Evaluating Implicit Neural Representations for Storing Multi-Channel Satellite Images

Dorus Hendriks (1559524)
Lucas van Middendorp (1554069)
Maurits Flos (1455125)
Sviatoslav Gladkykh (212878)
Guus Jacobs (1300598)

Abstract

Implicit neural representations (INRs) are a recent development with growing interest for their applications in image compression. While most research has focusesed on RGB images, this research explores the application of INRs in multichannel satellite images, assessing its capability to scale with multiple (correlated) channels. We assess the performance of various existing state-of-the-art INR compression models, including COIN, MINER, and NIF, for compressing a variety of satellite images captured by the Sentinel-2 ESA earth observation mission. Our findings demonstrate that while we can utilize INRs ability to encode complex inter-channel relationships within data, their effectiveness depend on image content and specific model architecture. Models such as MINER showed promise, particularly in scaling linearly with the number of channels, yet struggled with images containing fine details. The NIF model shows scalability in terms of compression efficiency, especially with 13-channel images. Yet it suffers a noticeable loss in image quality, with PSNR values significantly lower than competing models, likely due to limitations in the quantization process. We conclude that INR-based compression methods cannot achieve similar results to traditional compression methods such as JPEG, indicating a need for novel methods. The code to run the compression algorithms can be found here: https://github.com/lucasvm2511/2AMM20-group5 and our dataset here: https://github.com/dorushendriks/2AMM20-Sattelite-Dataset-Group-5/.

ACM Reference Format:
Dorus Hendriks (1559524), Lucas van Middendorp (1554069), Maurits Flos (1455125), Sviatoslav Gladkykh (212878), and Guus Jacobs (1300598). 2024. Evaluating Implicit Neural Representations for Storing Multi-Channel Satellite Images. In . ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

The storage and compression of digital images are traditionally based on the wavelengths captured by the human eye only, particularly red, green, and blue (RGB), commonly resulting in images with 3 channels. However, for various industrial and academic purposes, it is important to look at a variety of wavelengths, including those not visible to the human eye. Satellites like the Sentinel-2 from the Copernicus project [14] capture up to 13 different wavelengths thus resulting in image data with up to 13 distinct channels. These additional wavelengths allow for various detailed analyses, such as of bodies of water, minerals [26], vegetation, atmospheric properties [9], and for measuring industrial activity [7], which could not easily be done with just the visible spectrum.

However, although satellite images can provide valuable information, the high dimensionality of the image data combined with the enormous volume of data presents a significant technical challenge. As both the volume and resolution of satellite images increase, the demands on storage and retrieval systems proliferate [12]. Current storage approaches for satellite data access programs often store the images in batches of tiles of given sizes and treat each captured wavelength as a single grayscale bitmap of 1 channel in addition to a true color representation of the wavelengths in RGB, representing how human vision would perceive the sensed wavelengths. These bitmaps are stored using both lossy and lossless conventional image compression methods, depending on the volume and expected data traffic. The problem with this approach is that as satellites capture more wavelengths with greater separation, the resulting data volume increases significantly. Some satellite images like those captured by MODIS (Moderate Resolution Imaging Spectroradiometer) have a distinction between 36 wavelengths [13], while images captured by the Hyperion had a distinction of up to 242 spectral bands [10], this amount of different channels becomes difficult to store in a near-lossless manner using traditional compression algorithms efficiently.

This difficulty stems forth from the fact that most traditional compression algorithms are optimized for low-dimensional images, typically with 3 channels (RGB). These algorithms take advantage of spatial redundancies and patterns within those channels and compress images through methods like wavelet transforms, discrete cosine transforms, and efficient quantization and encoding. However, none of these methods are specifically designed for storing the often highly related channels found in satellite images. The high interdependence between the captured wavelengths in satellite images increases in adjacent channels as more wavelengths are captured across the same spectrum as satellites maintain more

distinction. However, this interdependence is not always present amongst all captured images, natural phenomena such as wildfires, storms, and unique geographical properties can have distinctly different effects on each captured wavelength band [20]. This unreliable but often present interdependence, combined with the current difficulty in storing the satellite images efficiently makes them a great target for evaluating the usage of implicit neural network (INR)-based methods for compressing multi-dimensional images.

INRs are a type of neural network model in which a signal is encoded in a continuous representation, allowing for that signal to be reconstructed using the network. Essentially, one can query the graph for some of its attributes, like requesting the RGB values at certain coordinates for a standard image. The efficient storage of or construction of such a network can be used to effectively compress signals such as images. Unlike traditional compression methods, INRs can leverage the inherent strengths of neural networks, in that they can learn and thus encode complex relations, making them uniquely interesting from a compression point of view.

The versatility of neural networks would possibly allow INR-based methods to model the interdependence in multi-channel data more efficiently than traditional image compression methods. Thus this research aims to determine whether INR-based compression methods are specifically suited for storing highly dimensional images, by evaluating them on a representative set of multi-spectral satellite images. Specifically, we aim to address the following research questions:

- How does compression of satellite images scale with the number of output channels?
- How does INR compression compare to traditional compression methods?

2 Theoretical Background

Implicit Neural Representations (INRs) are neural networks that model signals as functions $f(x)$ of input coordinates x. When this signal is an image or a 3D object, these coordinates are spatial coordinates. This was first described for images in [24] and for 3D objects and scenes in [19] [8] [3] [16]. The 3D case was popularized by the usage of NeRFs [17]. The input coordinates can also be temporal, e.g. when the signal being modeled is a sound fragment, or a mixture of temporal and spatial in the case of video. Specifically, for images they would thus be considered as signals f_{Image}, which map pixel coordinates to colors:

$$f_{Image}(x_1, x_2) \mapsto (c_{red}, c_{green}, c_{blue}).$$

At its core in INRs, the goal is to obtain a neural network such that the signal $f(x)$ is approximated by the network $\hat{f}_\theta(x)$, where θ contains the weights, biases and other variables of the network:

$$\hat{f}_\theta(x) \approx f(x).$$

This method of considering signals as functions can be based on the universal approximation theorem, which shows that, given enough parameters, neural networks can approximate continuous functions.

INR-based compression methods seek to find the most efficient way to store this function by finding patterns in the data that lead to significant simplifications in its representation. Other novel approaches include creating efficient compression-decompression algorithms (CODEX) that can instead generate an approximation of θ from a much smaller representation. Overall the goal of seeking to compress satellite imagery with varying wavelengths can be modeled as follows: Firstly, we have satellite images containing d channels, which we can thus consider as a function mapping image coordinates (x_1, x_2) to the intensity of each of our channels and can be described as

$$f : (x_1, x_2) \mapsto (c_1, c_2, ..., c_d).$$

Then to investigate compression we seek to approximate our function with

$$\hat{f}_\theta(x_1, x_2) \approx f(x_1, x_2).$$

The quality of the compression can be assessed with various metrics that will be introduced in section 4.

3 Experimental set-up

To determine whether INR-based methods are suitable for high-dimensional images, several INR-based image compression methods are implemented and modified to take in image-like data with a variety of channels.

3.1 Dataset

To construct our dataset, we have made use of the Copernicus Data Space Ecosystem Browser [4], which offers a wide variety of environmental data for the entire world. The data is captured by the Copernicus Sentinel-2A space program and offers 13 spectral bands, ranging from 443nm to 2190nm with resolutions ranging from 10 to 60m [27]. Using the publicly available data from the Copernicus browser, we have constructed a novel dataset consisting of 1024x1024-sized images with 13 channels which can be found in the npy format here: [18]. These are initially stored losslessly as 13 separate bitmaps using the JP2 format, which for these types of single-channel images can often provide the best overall compression if looked at on a per-channel basis for lossless or near-lossless compression [15].

3.1.1 Included Images. To provide a representative set, images that are close to ones often used in differing types of research have been chosen. With one exception, images are chosen to avoid those taken on cloudy days with a cloud coverage of above 10%, as for most practical use-cases (such as crop yield analysis), satellite images are filtered and stitched such that no clouds are included [4]. An overview of which images, in addition to the reason for their inclusion, is shown below,

- **To provide a diverse sample of typical landscapes:**
  - Beijing, China – September 14, 2023
  - IJsselmeer and surroundings, Netherlands – October 17, 2023
  - Sahara Desert (South Algeria) – October 18, 2023
  - USA Suburban Sprawl – October 12, 2024
  - Alps (Spring) – April 6, 2023
  - Tokyo outskirts, Japan – July 5, 2024
  - Amazon River Rainforest – September 27, 2023
- **To represent common satellite imagery use-cases:**
  - Great Barrier Reef – October 3, 2024
  - Bahamas during Hurricane Dorian – September 2, 2019
  - Melting Coastal Ice – September 20, 2024

- Wildfires in Yosemite, USA – July 26, 2018
- Congo Deforestation – October 11, 2024

The true color channels for these images can be observed in Figure 1. Here it should be noted that these figures are created directly from 3 distinct channels without postprocessing, which leads to less recognizable features. Nevertheless, one can apply a non-invertible function to create proper RGB values from these channels as is shown in Figure 2. The reason we focus on the former instead of the more visually pleasing RGB images, is because of the aforementioned non-invertible function. By focusing on the unprocessed data, we can reduce the inherent data loss due to the compression algorithms. For completeness' sake, the RGB images for all locations in the dataset can be found in the supplemental material (Appendix A).

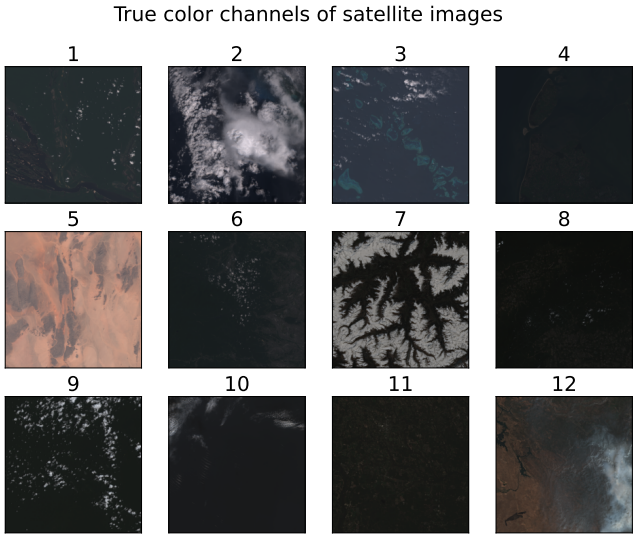

Figure 1: True color channels of 12 satellite images with max scaling.

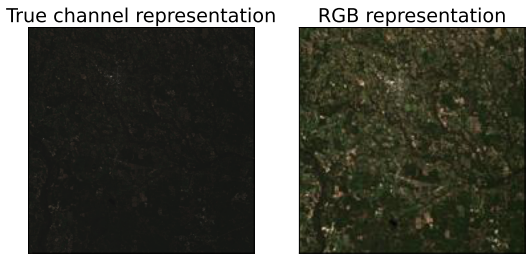

Figure 2: A comparison between an image using the true colors directly (left) and after applying several post-processing steps (right).

3.1.2 Data downsampling. Due to the different resolutions per spectral band, the sensor values must be post-processed to obtain all 13 values on a uniform grid. For this purpose, we employ a bilinear scaling to the lowest resolution (60m per pixel) such that the highest resolution bands (of 10m per pixel) are downscaled by a factor of 6. The result of these operations is a tensor of size $1024 \times 1024 \times 13$, which forms the target of our compression algorithms. For an alternative processing that uses upscaling to create small images used for classification tasks see [11].

3.1.3 Data scaling. To obtain a true color description of the dataset from the light intensity values per spectral band, a non-trivial transformation is required [22] using only 3 of the spectral bands. For our purposes, we consider the tensor obtained in the previous section to be the object that we need to compress. This choice was made to (1) prevent any loss of data from the non-invertible transformation to RGB values and (2) the spectral intensities can be used in other computations (e.g. for false color images). To maintain tractable neural networks, we further scale the data by the maximum such that all 13 features lie in the range $[0, 1]$ store the scaling factor as a way to transform the output back to the original domain.

3.1.4 Data properties. Satellite images often have strong correlations between the different channels, as in many cases the captured terrain reflects the same wavelengths. Figure 3 shows an overview of these correlations for a selection of our data. These are measured using the Pearson correlation, which measures the linear correlation and how well one channel can thus predict changes in another, and using the cosine similarity, which assesses the angle between channels as vectors, and can thus highlight the similarities regardless of magnitude.

When comparing the correlation between channels it is important to keep in mind that channels in images are often highly correlated simply due to the total luminosity of the scene. As a result, it is important to consider that high correlation amongst channels does not necessarily make them much easier to store, as even minor differences in magnitude can determine the colors in a very bright or very dark scene.

The figure clearly shows that although very strong correlations are present across channels within this data, these depend on the image type. Specifically, areas with high human activity, such as the city of Beijing, can have relatively unique data per band even for the often closely related infrared bands. While other images with low human activity and no vegetation such as those of the Sahara Desert, result in much more correlation between the channels.

Additionally, it is visible that one clear outlier within our data is the tenth channel. This channel represents the B10 spectral band, which is specifically chosen for detecting cirrus clouds and does not reach the surface. It will thus only be strongly correlated with other channels when clouds are particularly visible, such as in the image of the US wildfires. The correlation plots for the rest of the dataset are available in the supplemental material (Appendix A).

3.2 Methods

Our models of study can roughly be divided into two distinct groups: those that create a model for a single image and those that employ meta-learning. The first group allows model to signficantly overfit

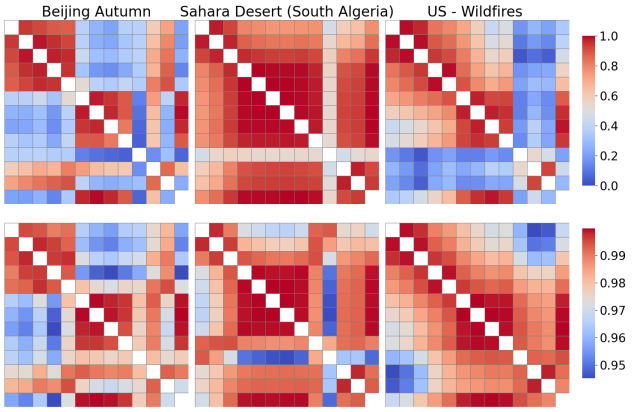

Figure 3: Similarity between the 13 channels for three Images. Absolute Pearson Correlation (Top) and Cosine Similarity (Bottom)

a particular network to the image, dedicating the entire network architecture to a single image. For this approach, it is necessary to store the weights of the entire model and the scaling factor to transform the output back to the original domain.

For the latter group however, the model architecture should generalize across the several different input images. It is here often the case that the network architecture is, at some point, rather narrow. It can be said that the output of neurons of this narrow strip of the architecture is the latent code generated by the input, since the output can be generated using the subsequent part of the network. In this approach, we can refer to the part prior to the latent code as the encoder and the subsequent part as the decoder. A major advantage of this, is that the encoder and decoder networks are shared among images, leading to a lower average storage space per image (which will be dominated by the latent codes if many images are compressed). However, there might be an inverse relation between the expressivity of the model and the quality of its outputs.

Chosen model architectures

We shall now briefly explain the state-of-the-art INR models that will be used and - where necessary - also adapted for the compression of 13-channel satellite images.

3.2.1 COIN. The SIREN-based [23] COIN (Compression with Implicit Neural representations) [5] model advances image compression by utilizing sinusoidal activation functions to effectively capture high-frequency details in image data. Unlike traditional neural networks that rely on ReLU or sigmoid activations, SIRENs [23] enable the model to learn intricate textures and fine structures inherent in images. COIN [5] encodes an image as a continuous function, mapping coordinates to pixel intensities through a neural network trained to minimize reconstruction loss. This implicit representation allows for significant parameter efficiency, as the model can achieve high fidelity with a relatively small number of parameters. Furthermore, COIN [5] employs techniques such as quantization and entropy coding to optimize storage and transmission, enabling it to achieve competitive compression rates while maintaining image quality. Overall, the SIREN architecture's ability to represent images continuously and accurately positions COIN as a promising solution in the field of image compression.

3.2.2 COIN++. Introduced by [6], COIN++ is an improved version of the COIN model. Its main contributions to the field is storing modulations of a base network for each image, instead of storing all weights of the INR directly. This is done by using a meta-learned base network and modulating this with a latent code. This modulation network consists of FiLM layers and its parameters are stored as a latent code that can reproduce these modulations. In the training process, both the shared base network and the modulations network are trained using Model-Agnostic Meta-Learning (MAML). These improvements result in better compression performance as well as faster compression, by needing only a couple of iterations to produce the modulations. Although not on par with SOTA compression codecs, COIN++ provides a basis for further research.

3.2.3 MINER. The Multiscale Implicit Neural Representation (MINER) [21] model is an advanced neural architecture for representing high-resolution signals, such as images, videos, and 3D models. MINER operates by breaking down signals into multiple scales, starting with a coarse approximation and progressively refining it at finer scales. Each finer layer predicts the residuals of the coarser layer, allowing the model to improve the output in a stepwise fashion.

By using smaller neural networks at each scale, MINER enhances computational efficiency and parallelism. This structure also allows flexible control over model size and output quality, depending on the number of scales used. The model further optimizes performance by refining only the areas that require additional detail, thanks to early pruning techniques.

For compression, MINER saves the model states at each scale, enabling efficient reconstruction of the original signal. The lightweight nature of this reconstruction process makes it suitable for high-dimensional data, like multi-channel satellite images, providing an adaptable and scalable solution for storage and retrieval.

3.2.4 NIF. Neural Image Format (NIF) [2] is a novel INR-based image compression method that maps input coordinates to pixels. This is achieved by training a representational network. Compared to earlier INR-based methods, the NIF method has significantly lower computational and memory requirements while achieving an improved reconstruction quality, thereby making encoding high-resolution images more feasible.

The architecture consists of two modules, namely the *Genesis Network* and the *Modulation Network*. The *Genesis Network* takes coordinates as input and calculates the features of the corresponding pixel. The network contains sinsuoidal activations (SIREN) and Bottleneck Layers in which the number of features is reduced proportionally to its depth. The *Modulation Network* varies the period of the sinusoidal activations of the Genesis Network to allow it to adapt to variations in frequency across different regions of the image. This leads to more flexible and accurate image representation.

After the network has been trained, its parameters are quantized and compressed to minimize the size of the encoded data. The NIF method outperforms established image compression methods in encoding high-resolution images with high speeds.

3.3 Model evaluation

Each image value takes up 16 bits or 2 bytes, which results in input images of size

$$1024 \times 1024 \times 13 \times 2 \text{ B}/1024\text{B/KB} = 26,624 \text{ KB}.$$

It is clear that any compression algorithm should have a smaller storage size than the aforementioned number, while still achieving a reasonable quality. It was observed that training of the models and inference over the spatial domain is highly parallelizable and can take up to 96 GB of GPU memory. For this reason, we have opted to utilize the resources of Dutch national supercomputer Snellius [25], namely an NVIDIA H100 Tensor Core GPU. This significantly sped up training and evaluation.

4 Results

4.1 General set-up

In this section, we will lay out the general procedures and measures that are relevant for compression the satellite images. We evaluate the quality of our approximation using Peak Signal-to-Noise Ratio (PSNR), which can be calculated as follows:

$$\text{PSNR} = 10 \log_{10} \left(\frac{\text{MAX}^2}{\text{MSE}} \right),$$

where MAX is the maximum pixel value and MSE is the mean squared error. Frequently the maximum pixel value will be 255 for 8-bit images. In our case, however, 16 bit float values will be used for the images such that we will compute the maximum signal directly from the data. The MSE can be calculated as follows

$$\text{MSE} = \frac{1}{W \cdot H} \sum_{x=1}^{W} \sum_{y=1}^{H} \left(f(x,y) - \hat{f}_\theta(x,y) \right)^2,$$

where W and H are the width and height respectively and here equal to 1024. Note that the PSNR metric is invariant to any scaling of the pixels by constant factors. It should be noted that this metric is also one that is employed in many of our compression models as a loss function which allows the backpropagation step. By comparing the size of the total stored data of various INR compression methods for set PSNR targets, it can determined whether INR-based compression methods could be uniquely suitable for storing satellite images. The literature on image compression here often uses the bits-per-pixel or bpp metric, which denotes the number of bits required to store a single pixel of the image. This value can be easily computed by dividing the storage space of our compression by the width and height of the input image. In this paper, we expand on this metric to consider the bpp per channel value, which takes into account that we will compress both the 13-channel images directly and also only the 3 true color channels. This allows for a more fair comparison between methods, as well as directly extracting information on the scaling of various models with the number of channels.

4.2 Approaches

We shall now discuss the performance of each of our chosen models. Then, in section 5, we shall aggregate our insights and draw general conclusions about the INR-based compression techniques.

4.2.1 COIN. To address the research questions outlined in Section 1, we adapted the COIN [5] model (see Section 3.2.1) to compress 13-channel images by modifying the architecture, specifically the size of the final layer. To evaluate how network architecture influences compression quality, we systematically varied the number of hidden layers and neurons per layer. In our experiments, we tested networks with 1 to 3 hidden layers, and the number of neurons per layer ranged from 5 to 200, with an increment step of 5. This resulted in a total of 120 distinct networks trained per image. Additionally, the same set of models was trained on standard 3-channel images to compare compression performance between the two types of images.

All models were trained under a consistent configuration, as follows:

- Learning rate: 1e-3
- Maximum number of steps: 10,000 (although training typically terminated early due to early stopping)
- Early stopping: patience of 50, with a minimum change in loss of 1e-4
- Loss function: Mean Squared Error (MSE)

Model with the lowest loss is selected for the future evaluation purposes.

The training was conducted using an NVIDIA Tesla P100 GPU.

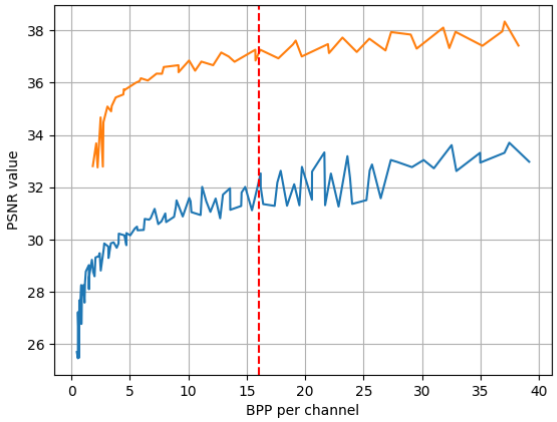

Figure 4: PSNR against BPP per channel plot for the COIN [5] model of both 3-channel (orange) and 13-channel (blue) images. The red dotted line corresponds to the BPP per channel of the uncompressed image.

To visualize the trade-off between model size and quality for both 3-channel and 13-channel images, we plotted Bits-per-Pixel per channel against Peak Signal-to-Noise Ratio (PSNR), as shown in Figure 4. The BPP per channel metric can be interpreted as "How many bits we need to store a single value in the image tensor" and is calculated based on the size of the file containing the model's weights, stored in standard 32-bit floating-point precision without additional compression, ensuring fairness in the comparison. Since the model sizes are fixed for each image (and, consequently, the BPP per channel), we averaged the PSNR values across all images

in the dataset, separately for 3-channel and 13-channel images. The blue curve in Figure 4 represents the performance on 13-channel images, while the orange curve corresponds to 3-channel images. Additionally, we include a red vertical line in the plot, representing the BPPs per channel of the original uncompressed representations of the images (16 bits), as found in the dataset. This allows us to assess whether the compression achieved by the model provides a meaningful reduction in size compared to the original representations.

Based on the plot shown in Figure 4, it is evident that COIN [5] models trained on 13-channel images significantly underperform compared to those trained on 3-channel images. Notably, there is no PSNR value achieved for 13-channel images that can be matched by simply proportionally increasing the model size used for the corresponding PSNR in the 3-channel case. Our hypothesis for this outcome is that the image resolution (1024×1024) is too large for a single MLP-based model, particularly when accounting for the additional 10 channels in the satellite images, which increases the complexity of the data and makes it harder for the model to maintain comparable performance. The results might improve if we consider training COIN [5] models separately for different regions of the image. However, our objective was to demonstrate how the standard COIN [5] model performs in our experimental setup without making significant modifications to the approach. By maintaining consistency, we aimed to assess the inherent limitations of the model when applied to high-resolution, multi-channel satellite images in its default form.

4.2.2 COIN++. Even though COIN++ was introduced as an improvement over the original COIN model, this model was optimized for standard datasets such as CIFAR and MNIST. Since the model uses meta-learning and latent variables as a representation, the model was trained on the national supercomputer to accelerate the training process on these larger images. Some hyperparamater to tune for COIN++ are: dimensionality of the latent representation, size of the base network, dimensionality of the base network and the architecture for the modulating network.

Even though the architecture is promising, the implementation was not optimized for large images, causing out-of-memory errors during training. The configuration had to be adjusted for this computation constraint, but this did not provide a suitable compression performance to properly answer the research question.

Furthermore, the model is well-suited for dataset with a common basis, such as MNSIT digits. Here the modulations to a shared network could come in handy. However, we hypothesize that the big diversity of satellite images does not allow for efficiently utilizing this base network, because of the lack of shared elements.

4.2.3 MINER. As laid out in subsection 3.2, the MINER model iteratively improves its predictions by estimating the error of prior predictions at increasingly fine scales. As such, the model readily allows us to extract datapoints on the quality of compression per image by considering the model that uses all scales up to some $j \leq L$, where L is the total number of layers. During the training procedure, we have observed that some hyperparameter settings consistently gave superior results and as such, they are considered fixed in this section. These are as follows: a kernel size and stride of 2 for the folding operations, only 1 hidden layer, a learning rate of 5×10^{-4}, a maximum number of 4096 chunks that the image is split into at each scale and a stopping and target MSE value of 10^{-6}. All of the images could be compressed in just 5 minutes on a NVIDIA GeForce RTX 4070 SUPER graphics card. The results of using this model with 6 scales are presented in Figure 5.

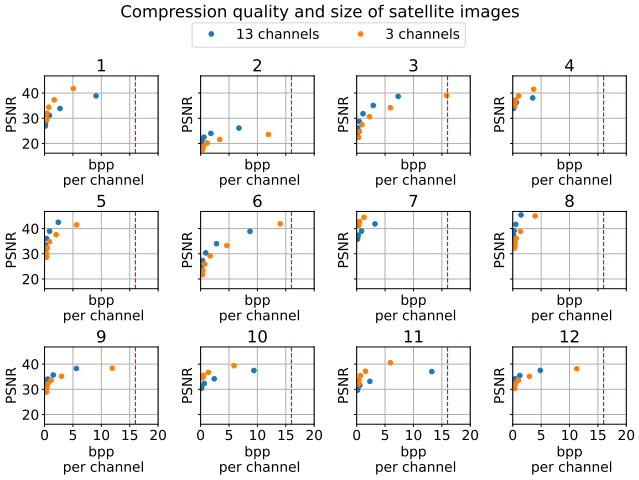

Figure 5: Compression plots using the MINER model at different scales with one plot per 1024×1024 13-channel satellite image, where the red line denotes the 16 bits that is originally used to store 1 value.

First of all, it is readily observed that the model achieves very good (around 40 PSNR) quality for every satellite image except for the second, which is only compressed to to around 25 PSNR. From Figure 1 and diagnostics from the training procedure, this mismatch is mainly due to the many locally differing features that are present in the clouds. Since MINER is a multiscale model, it would need to devote a lot of resources to accurately describe all small differences and is therefore less suited to this type of images. As for the comparison between the 3 and 13 channel images, we observe that the overall information per bit is higher for 13 channels about half of the time (images 2, 3, 5, 8, 9 and 12) and vice versa for the remaining half. All in all, this would suggest that this architecture scales linearly with the number of channels at a constant level of quality.

We shall now focus our attention on the quality of compression for individual images. First of all, we note that the model achieves a compression of at least 50% with a PSNR of at least 35 in 7 of the 12 cases and if we allow 10 bits per pixel per channel, we find that 9 out of 12 achieve this quality. It is imperative that we also observe in which cases the model performs less well. Notably, at the aforementioned quality level and storage size, we cannot compress images 2, 6 and 11. From an investigation of the model metrics, precisely these images show an unusually high number of very fine scales. This is indicative of very fine-grained features that are difficult to predict in the coarser layers and therefore corresponds to a more detailed image. Due to the architecture, it is readily explainable that the MINER model is most performant when there are fewer local variations in the images. This is a direct result of the

early pruning practices that MINER employs when it subdivides the prediction grid into various scales.

4.2.4 NIF. To run the experiments for the NIF model, a few modifications had to be made. First, the model had to be adapted to take 13-dimensional NPY files [1] as input. Additionally, scaling was implemented using Min-Max scaling. This prevents the decompressed images from having a white hue over them due to the pixel values being scaled too high during the (de)compression process. This is caused by quantization scaling the image in completely different ranges compared to the original image. Finally, the custom loss function that is originally used in the NIF paper is disregarded. Instead, we alternate between L1 Loss and MSE Loss as part of the hyperparameter tuning.

Additional hyperparameter tuning is performed by varying the *hidden sizes* of the model (i.e. of the Genesis and Modulator networks) and the configurations of the *weights restart* process. The first option for the hidden sizes is [120, 40, 28, 16] & [16,8] for the Genesis and Modulator networks, respectively. The second option is [680, 260, 65, 30] & [128, 32, 16]. In both cases the option to use cache is set to False. The weights restart process has its intervals for restart parameters $[A_{start}, A_{end}]$ set to [0.125, 0.7] or [0.175, 0.65], while $[R_{start}, R_{end}]$ is kept constant at [0.9, 0.2]. The names of the hyperparameter configurations are formatted as $\mathbf{l}x\mathbf{h}y\mathbf{w}z$, where $x, y, z \in \{1, 2\}$ indicate the loss function, hidden sizes, and weights restart setup, respectively. All other hyperparameters have been left untouched and have thus stayed in their default settings as set in the original NIF paper [2].

The NIF model is trained for each unique combination of images and hyperparameter configuration. Additionally, each of these combinations is trained using the full 13-dimensional image as well as using only the RGB channels. This results in a total of 192 NIF models being trained on the Snellius supercomputer. The results are shown in Figure 6 and Figure 7. There does not seem to be a pattern visible in the data in these plots. Most of the bpp per channel (bppc) values for the rgb images lie in the range 0.5-0.55, while these same values for the 13-channel images are in the 0.11-0.13 range. Since the NIF model, for similar PSNR values, achieves (at least 4 times) lower bppc values for the 13D images compared to the rgb images, it is clear that the compression performed by the NIF model scales quite well with the number of output channels.

It makes sense that all datapoints in Figures 6 and 7 are in the same bppc range, since this metric is affected by the model architecture and size of the latent space, which should have been varied in identical ways for the 3- and 13-channel images. The small variations in bppc are likely to be caused by the quantization that is performed by the NIF model.

The compression quality of the NIF model is subpar, with PSNR values never exceeding 35 and in some cases even reaching values below 20. A possible explanation for this could be that quantization destroys many of the patterns that the model learned during training. PSNR values were consistently very high (in the range 45-48) during training, sometimes even reaching values of over 50. Future work can focus on the implementation of different quantization schemes, or leaving quantization out entirely.

Tables 1 and 2 show that the configurations with the second setup of hidden sizes (which is indicated by *h*2 and larger compared to the first setup) results in higher PSNR values and thereby better quality compression. Unfortunately, this does not outweigh the larger increase in bppc for these models. Furthermore, the second option for *weights restart* seems to slightly outperform the first option, but the difference seems to be too small to be significant. Still, it might be interesting to decrease the intervals for restart parameters to be varied during the training process even more in future work. Finally, switching between L1 and MSE loss does not seem to affect the performance of the model.

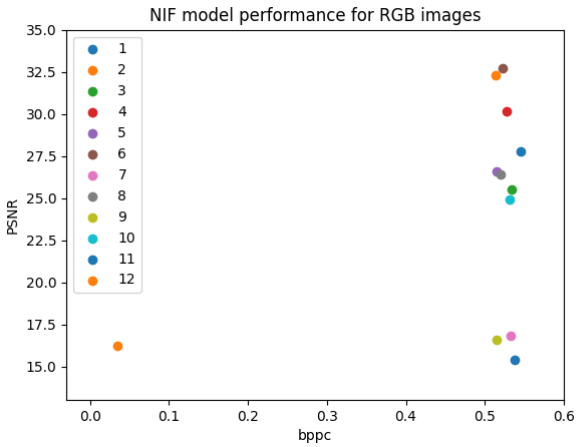

Figure 6: Performance of the NIF model as trained on 12 different RGB images, where bppc represents the Bits per pixel per channel. For each image, only the result of the model configuration that results in the highest PSNR value for that image has been plotted.

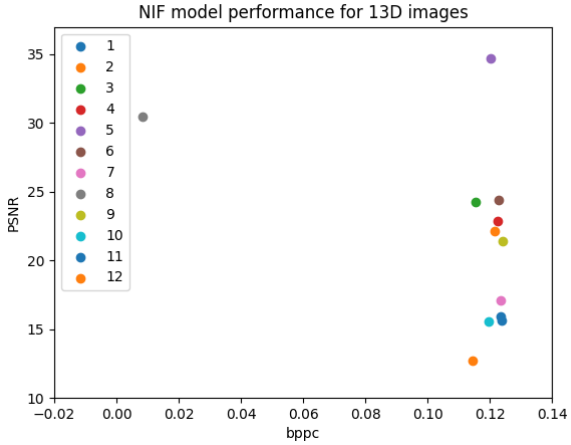

Figure 7: Performance of the NIF model as trained on 12 different 13D sattelite images. For each image, only the result of the model configuration that results in the highest PSNR value for that image has been plotted.

Model Configuration	Average bppc	Average PSNR	bppc for max PSNR	Maximum PSNR
l1h1w1	3.58e-2	20.5	3.63e-2	29.8
l1h1w2	3.58e-2	21.3	3.63e-2	29.8
l1h2w1	5.31e-1	23.9	5.22e-1	31.7
l1h2w2	5.27e-1	23.9	5.23e-1	32.7
l2h1w1	3.58e-2	20.5	3.63e-2	29.8
l2h1w2	3.57e-2	21.3	3.63e-2	29.8
l2h2w1	5.31e-1	23.9	5.22e-1	31.7
l2h2w2	5.27e-1	23.9	5.23e-1	32.7

Table 1: Performance of the NIF model for the rgb images averaged over each hyperparameter configuration. The column on the right shows the highest PSNR value as obtained by a configuration, and the column left of that shows the accompanying BPP per channel.

Model Configuration	Average bppc	Average PSNR	bppc for max PSNR	Maximum PSNR
l1h1w1	8.28e-3	17.7	8.54e-3	30.5
l1h1w2	8.26e-3	17.9	8.39e-3	29.3
l1h2w1	1.21e-1	21.1	1.20e-1	34.7
l1h2w2	1.21e-1	21.1	1.21e-1	34.7
l2h1w1	8.28e-3	17.7	8.53e-3	30.5
l2h1w2	8.26e-3	17.9	8.39e-3	29.3
l2h2w1	1.21e-1	21.1	1.20e-1	34.7
l2h2w2	1.21e-1	21.1	1.21e-1	34.7

Table 2: Performance of the NIF model for the 13-channel images averaged over each hyperparameter configuration.

4.2.5 JPEG. To validate our approach, we use the widely adopted JPEG codec on both the 3-channel and 13-channel image, to see whether NF compression actually outperforms standard codecs and to see whether it provides benefits for multi-channel images. JPEG employs discrete cosine transform (DCT) and quantization to reduce file size while preserving visual quality. Figure 8 shows the compression results for the JPEG images. Each datapoint corresponds to a different quality setting of the compression, in the range (80,100) We can see that it does a good job for both the 3-channel and 13-channel images, with bpp per channel values of below 5. Furthermore, we see that the size needed for 13 channels is comparable to 3 channels, meaning JPEG efficiently scales with channels, even though JPEG is not designed to use these extra channels. In general, the compression performance is better than the INR models.

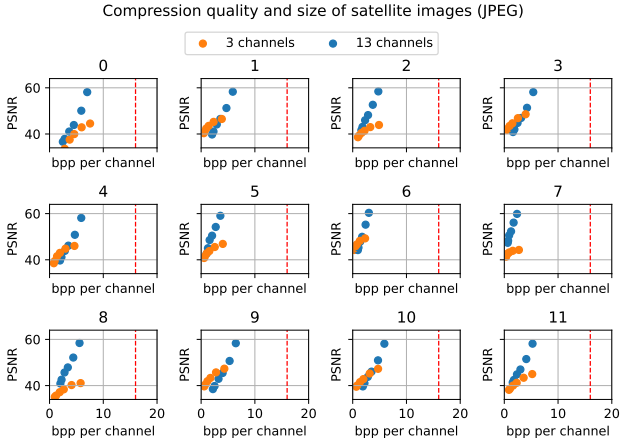

Figure 8: Compression plots using the JPEG codec at different quality levels with one plot per 1024 × 1024 13-channel satellite image, where the red line denotes the 16 bits that is originally used to store 1 value.

5 Conclusions

This research investigated the application of Implicit Neural Representations (INR) to compress high-dimensional satellite images and compared the performance of different INR methods against traditional compression techniques like JPEG. The findings reveal that the performance of INR-based methods varies significantly depending on the chosen model architecture and the nature of the image data.

The COIN model failed to efficiently compress large, multi-channel satellite images. This simple MLP-based architecture was not able to handle complexity introduced by additional channels, which indicates that it is not well-suited for high-dimensional images.

Due to problems with running, computational and time constraints, we were unable to evaluate the COIN++ model properly.

On the other hand, the MINER model achieved more promising results. Its multi-scale approach allowed for reasonably good compression in most cases, though it struggled with images containing fine local variations, such as clouds or highly detailed landscapes. All in all, MINER demonstrated linear scalability with the number of channels, suggesting its potential for satellite imagery where channel dependencies are more predictable.

The NIF model shows scalability in terms of compression efficiency, particularly when trained on 13-channel images compared to RGB images. However, the quality of compression suffered, as reflected by lower PSNR values. This indicates that while the NIF model can store data very compactly, it significantly loses image quality in the process.

When compared to traditional methods, JPEG significantly outperformed the INR-based models in both quality and computational efficiency. With PSNR values regularly exceeding 40 across all image types, JPEG maintained superior compression at a fraction of the computational cost, demonstrating that traditional methods

still outperform neural network-based approaches in practical scenarios.

Future Work

Future research may focus on addressing the limitations identified in this study. First, optimizing the COIN++ model to assess its suitability for multi-channel data. Additionally, more extensive hyperparameter tuning could improve the performance of models like MINER and NIF. Developing new or evaluating more novel INR architectures specifically designed for multi-channel compression could yield better results. Testing on larger and more varied datasets is also necessary to validate the findings. Finally, exploring hybrid approaches that combine traditional compression methods with INR models may offer a balanced solution between compression quality and efficiency.

References

[1] [n. d.]. Numpy NPY format. https://numpy.org/devdocs/reference/generated/numpy.lib.format.html. [Accessed 24-10-2024].

[2] Lorenzo Catania and Dario Allegra. 2023. NIF: A Fast Implicit Image Compression with Bottleneck Layers and Modulated Sinusoidal Activations. In *Proceedings of the 31st ACM International Conference on Multimedia* (Ottawa ON, Canada) *(MM '23)*. Association for Computing Machinery, New York, NY, USA, 9022–9031. https://doi.org/10.1145/3581783.3613834

[3] Zhiqin Chen and Hao Zhang. 2019. Learning Implicit Fields for Generative Shape Modeling. arXiv:1812.02822 [cs.GR] https://arxiv.org/abs/1812.02822

[4] Copernicus Open Access Hub. 2024. Copernicus Browser - Sentinel Data. https://browser.dataspace.copernicus.eu/ Accessed: 2024-10-17.

[5] Emilien Dupont, Adam Goliński, Milad Alizadeh, Yee Whye Teh, and Arnaud Doucet. 2021. COIN: COmpression with Implicit Neural representations. arXiv:2103.03123 [eess.IV] https://arxiv.org/abs/2103.03123

[6] Emilien Dupont, Hrushikesh Loya, Milad Alizadeh, Adam Goliński, Yee Whye Teh, and Arnaud Doucet. 2022. COIN++: Neural Compression Across Modalities. arXiv:2201.12904 [cs.LG] https://arxiv.org/abs/2201.12904

[7] Christopher D Elvidge, Mikhail Zhizhin, Tamara Sparks, Tilottama Ghosh, Stephen Pon, Morgan Bazilian, Paul C Sutton, and Steven D Miller. 2023. Global Satellite Monitoring of Exothermic Industrial Activity via Infrared Emissions. *Remote Sensing* 15, 19 (2023), 4760.

[8] SM Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S Morcos, Marta Garnelo, Avraham Ruderman, Andrei A Rusu, Ivo Danihelka, Karol Gregor, et al. 2018. Neural scene representation and rendering. *Science* 360, 6394 (2018), 1204–1210.

[9] Alfonso Fernández-Manso, Oscar Fernández-Manso, and Carmen Quintano. 2016. SENTINEL-2A red-edge spectral indices suitability for discriminating burn severity. *International journal of applied earth observation and geoinformation* 50 (2016), 170–175.

[10] Mark A Folkman, Jay Pearlman, Lushalan B Liao, and Peter J Jarecke. 2001. EO-1/Hyperion hyperspectral imager design, development, characterization, and calibration. *Hyperspectral Remote Sensing of the Land and Atmosphere* 4151 (2001), 40–51.

[11] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. 2019. EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 12, 7 (2019), 2217–2226. https://doi.org/10.1109/JSTARS.2019.2918242

[12] Mahdi Jemmali, Wadii Boulila, Asma Cherif, and Maha Driss. 2023. Efficient Storage Approach for Big Data Analytics: An Iterative-Probabilistic Method for Dynamic Resource Allocation of Big Satellite Images. *IEEE Access* 11 (2023), 91526–91538. https://doi.org/10.1109/ACCESS.2023.3299213

[13] CO Justice, JRG Townshend, EF Vermote, E Masuoka, RE Wolfe, Nazmi Saleous, DP Roy, and JT Morisette. 2002. An overview of MODIS Land data processing and product status. *Remote sensing of Environment* 83, 1-2 (2002), 3–15.

[14] S Jutz and MP Milagro-Perez. 2020. Copernicus: the european earth observation programme. *Revista de Teledetección* 56 (2020), V–XI.

[15] Michael W Marcellin, Michael J Gormish, Ali Bilgin, and Martin P Boliek. 2000. An overview of JPEG-2000. In *Proceedings DCC 2000. Data compression conference*. IEEE, 523–541.

[16] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. 2019. Occupancy Networks: Learning 3D Reconstruction in Function Space. arXiv:1812.03828 [cs.CV] https://arxiv.org/abs/1812.03828

[17] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. arXiv:2003.08934 [cs.CV] https://arxiv.org/abs/2003.08934

[18] Group 5 of 2AMM20. 2024. Multi-channel Satellite Image Dataset. https://github.com/dorushendriks/2AMM20-Sattelite-Dataset-Group-5.

[19] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. 2019. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. arXiv:1901.05103 [cs.CV] https://arxiv.org/abs/1901.05103

[20] Dimitris Poursanidis and Nektarios Chrysoulakis. 2017. Remote Sensing, natural hazards and the contribution of ESA Sentinels missions. *Remote Sensing Applications: Society and Environment* 6 (2017), 25–38.

[21] Vishwanath Saragadam, Jasper Tan, Guha Balakrishnan, Richard G. Baraniuk, and Ashok Veeraraghavan. 2022. MINER: Multiscale Implicit Neural Representations. arXiv:2202.03532 [cs.CV] https://arxiv.org/abs/2202.03532

[22] Sentinel Hub. 2024. L2A Optimized - Custom Scripts. https://custom-scripts.sentinel-hub.com/sentinel-2/l2a_optimized/ Accessed: 2024-10-17.

[23] Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. 2020. Implicit Neural Representations with Periodic Activation Functions. arXiv:2006.09661 [cs.CV] https://arxiv.org/abs/2006.09661

[24] Kenneth O Stanley. 2007. Compositional pattern producing networks: A novel abstraction of development. *Genetic programming and evolvable machines* 8 (2007), 131–162.

[25] SURF. 2024. Snellius: de Nationale Supercomputer. https://www.surf.nl/diensten/snellius-de-nationale-supercomputer. Accessed: 2024-10-17.

[26] Harald Van der Werff and Freek Van der Meer. 2015. Sentinel-2 for mapping iron absorption feature parameters. *Remote sensing* 7, 10 (2015), 12635–12653.

[27] Tianxiang Zhang, Jinya su, Cunjia Liu, and Wen-Hua Chen. 2018. Potential Bands of Sentinel-2A Satellite for Classification Problems in Precision Agriculture. (06 2018). https://doi.org/10.1007/s11633-018-1143-x

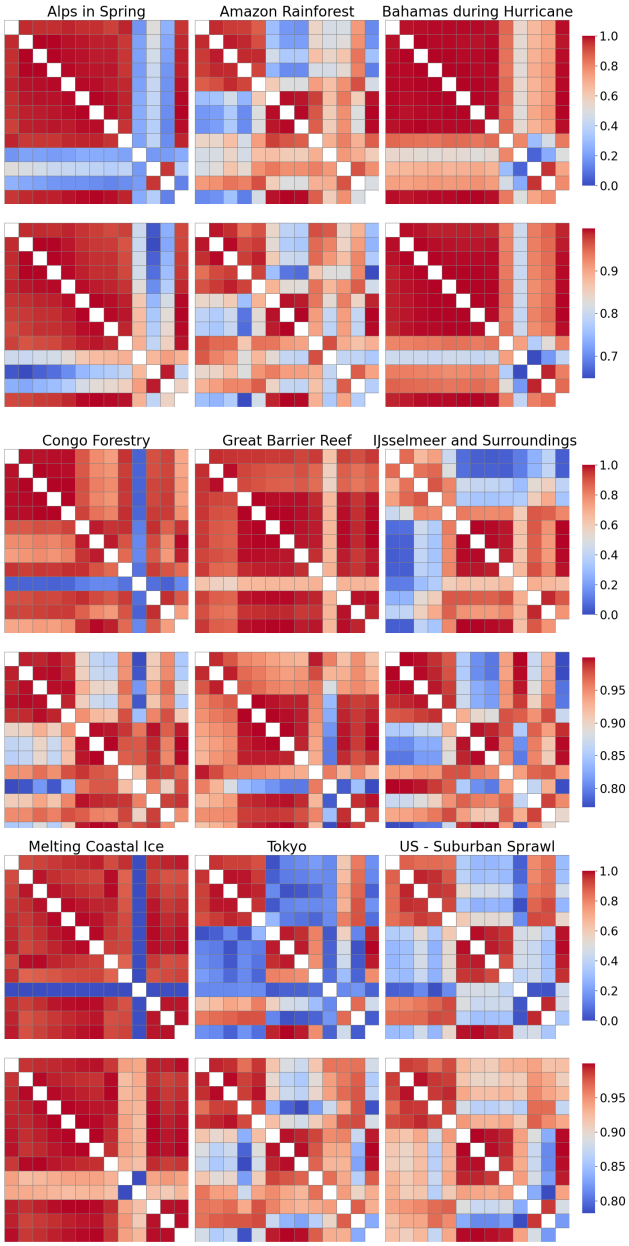

Figure 10: Similarity between the 13 channels for the nine Images that were not shown in Figure 3. Absolute Pearson Correlation (Top row for each group of 3 images) and Cosine Similarity (Bottom row for each group of 3 images).

A Dataset

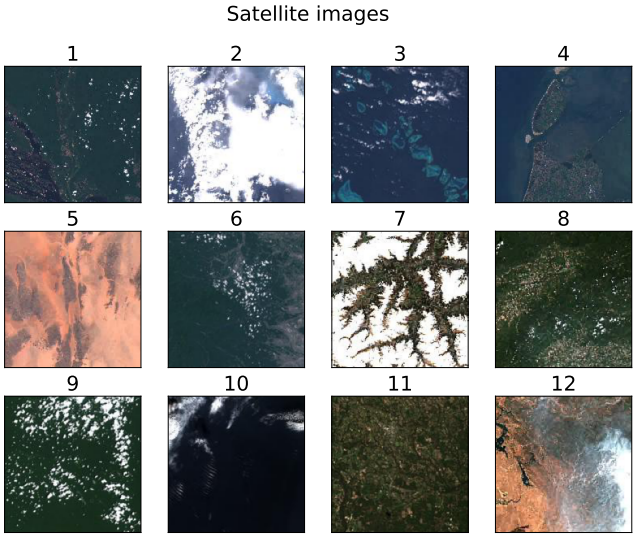

Figure 9: RGB representations of the satellite images.