, H.-M. Chu, J. S. Li, H. Kazemi, F. Huang, M. Goldblum, J. Geiping, and T. Goldstein, "Cold Diffusion: Inverting Arbitrary Image Transforms Without Noise," pp. 1–22, 2022. [Online]. Available: http://arxiv.org/abs/2208.09392

[201] E. Hoogeboom and T. Salimans, "Blurring Diffusion Models," no. c, pp. 1–13, 2022. [Online]. Available: http://arxiv.org/abs/2209.05557

[202] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," *Advances in Neural Information Processing Systems*, vol. 32, no. NeurIPS, 2019.

[203] W. Zhao, L. Bai, Y. Rao, J. Zhou, and J. Lu, "UniPC: A Unified Predictor-Corrector Framework for Fast Sampling of Diffusion Models," 2023. [Online]. Available: http://arxiv.org/abs/2302.04867

[204] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2022-June, pp. 10 674–10 685, 2022. [Online]. Available: http://arxiv.org/abs/2112.10752

[205] S. Xie and Z. Tu, "Holistically-Nested Edge Detection," *International Journal of Computer Vision*, vol. 125, no. 1-3, pp. 3–18, 2017.

[206] P. Ebel, Y. Xu, M. Schmitt, and X. X. Zhu, "SEN12MS-CR-TS: A Remote Sensing Data Set for Multi-modal Multi-temporal Cloud Removal," *IEEE Transactions on Geoscience and Remote Sensing*, 2022.

[207] D. Lin, G. Xu, X. Wang, Y. Wang, X. Sun, and K. Fu, "A Remote Sensing Image Dataset for Cloud Removal," pp. 1–4, 2019. [Online]. Available: http://arxiv.org/abs/1901.00600

[208] K. Perlin, "Image Synthesizer." *Computer Graphics (ACM)*, vol. 19, no. 3, pp. 287–296, 1985.

[209] W. Dong, X. Zhang, and C. Zhang, "Generation of cloud image based on Perlin noise," *Proceedings - 2010 International Conference on Multimedia Communications, Mediacom 2010*, no. 1, pp. 61–63, 2010.

[210] X. Pan, F. Xie, Z. Jiang, Z. Shi, and X. Luo, "No-Reference Assessment on Haze for Remote-Sensing Images," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 12, pp. 1855–1859, 2016.

[211] Z. Xu, K. Wu, W. Wang, X. Lyu, and P. Ren, "Semi-supervised thin cloud removal with mutually beneficial guides," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 192, no. August, pp. 327–343, 2022. [Online]. Available: https://doi.org/10.1016/j.isprsjprs.2022.08.026

[212] A. Makarau, R. Richter, R. Müller, and P. Reinartz, "Haze Detection and Removal in Remotely Sensed Multispectral Imagery," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–11, 2013. [Online]. Available: https://pdfs.semanticscholar.org/2f54/6f38ed335c94ffb0793470f412d8dea57733.pdf

[213] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018.

[214] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *7th International Conference on Learning Representations, ICLR 2019*, 2019.

[215] "S2 MPC: Level-1 algorithm theoretical bases document," European Space Agency, Tech. Rep., 2023, ref. S2-PDGS-MPC-ATBD-L1.

[216] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of the*

Thirty-First AAAI Conference on Artificial Intelligence, ser. AAAI'17. AAAI Press, 2017, p. 4278–4284.