MSc Artificial Intelligence
Master Thesis

Learning to Adapt:
A Representation Learning Fine-Tuning Method for Incomplete Multispectral Imagery

Darie Petcu
14628937

August 18, 2024

48 ECTS
December 2023 - August 2024

Supervisor:
Inske Groenen

Examiner:
Dr Arnoud Visser

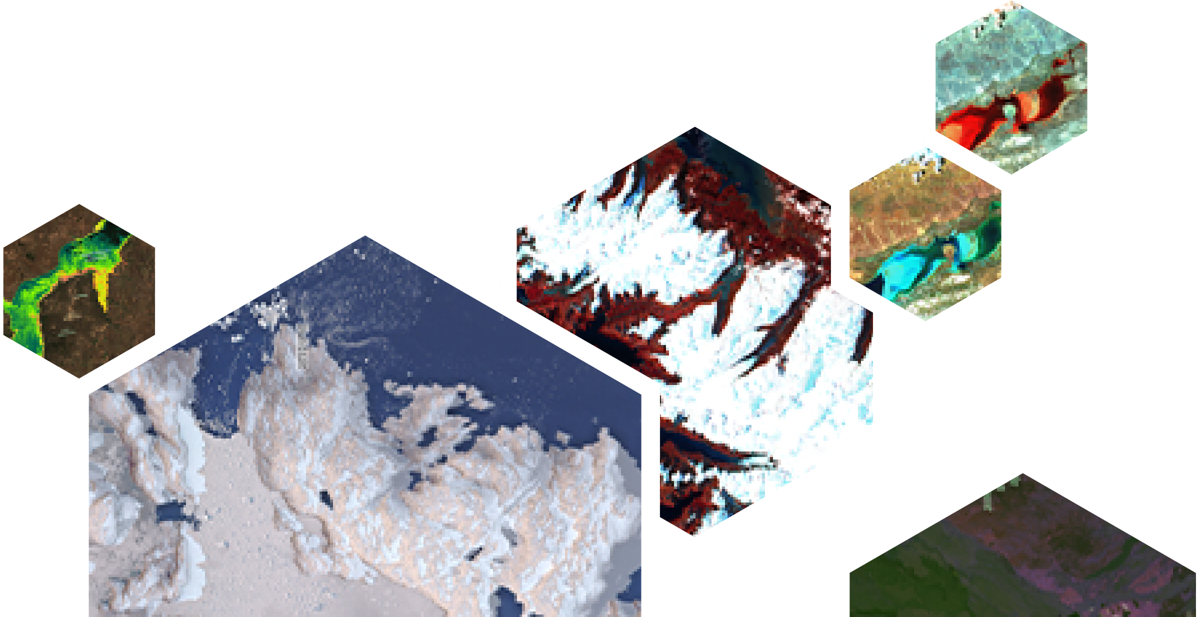
Second reader:
Dr Stevan Rudinac

Universiteit van Amsterdam

Abstract

With recent years' leaps forward in the field of deep learning, including input masking at training time, it is now easier for AI models to process incomplete data. In the field of remote sensing, data is often discarded due to imperfections such as dead pixels. This results in wasting collected information that could be used for training. The newest deep learning studies on multispectral satellite imagery present findings that also apply under conditions of incomplete data. Yet, using incomplete data as a central part of the model training procedure - in the form of masking certain spectral bands - has not been studied in-depth.

This paper investigates the effects of masking one or multiple spectral bands of multispectral satellite imagery. A contrastive learning fine-tuning method is proposed to infer the missing information, by leveraging the learnt representations from a foundation model encoder. Then, the effect of the proposed fine-tuning regime is evaluated on two downstream tasks: scene classification and image segmentation. Results suggest that recovering information within the masked spectral bands from the unmasked ones is possible. Scene classification performance on incomplete input increases when using an encoder that was fine-tuned via the proposed method. However, results have been inconclusive on the image segmentation task.



Contents

1 Introduction 2

2 Literature Overview 5
2.1 Background Setting . . . 5
2.1.1 Remote Sensing . . . 5
2.1.2 Representation Learning and Foundation Models . . . 6
2.1.3 Masked and Incomplete Input . . . 8
2.2 Related Work . . . 9
2.2.1 Used Pre-Trained Models . . . 9
2.2.2 Benchmark Datasets and Downstream Tasks . . . 10

3 Method 12
3.1 Data . . . 12
3.1.1 Pre-Processing and Augmentations . . . 12
3.1.2 Image Masking . . . 12
3.2 Model Architecture . . . 14
3.2.1 Encoder . . . 14
3.2.2 Decoders . . . 14
3.3 Learning Objective . . . 15
3.3.1 Baseline Experiment . . . 15
3.3.2 Downstream Tasks . . . 16

4 Experiments 18
4.1 Baseline Experiments . . . 18
4.2 Downstream Experiments . . . 18
4.2.1 Scene Classification . . . 20
4.2.2 Image Segmentation . . . 20

5 Discussion 23
5.1 Analysis of Results . . . 23
5.1.1 Comparison between different sets of masked bands . . . 23
5.1.2 Answering the research questions . . . 24
5.2 Limitations . . . 27
5.2.1 Proposed Method . . . 27
5.2.2 Band Combinations . . . 27
5.2.3 Dataset Shortcomings . . . 28
5.3 Future Work . . . 28
5.3.1 Alternative Hypermarameters . . . 28
5.3.2 Improving the Experiments . . . 29

6 Conclusion 31

A Extra Tables 32

B Extra Figures 35

List of Figures

1.1 Landsat-1 image of Dallas in false color, 25.07.1972 2
1.2 Imperfections in EuroSAT images 3

2.1 CLIP Loss Formula 7
2.2 Cross-Entropy Loss Formula 7
2.3 Cosine Similarity Formula 8
2.4 Examples of EuroSAT Classes 10
2.5 Atmospheric Influence on EuroSAT 11

3.1 Visible RGB, Shortwave Infrared, Agricultural RGB Indexes Visualized. 13
3.2 Proposed Fine-Tuning Method 16
3.3 Average Accuracy Formula 17
3.4 mIOU Formula 17

4.1 Baseline Experiments Validation Loss 19
4.2 Plot of Scene Classification Accuracy 21
4.3 Plot of Image Segmentation Average Accuracy 22
4.4 Plot of Image Segmentation mIOU 22

B.1 Number of images per class in EuroSAT-MS dataset. 35
B.2 Comparison of Masking Schedules 36
B.3 Examples of Low Quality MODIS Labels 36
B.4 Segmentation with B8 Masked 37
B.5 Segmentation with RGB Masked 37
B.6 Segmentation with B8, B9, B10, B11, B12 Masked 38
B.7 Visualizing all bands of an image where bands B2, B3, B4 (visible RGB) were masked 38
B.8 Visualizing all bands of an image where bands B2, B8, B11 (Agricultural RGB) were masked 38
B.9 Visualizing all bands of an image where bands B4, B8, B9, B13 were masked . . 39
B.10 Visualizing all bands of an image where bands B2, B4, B8, B11 were masked . . 39
B.11 Visualizing all bands of an image where bands B8, B9, B10, B11, B12 were masked 39

List of Tables

4.1 Single-Band Baseline Experiment Results 18
4.2 Multi-Band Baseline Experiment Results 19
4.3 Scene Classification Experiment Results 20
4.4 Image Segmentation Experiment Results 21

A.1 Sentinel-2 Spectral Bands 32
A.2 Overview of complete and simplified IGBP classification scheme 33
A.3 Scene classification accuracy at different training stages, using a fine-tuned encoder. 33
A.4 Scene classification accuracy at different training stages, using a non-fine-tuned encoder. 34
A.5 Comparison of implemented masking schedules. Mask Strategy column specifies masking schedule, together with mask ratio (initial and final mask ratios are specified in the case of gradual and staircase schedules). 34
A.6 Additional experiment runs with various mask ratios 34

Chapter 1

Introduction

In 1972, the field of remote sensing took a major step forward when the first satellite multispectral images of Earth were captured by the Landsat-1 mission. The earliest image available in this satellite's archive is an aerial view of Dallas, USA, depicted in *false color* in Figure 1.1. This image displays spectral bands outside the visible light spectrum, hence the name "false color". Forty years and tens of satellite missions later, multispectral imagery is still widely used for tasks that require imagery from above. The technological advancements that have been made since then have resulted in:

1. Higher resolution images - going from a spatial resolution of $60m$ per pixel to $< 1m$ per pixel in some modern satellites; [1]

2. Larger coverage of spectral bands - certain satellites can capture up to 200+ bands to form multispectral and hyperspectral imagery [2];

3. An increase in advanced applications using these images - relying on improved computers, new scientific findings, and more recently on artificial intelligence (AI).

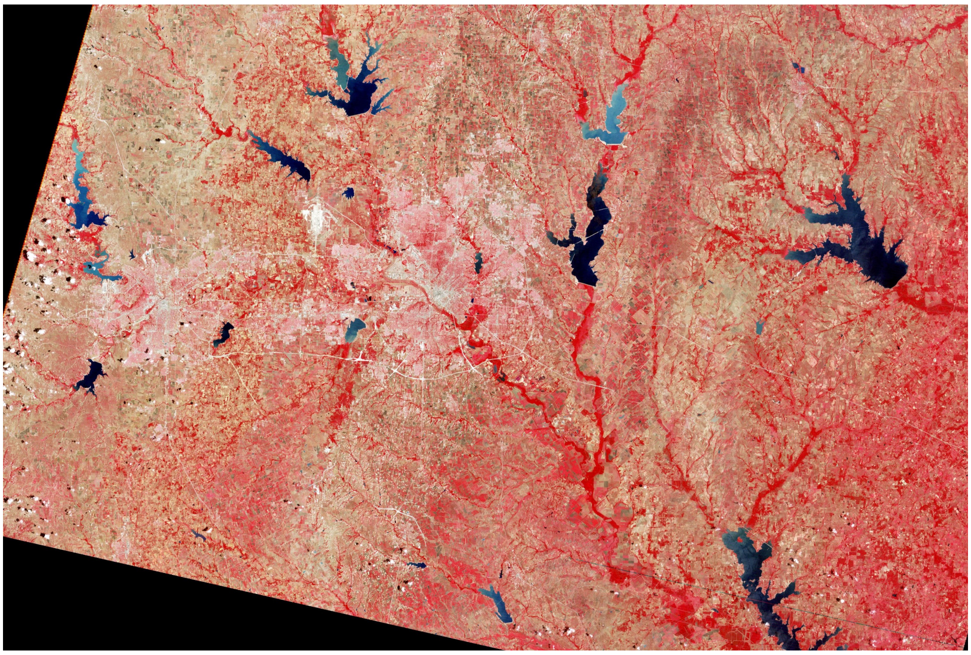


Figure 1.1: Landsat-1 image of Dallas in false color, 25.07.1972

All of the listed points can benefit from the use of AI [3]. Several typical tasks for AI have been applied to remote sensing, such as classification, image segmentation, and predicting

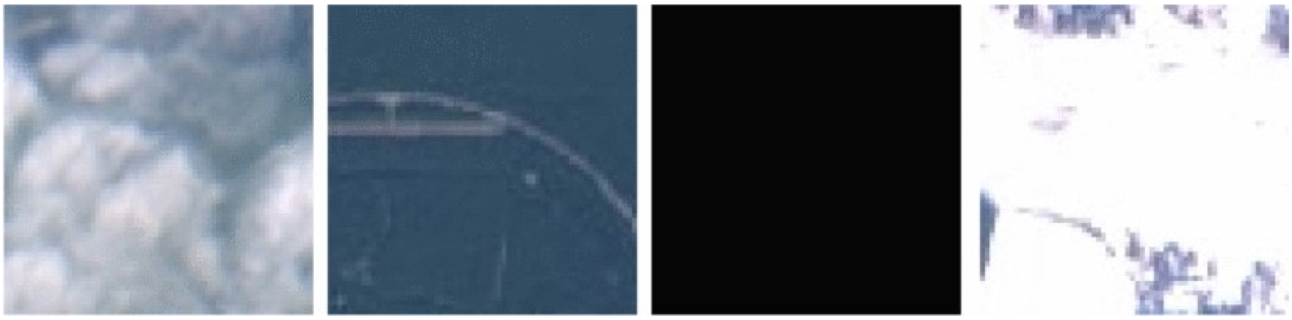


Figure 1.2: RGB view of four images sampled from the EuroSAT dataset, showcasing imperfections. From left to right: clouds, mislabeling, dead pixels, snow.

continuous numerical values. Figure 2.4 contains a number of images sampled from a benchmark remote sensing multispectral dataset used in this study, EuroSAT [4]. While EuroSAT is commonly used for scene classification and other vision tasks, it also contains examples of a commonly occurring element in Earth Observation: imperfections in the data. Scenes covered by clouds, dead pixels, ice or snow, as illustrated in Figure 1.2, can significantly affect performance on the intended task. In some cases, AI algorithms are able to bypass these imperfections and carry out their tasks regardless, which is what the experiments presented in this project focus on. However, many factors come into play here, including how imperfect the images are, how much training data is available to the AI model, and the strengths and weaknesses of the selected AI architecture. Although these conditions remain challenging, the ability of current state-of-the-art AI models in remote sensing to deal with them is improving.

The current paradigm in artificial intelligence leans towards using encoder-decoder architectures to solve tasks, for example: using Autoencoders [5] for input reconstruction, or training a task-specific decoder on a foundation model. Foundation models are high-performing, general-purpose architectures that achieve state-of-the-art on many benchmark prediction datasets [6]. When trained on large amounts of broad data, using carefully selected parameters and architectures, encoders from foundation models grow more generalisable and capable of facilitating information to the decoder. Encoders from pre-trained foundation models can be used in combination with various decoder architectures, thereby using the same model to effectively encode input for multiple purposes [7]. The goal of this study is to increase the ability of AI models to encode incomplete input, with emphasis placed on thoroughly studying and training encoders on incomplete multispectral image data.

In computer vision, foundation models are often trained by using self-supervised learning, which solely uses the unlabelled training data instead of relying on annotated datasets. Self-supervised learning is commonly formulated by using either masked input tasks (resulting in the masked autoencoder architecture [8] and masked image modelling framework [9]), or contrastive learning [10], [11]. Contrastive self-supervised learning [12] can be used to train multimodal models, which encode multiple modalities to the same embedding space, enabling processing and associating information of multiple data modalities.

In the context of remote sensing, several pioneering foundation models have been developed to process multispectral satellite imagery. Studies such as [13], [14], [15] employ the two aforementioned self-supervised techniques to train their remote sensing foundation models. These models can be used for multiple tasks pertaining to remote sensing, such as land cover segmentation, remote sensing scene classification, and crop monitoring. Given their large number of parameters, foundation models need large amounts of training data, and in certain cases have been trained on 100.000 satellite images [14].

The problem at hand is that satellite imagery data is often discarded because of imperfections, such as cloud coverage or sensor malfunctions. Considering the large amounts of training data required to optimize large models, it is a waste of resources to use the gathered data in an ineffective manner. In other applications of computer vision, progress has been made in utilising incomplete input data. [16]'s work in robotics has produced models that can adapt to missing information at the level of certain sensors, providing evidence that imperfect data does not necessarily need to be discarded. Recent remote sensing research shows their proposed models also work on incomplete images [17], [18], [19]. However, processing incomplete multispectral imagery has not yet been researched in sufficient depth.

In this paper, a potential solution to decreasing the amount of wasted data is investigated. The centerpiece of this research is using multispectral satellite imagery and masking a portion of the spectral bands. The masking technique is a way to simulate missing input data, which could arise from atmospheric conditions covering key elements, dead pixels, or file corruption. To this extent, this study proposes a fine-tuning method for encoders of multispectral satellite imagery, where some of the spectral bands have been partly or completely masked. In this paper, images that have had one or multiple bands masked shall be referred to as *incomplete images*. By fine-tuning the encoder on the incomplete images, the goal is to make it less vulnerable to gaps in the input data.

To fulfill this purpose, the main challenge lies in fine-tuning the encoder such that it produces embeddings of incomplete images that are close to the embeddings produced from complete images. A foundation model's encoder is used to create embeddings of complete images. An identical encoder if fine-tuned to produce similar embeddings from incomplete images, using representation learning. The encoder is deemed capable to adapt to and infer the missing information if it can produce these similar embeddings, following the proposed fine-tuning method.

The performance of the fine-tuning method can be measured by comparing downstream task results on complete and incomplete multispectral images, encoded using fine-tuned and non-fine-tuned encoders. Since some information is removed by masking, downstream task performance is expected to drop whether the encoder has been fine-tuned or not. Should the fine-tuned encoder be able to mitigate the performance drop better than the non-fine-tuned encoder, it can be argued that the fine-tuning method works.

For the remainder of this article, an encoder that underwent the proposed fine-tuning method on incomplete multispectral satellite images shall be referred to as a *fine-tuned encoder*. When it was not fine-tuned as such, the encoder extracted from the foundation model shall be referred to as a *pre-trained encoder*.

Main RQ: Can the fine-tuned encoder positively influence downstream task performance and enable information extrapolation under incomplete input conditions of multispectral imagery?

RQ0: Is a fine-tuned encoder able to converge on the task of aligning its embeddings of incomplete images to the embeddings of the corresponding complete images, sourced from a pre-trained encoder?

RQ1: Under conditions of incomplete images, does a decoder's performance on a downstream task increase when using a fine-tuned encoder, compared to using a pre-trained encoder?

RQ2: How does a decoder's performance on incomplete images, encoded by the fine-tuned encoder, compare to its performance on complete images, encoded by the pre-trained encoder?

RQ3: Does using the fine-tuned encoder on incomplete images affect a decoder's learning curve on a downstream task, compared to using the pre-trained encoder on complete images?

Chapter 2

Literature Overview

This chapter lays out a description of the research landscape in remote sensing and defines the gap this research aims to fill. Then, it dives into the theoretical side of techniques and resources that are relevant for this project.

2.1 Background Setting

This section is divided into multiple parts, each of which introduces one key element of this project: remote sensing, representation learning, and masked input. The greater context and theoretical background are introduced, thereby building a foundation for the next section that dives into the specific methods employed by this research.

2.1.1 Remote Sensing

Remote sensing started out with the need to examine objects using imagery captured from above. The applications of remote sensing images were many, such as tracking agricultural crops, water levels, meteorology, and mapping [20]. In the early 1970s, when the first studies within the domain of multispectral satellite imagery analysis were conducted [21], analog technology imposed strong limitations. As technology progressed, analysis became increasingly digital, advancing our research capabilities concerning remote sensing imagery. Acquiring information outside the visible light spectrum confers researchers the means to observe objects on our planet from space, even when these objects reflect light outside the visible light spectrum. A crucial step in the use of this information is choosing to examine the frequency bands that are most relevant for the objects of interest.

Spectral Band Combinations

Remote sensing uses many different sensors to register various types and modalities of information, ranging from visible light, to multi- or hyper-spectral images and LIDAR data. Due to the amount of information received from this plethora of sensors, the collected data can and should be processed in different ways, depending on the purpose it is used for. It is important to focus on the part of the data that is relevant for the task at hand.

Many satellites orbiting the Earth record multispectral imagery, such as Sentinel-2, QuickBird, and the Landsat missions [22]. Depending on the purpose for which a captured image is used, certain bands are selected. The reason behind this is the *spectral signature* of objects, which refers to the amount of light that an object reflects at different frequencies, and is particular to each type of object. For example, plants that perform photosynthesis do not reflect much ultraviolet light due to absorbing it during photosynthesis; but they reflect more light in the near-infrared (NIR) part of the light spectrum. As such, one can extract more information about objects of interest by focusing on the spectral bands that typically offer key information about these objects [23].

This paper focuses on crop & vegetation-related tasks, as described in the *Selecting Masked Bands* paragraph of Section 3.1. Given the wide array of agricultural applications for multispectral imagery, there is a strong incentive to improve the usability of satellite imagery containing artifacts in the agriculture-related bands.

When using satellite imagery for monitoring vegetation, the near-infrared and red bands are particularly useful [23]. Moreover, for agricultural tasks in general, the "Agricultural RGB" index is commonly used. This is comprised of visible blue, NIR, and shortwave-infrared (SWIR). According to [24], the normalized difference vegetation index (NDVI), commonly referred to as the vegetation index, is also very useful for observing vegetation. These make use of the visible red and near-infrared (VNIR) band, normalized using the visible red readings. Lastly, the Short Wave Infrared Index is also important for agricultural tasks: it shows vegetation in various shades of green and can reveal the density of vegetation [25]. The *incomplete images* used in this paper are missing some or all of these bands, in order to improve the task performance in vegetation-related tasks when the data is incomplete.

For visualization purposes, a combination of three bands within or outside the visible light spectrum are selected as the three color channels of an image. Such images are commonly referred to as *false color images* when they contain at least one band outside the visible light spectrum. Common false color combinations are "Color Infrared" (visible green, visible red, VNIR), "Short-Wave Infrared" (visible red, VNIR, SWIR), or Moisture Index (VNIR, SWIR).

Using AI in RS

Artificial intelligence (AI) was first used in remote sensing to support classification tasks [26, Chapter 1.]. Nowadays, a good share of RS applications are carried out with Deep Learning-based solutions[27]. Recent advancements enable tackling previously inaccessible, computationally demanding tasks, while other tasks can now be completed with more precision or shorter processing times.

In multispectral imagery, AI can be useful by extracting key features and information from different bands of an image. As detailed in Subsection 2.1.2, models can learn associations between different parts of an input, such as different channels. This can be seen in the work of [13], [11]. These examples are versions of the masked autoencoder [8] and CLIP models [10], trained on multispectral imagery.

2.1.2 Representation Learning and Foundation Models

As explained in Section 1, generalisable foundation models represent current state-of-the-art in prediction, due to their increased capacity to extract meaningful information from the input data. The field of representation learning focuses on identifying, interpreting, and potentially making use of the relevant encoded information for subsequent learning tasks. According to [6], representations in the latent space of foundation models are able to preserve relationships present in the input space.

The CLIP model and loss function

CLIP (Contrastive Language-Image Pre-training) [10] is a recent impactful advancement in representation learning that has been adopted in many domains. A non-exhaustive list of examples is: speech processing [28], 3D vision [29], image generation through diffusion models

$$\mathcal{L} = \frac{1}{2N}\left[\sum_{i=1}^{N} \mathcal{L}_{caption}(c_i, I_{1,\dots,N}) + \sum_{i=1}^{N} \mathcal{L}_{image}(I_i, c_{1,\dots,N})\right]$$

Figure 2.1: CLIP loss formula. N is the number of elements in a batch, c_i is the latent embedding of caption i, I_i is the latent embedding of image i. $\mathcal{L}$ is the cross-entropy loss.

$$l_n = -\log \frac{exp(cos(x_n, y_n))}{\sum_{c=1}^{C} exp(cos(x_n, y_c))}$$

Figure 2.2: Cross-entropy loss formula including softmax, between cosine similarity scores, for element n of a batch. C is the batch size, x_n is the embedding of element n, y_n is the embedding of the correctly matching element n in the batch, and y_c are the embeddings of all possible matching elements to element n of the batch.

[30], video-text retrieval [31]. In the domain of remote sensing, SatCLIP [14], Multimodal RS-CLIP [32] and RemoteCLIP [11] all adopt this method. Of these examples, SatCLIP is used as a pre-trained model within our study, because the authors have made the code and pre-trained model available online.

The CLIP method was initially proposed by [10]. In their study, they propose a pre-training method that incorporates contrastive learning to learn to match images to their corresponding captions and vice versa. First, two different data modalities are encoded, and the latent features from each modality are extracted. Then, trainable linear projections are used to map the extracted features to a shared multimodal latent space. After the linear projection layer, cosine similarity is used to measure the distance between pairs of embeddings from the visual and text modalities, and cross-entropy loss is used to measure the amount of image-caption pairs that have been correctly matched. Thus, a greater distance (lower similarity) between embeddings of an image-caption pair results in an increased loss value.

What this accomplishes is the latent representations of associated elements, such as an image and its caption, being pushed closer towards each other within the latent space. Due to the contrastive aspect, the non-matching embeddings are simultaneously being pushed away from each other. A well-structured latent space emerges from this method, due to the discrete nature of the text modality [33]. The emerging structure of the latent space also enables zero-shot transferring of the network to downstream tasks [10].

As can be observed in Equation 2.1 , the contrastive loss is split into two parts: the caption and the image loss objectives. In the caption part of the loss, the embedding of each caption is compared to all embeddings of images in the batch. In the image part, the embedding of each image is compared with all embeddings of captions in the batch. To perform all of these comparisons, the *cosine similarity* (Equation 2.3) is used as a similarity metric between embeddings. This similarity is scaled by the temperature parameter τ, to help with scaling in the loss computations. Then, *cross-entropy loss* is used on the similarity scores, using the index of the corresponding element as the label. Equation 2.2 shows the cross-entropy formula, and represents one single summand from the $\mathcal{L}$ elements of the CLIP Loss equation.

Using CLIP for downstream tasks

Starting from the study that introduced CLIP, this method's potential to train a classifier was also researched. [10] shows their ResNet50-based model can perform zero-shot classification.

$$cos(\mathbf{emb}_{caption}, \mathbf{emb}_{image}) = \tau \cdot \frac{\mathbf{emb}_{caption} \cdot \mathbf{emb}_{image}}{||\mathbf{emb}_{caption}||\ ||\mathbf{emb}_{image}||}$$

Figure 2.3: Cosine similarity formula between embeddings of an image and a caption. τ represents the learnable temperature parameter., $\cdot$ represents dot product.

They attribute this feat to the emergent structure within the latent space. Models that can also act as unsupervised classifiers, aside from their main purpose, can arise by applying representation learning practices to latent spaces resulting from CLIP, as seen in [34]. Moreover, models trained using CLIP have proved to be effective encoders, especially in cross-modal settings. One such example would be in Text-Conditioned Diffusion Models, as explained by [30]. These studies bring strong arguments for using CLIP during an encoder's training procedure.

2.1.3 Masked and Incomplete Input

This research focuses on input masking. Requiring the model to predict masked parts of the input is a common method to train current state-of-the-art architectures, such as the masked autoencoder [8], Mask R-CNN [35], and in Natural Language Processing the Generative Pretrained Transformer [36] and BERT language encoder [37]. Examples of vision transformer-based architectures trained on multispectral RS data that support input masking are [13] and [17].

Masking appears to provide multiple useful advantages to effectively train a neural network. First, it is an accessible way of creating a self-supervised learning objective: without requiring any labels, parts of the input can simply be masked to create an input reconstruction learning objective. Thus, the network is forced to use the broader context from the input in order to fill in the missing parts [8]. This reasoning determined the decision to adopt input masking in this project by masking some of the color bands.

Second, it creates a non-trivial task for the network. Only allowing a model to observe parts of the input should lead to the model learning associations between observed features. If the complete input were available instead, such associations might not have been captured by the model, due to being shadowed by other, more prominent features. This training objective, in turn, should result in a network that models deeper connections between features of the input, thereby becoming more precise at extracting and structuring information.

In order to implement masking, a number of decisions must be made, such as what to mask and how to mask it. Masking can be accomplished by either removing the selected content altogether, or replacing it with a *mask value*. According to [38], examples of basic masking are the in-sequence prediction that occurs in "Image GPT" (iGPT) [39] and the random patch masking in the Masked Autoencoder [8]. Once the input parts that should be masked have been selected, together with how the masking is performed (removing the masked elements or replacing them by a mask value), the input can be masked. After masking the appropriate items, the masked input is fed to the model.

For the *mask value*, the choice stands between a special value to symbolize the mask, or a valid input value that overwrites the masked information. In visual input, these two options refer to replacing the masked areas with a non-valid pixel value (for example -1, as seen in [40]), or with a value that could normally be encountered in the input, such as the mean pixel value [41], [42], or 0 [43]. Choosing the shape and behaviour of the mask is equally important. Many studies split the input image into equal-sized rectangular patches, and randomly select a number of patches to mask, which stays constant during the entire training [8], [41]. Other papers maintain the masked patches strategy, but may increase the masking ratio as training progresses [42].

2.2 Related Work

The following subsections build on the theory that was discussed above. Each subsection covers recent work that directly pertains to this project.

2.2.1 Used Pre-Trained Models

SatCLIP

A foundation model trained using a CLIP objective was used in this project. SatCLIP, a model proposed in [14], has two components: a vision encoder and a location encoder. It was trained using a CLIP-inspired objective to match pairs of coordinates and multispectral images from the Sentinel-2 satellite. Multiple models were trained in the SatCLIP study, using different vision and location encoder backbones. For the vision encoder, ResNet18, ResNet50 [44], and ViT16 [45] are used. For the location encoders, the distinction is made by the number of Legendre polynomials, which control the spatial smoothness of the location's representation. Section 3.2 elaborates on which model was selected for this study, and why.

ResNet50

ResNet50 is a widely used model in computer vision. Introduced by [44] in 2016, this architecture can still match the performance of newer models in some cases, such as [14]. It is, therefore, still commonly used as a vision encoder backbone, as part of a task-specific model. The main contribution of the ResNet architecture is tackling the problem of vanishing gradients, which were a common occurrence in deep architectures. By adding the input of a layer to the output of the same layer, before the activation function, the layer is trained to learn the *residual function* with respect to the layer's input. The ResNet family of architectures re-defined state-of-the-art performance on a plethora of vision tasks [44].

DeepLabV3

Introduced by [46], DeepLabV3 is a semantic segmentation architecture comprised of a vision encoder, an atrous convolution (dilated convolution) component, and a CNN-based decoder. The atrous convolution is used to extract the finer details from the vision encoder's output. The decoder uses the low-level features from the encoder, in order to process general spatial information, together with the output from the atrous convolution.

Of these components, the atrous convolution is what sets this architecture apart. It requires using wider convolutional kernels, and inserting zeroes into them such that only a subset of the input pixels, which are not adjacent to each other, are used in the convolution. The number of zeroes in the convolutional kernel is called *rate*, whereby normal convolution would use $r = 1$ and $r = 2$ means that every other pixel is involved.

This way, the network's receptive field can be widened and therefore cover more spatial context. Moreover, convolving every other pixel means that stacking multiple of these layers results in acquiring a more densely populated feature map [47]. In the case of DeepLabV3, a variant of atrous convolution called "atrous spatial pyramid pooling" (ASPP) is used instead. This consists of convolving the input using multiple kernels of different sizes and rates, then performing image pooling on the output of all these convolutions.

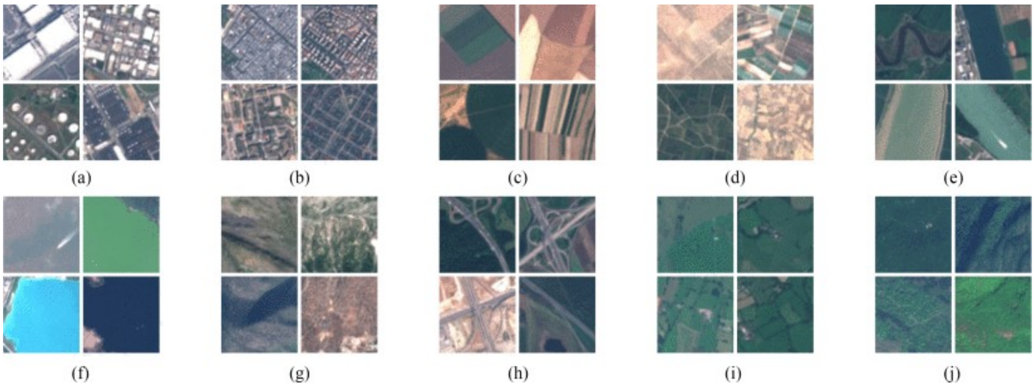


Figure 2.4: Examples 64 × 64 pixel images from EuroSAT dataset, showing the 10 classes: (a) Industrial buildings. (b) Residential buildings. (c) Annual crop. (d) Permanent crop. (e) River. (f) Sea and lake. (g) Herbaceous vegetation. (h) Highway. (i) Pasture. (j) Forest.

The decoder's output is an image of size ($num\_classes \times image\_size \times image\_size$). Each image channel is associated with one class, and contains per-pixel logits representing the probability of that pixel belonging to that class. The output logits do not sum to 1, so applying an additional softmax layer is required.

In this study, a DeepLabV3 model was employed. It was pre-trained on multispectral satellite imagery using the DFC2020 dataset [48], which is a subset of the SEN12MS dataset [49] used in this study. The model was implemented and trained by Lukas Liebel, all related resources are available at this GitHub repository.

2.2.2 Benchmark Datasets and Downstream Tasks

The main contribution of this project is introducing a fine-tuning method to guide the representations of the incomplete images, in order to closely resemble the embeddings of their complete counterparts. The efficacy of this method can be evaluated through the loss values, as well as further testing on downstream tasks. The state-of-the-art benchmarks for these downstream tasks and associated datasets must be discussed, as they represent the performance targets for the training method that is introduced.

EuroSAT

EuroSAT is an image classification dataset comprised of 27000 labeled and geo-referenced multi-spectral images. The Sentinel-2 satellite is the source of these images, and therefore this dataset contains 13 spectral bands. The names, spatial resolutions, and central wavelengths of these bands can be inspected in Table A.1, sourced from the EuroSAT dataset paper [4]. State-of-the-art classification accuracy on this dataset is 99.17% [50], which was achieved using Wide Residual Networks on RGB data. Using all 13 spectral bands, the state-of-the-art accuracy was achieved by [51] with 98.78%, using a Convolutional Neural Network.

An overview of the classes in EuroSAT can be observed in Figure 2.4. Moreover, an example of sub-optimal images can be found in Figure 2.5, which shows sampled images that exhibit a color shift due to atmospheric images.

SEN12MS

SEN12MS is a multispectral dataset introduced by [49]. It contains patches of Sentinel-1, Sentinel-2, Sentinel-2 cloudy, and Land Cover labels based on the MODIS system [52]. The

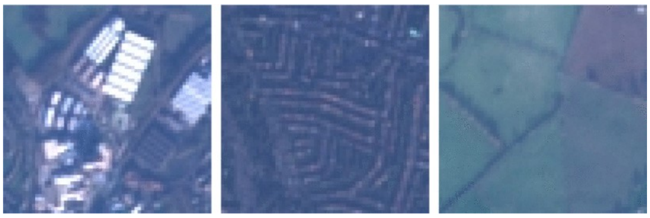


Figure 2.5: Images sampled from EuroSAT that were affected by atmospheric conditions and show a color tint.

name of this system stands for moderate resolution imaging spectroradiometer. These patches are geo-referenced to showcase the same location across all data modalities.

The dataset is structured in nested folders, using the following criteria: first by meteorological season (annotated as *seed value*), then by region of interest (annotated as *scenes*), and finally each scene is divided into several patches.

Due to computational limitations, this study only makes use of the spring season (seed 1158), which is comprised of data captured between 1 March - 30 May 2017. Spring was selected to keep a balanced level of snow and ice occurrence. They would not be a common characteristic of the images, but would appear more than in the summer split. Furthermore, since this study only uses Sentinel-2 multispectral data, and is not concerned with the cloud removal task, only the "s2" and "lc" data modalities are used. This subset of SEN12MS contains a total of 40883 pairs of Sentinel-2 13-band 256 × 256 images and MODIS land cover 256 × 256 labels.

The MODIS images contain four channels, depicting four different land use-land cover classification schemes: IGBP, LCCS land cover, LCCS land use, LCCS surface hydrology. Only the IGBP (International Geosphere-Biosphere Programme) [53] labels are used in this study, because the DeepLabV3 pre-trained model that was used for the image segmentation task was solely trained using this labeling scheme. Moreover, this pre-trained model was part of the IEEE GRSS Data Fusion Contest 2020 [48], which made use of a *simplified version* of the IGBP labels, reducing the number of classes from 17 to 10. An overview of the complete and simplified classes can be observed in Table A.2. It should be mentioned that, while the Sentinel-2 data involves a spatial resolution of 10-60m (depending on the spectral band - see Table A.1), the MODIS labels only have a spatial resolution of 500m. Aside from the low resolution of the labels, another aspect to keep in mind is that the IGBP label itself is imperfect; its accuracy is estimated to be about 67%. [52].

Chapter 3

Method

The following sections describe the methodology used in this project: Section 3.1 describes the used datasets, together with any pre-processing that was performed, as well as the band masking procedure. Section 3.2 then elaborates on the employed models and the connection to the task they were used for. Lastly, Section 3.3 dives into the tasks that the models were deployed on, and explains the performance metrics used to measure and interpret each task.

3.1 Data

The data used in this research consists of multispectral images sourced from the Sentinel-2 satellite. Multispectral images contain more frequency channels than the "standard" RGB, by including frequency bands outside of the visible light spectrum. Table A.1 in the Appendix shows the names, spatial resolutions, and wavelengths of all the spectral bands captured by the Sentinel-2 satellite.

The datasets were processed using the same series of augmentations, ensuring the data distribution matches the one from the pre-training phase of the SatCLIP encoder. Thus, the encoder would not need to adapt to a new data distribution. Instead, the fine-tuning enables the encoder to adapt to the decreased amount of information present within the masked images. After applying the augmentations to the images, every batch is masked as described in Section 3.1.2, then fed to the model.

3.1.1 Pre-Processing and Augmentations

The **EuroSAT dataset** was used for the baseline and scene classification experiments. The images in this dataset are 64×64 pixels, and it contains all 13 spectral bands that the Sentinel-2 satellite captures. To match the input dimension required by the used encoder (256×256 pixels), the dataset was upscaled by a factor of 4, using bilinear interpolation.

The same image augmentations as in the original SatCLIP project are used. Random horizontal and vertical flipping are applied, followed by a Gaussian blur with kernel size $= 3$. For the image segmentation experiments, only the Gaussian Blur was used.

3.1.2 Image Masking

The core idea of this project is to create a pipeline which can adapt to incomplete information, emulating a sensor malfunction. The desired outcome is for the proposed method to be applicable in any sensor fusion task, and not limited to multispectral imagery. The Sentinel-2 satellite uses a single multispectral imager to capture all the spectral bands [54]. However, seeing as the spectral bands capture different information, the decision was made to treat each spectral

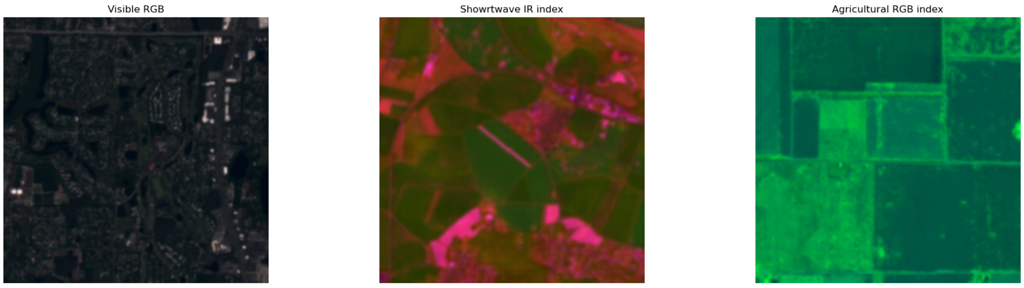


Figure 3.1: Visible RGB, Shortwave Infrared, Agricultural RGB Indexes Visualized.

band as a separate sensor. A sensor malfunction would mean that the information available in one or multiple of these bands is not available to the model. Thus, some or all pixels in the selected bands are masked, by overwriting them with a mask value.

The masking is performed in the dataloader, during the experiment run and not as a preprocessing step. This means that the same images could be masked in different ways on different epochs or experiment runs. The following settings determined the specifics of the masking procedure at each experiment run:

Selecting Masked Bands

The selection of masked bands was influenced by the downstream tasks performed in this project. Seeing as the majority of the classes from both downstream experiments are agricultural categories, multiple sets of bands that are commonly used in combination to perform agricultural tasks were selected. Figure 3.1 shows visualizations of the RGB, SWIR and Agricultural RGB indexes used in this study. As explained in Section 2.1.1, the Agricultural RGB Index is widely used for monitoring plant health. It is comprised of Sentinel-2's bands B02 (blue), B08 (NIR), and B12 (SWIR-1).

Figures B.7, B.8, B.9, B.10, B.11 in the Appendix visualize all sets of masked bands used in this study. Experiments were ran by masking combinations of one, two, and all three of these bands. Aside from the Agricultural RGB index, bands that make up the NDVI index, another commonly used index, were also included in the masked bands of the experiments: B4 (visible red) and B8 (NIR). In the experiments involving more than three masked bands, mixtures between these indexes and the visible light bands were selected for masking. The B4, B8, B9, B11 and the B4, B8, B9, B13 combinations contain bands that are found within the Agricultural RGB index (B2, B8, B12), the Short-Wave Infrared Index (B4, B9, B13), and the NDVI (B4, B8). Lastly, the B8, B9, B10, B11, B12 combination is meant to mask a large portion of the VNIR and SWIR bands. This severely limits access to the information found in these spectral bands, all relevant for agriculture [23].

Masking Schedule & Ratio

The masking ratio is a value between 0 and 1, and represents the coefficient of total pixels that should be masked in each band. The masking schedule refers to how masking progresses throughout training. Three options were available: constant, gradual, and staircase. Constant masking entails masking the same amount of pixels regardless of the training epoch. Gradual masking refers to increasing the masking ratio at each new epoch. Staircase masking is a mix of the previous two, where the masking ratio is increased every few epochs, and remains steady until the next increase. Upon experimenting with the three options, it was decided to only use constant masking in the main study. The other masking schedules achieved comparable results, but the timeframe of this study did not allow for an extensive comparison between these masking schedules across all the experiments. The Appendix contains a brief analysis of the masking schedules, which can be seen in Figure B.2 and Table A.5.

Mask Value

Lastly, the mask value refers to the pixel value used to overwrite the masked pixels. Multiple options were researched: -1, 0, the mean pixel value across the entire dataset, and the mean-per-band pixel value across the dataset. When the mean-per-band was used, pixels were masked with the mean value specific to the band that is being masked e.g. pixels in band 2 would be masked with band 2's mean pixel value across the dataset.

Preliminary experiment runs with these mask values revealed the mean-across-dataset mask value to perform best. Using -1 as the mask value, the model did not converge. If 0 was used, the model would begin to converge but get stuck in a local minimum, compared to using one of the mean values. In the final results provided in this study, the mean-across-dataset value was used due to slightly superior performance compared to the mean-per-band.

3.2 Model Architecture

3.2.1 Encoder

SatCLIP

As encoder, the SatCLIP pre-trained model with a ResNet50 vision backbone, made available by [14], was used. The location encoder within the pre-trained SatCLIP was not used in this study, as it was not relevant for the performed experiments. The decision to use this encoder was made for two reasons: it was pre-trained on multispectral data, and the latent space was formed using a contrastive loss, therefore continuing to use a contrastive loss should not cause major modifications to the structure of the latent space. The ResNet50 backbone with $L = 40$ Legendre polynomials was selected from the SatCLIP pre-trained models. This choice was made due to the slight edge over the other models in the original experiments, and also due to the smaller number of parameters within the vision encoder (compared to the Vision Transformer backbone alternative).

The SatCLIP model was pre-trained on the S2-100k dataset, also curated by the same authors as part of the same study [14]. This dataset contains 100000 Sentinel-2 image tiles of $13 \times 256 \times 256$ pixels, together with location annotations. The datapoints were ”nearly uniformly distributed across global lands mass”. The authors argue this is a big improvement over previous multispectral satellite imagery datasets, which tend to under-represent non-western geographical areas.

SatCLIP [14] uses the CLIP objective to create a shared multimodal latent space between the image and location embeddings. Both encoders found within SatCLIP can be used for multiple downstream tasks, which is shown in the original study by outperforming similar models on the majority of experiments. The latent space of SatCLIP was also inspected via Principal Component Analysis, exhibiting a well-structured division between different principal components, which can be associated with different biomes.

3.2.2 Decoders

MLP: Scene Classification Task

For the scene classification task, multi-layer perceptron (MLP) with a single hidden layer of 64 neurons was used. The input size is 256, in order to match the received embedding from the

SatCLIP encoder. The output layer has one neuron per class, resulting in 10 neurons in total to match the EuroSAT dataset. All layers use ReLU activation [55]. The MLP was trained from scratch, with random initialization, using an Adam optimizer [56], for 20 epochs.

DeepLabV3: Land Use & Land Cover

The image segmentation task was carried out by using a pre-trained DeepLabV3 implementation. As the model had already been trained on Sentinel-2 images for this task, it is the perfect candidate to measure the effect of the proposed encoder fine-tuning method. It removes the need to first train the decoder from scratch on this type of data or downstream task.

The pre-trained model involved a ResNet-101 architecture as an encoder, which resulted in a slight mismatch in input dimensions compared to the ResNet50 encoder used in this study. In the DeepLabV3 pipeline, this ResNet101 provides two outputs. The first one is the output of the first Bottleneck block of the architecture, and is a low-level features representation of dimension ($batch\_size \times 256 \times 64 \times 64$). The second output consists of a high-level feature representation, and is the architecture's output right before the final fully-connected layer. The shape of this output is ($batch\_size \times 2048 \times 16 \times 16$).

In order to swap out the encoder from the checkpoint with the SatCLIP fine-tuned encoder, the intermediate features extracted from the encoder needed to have the shape required by the rest of the DeepLabV3 architecture. The low-level features of both ResNet50 and ResNet101 have the same shape, but the high-level features of ResNet50 are of shape ($batch\_size \times 2048 \times 8 \times 8$). As such, bilinear interpolation was used to scale the output by a factor of 2 before feeding it to the DeepLabV3 decoder.

Despite having trained their model on the simplified IGBP labels (see Table A.2, the pre-trained model outputs 21 classes in order to match the full MODIS label system. Due to using the simplified IGBP labeling in this study, the final convolutional layer of the pre-trained DeepLabV3 model was also modified to output 10 channel images instead of 21.

3.3 Learning Objective

The conducted experiments can be split into the baseline and downstream experiments. The learning objectives used throughout these runs are described in the following paragraphs.

3.3.1 Baseline Experiment

The baseline experiments consisted of a self-supervised learning task, in the form of aligning the latent representations of the incomplete and complete images using a contrastive loss. The main purpose of these experiments was to observe whether the model is able to converge on the task of encoding incomplete images. Subsequently, the encoders trained in these experiments were frozen and re-used in the downstream tasks.

A contrastive loss was used to align the representations of the masked images to those of the complete images. To this extent, two separate instances of the pre-trained vision encoder from SatCLIP were used. One model was fully frozen, and provided the latent space representations of the complete images. The other encoder was fine-tuned using the contrastive loss. The trainable encoder was fed the incomplete images, and contrastive learning was used to match each incomplete image's embedding to the embedding of the corresponding complete image from the frozen encoder. Thus, the trainable encoder could learn to output latent embeddings of incomplete images that correspond to the corresponding embeddings of the complete images, and differ from the embeddings of the other complete images in each batch. Figure 3.2 visualizes this training objective.

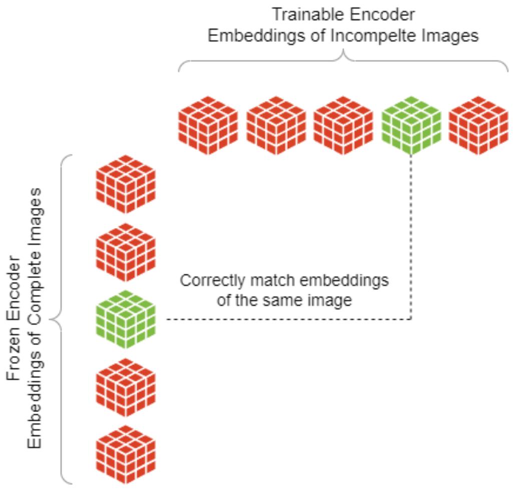

Trainable Encoder
Embeddings of Incompelte Images
Frozen Encoder
Embeddings of Complete Images
Correctly match embeddings
of the same image

Figure 3.2: Visualization of proposed encoder fine-tuning procedure: embeddings of the same image sourced from different encoders must be matched.

The core idea behind this strategy is to facilitate learning to create embeddings of incomplete images that are as similar as possible to those of complete images. Aligning the incomplete and complete images' embeddings would mean that the encoder is capable of adapting to the lack of data. The distance between the two embeddings can be interpreted as a measurement for how much of the removed information can be recovered by the fine-tuned encoder. If this distance is very small, it means that almost all the information that would be stored in the complete image's embedding is still present in the latent representation of the incomplete image. Conversely, a larger distance would mean that the fine-tuned encoder is not as capable of inferring the missing information. If the distance is greater than the distance between the complete image's embedding and a pre-trained encoder's latent embedding of the incomplete image, then the fine-tuning method does not work as intended.

As explained in Section 2.1.2, the CLIP method uses cross-entropy loss on logits obtained from cosine similarities is used as the learning objective. Lastly, the temperature parameter τ within the cosine similarity was kept constant at $\tau = e^{2.8952996730804443}$. This value was obtained during the original SatCLIP study, by training it as a parameter within the loss function [14].

3.3.2 Downstream Tasks

Both the scene classification and the image segmentation experiments involve the cross-entropy loss, since both prediction tasks use given ground truth labels. The implemented cross-entropy loss includes a softmax layer that computes logits from the network's output.

The scene classification experiment is a single-label classification task. The cross-entropy loss was used to train the MLP network during this experiment. Despite the slight class imbalance within EuroSAT, which can be seen in Figure B.1, class weights were not used when computing the cross-entropy loss.

During the LULC experiments, the softmax was computed across the channel dimension to obtain an *argmax* of predictions at the level of each pixel in the image. The 'mean' reduction method was used to compute the final loss value, and once again class weighting was not used. The LULC experiment also computed the *average accuracy* and *mean intersection-over-union* metrics, which can be seen in Equations 3.3 and 3.4. Both metrics make use of a *confusion matrix*, which keeps track of all the model's correct and incorrect outputs by counting true positives (TP), false positives(FP), true negatives(TN), and false negatives(FN) per each class.

$$Average\ Accuracy = \frac{\sum_i^C CM_{ii}}{\sum_i^C \sum_j^C CM_{ij}}$$

Figure 3.3: Average accuracy calculated from a confusion matrix. CM_{ij} is element from row i, column j of the confusion matrix. C is the total number of classes.

$$mIOU = \frac{1}{C}\sum_i^C \frac{TP_i}{TP_i + FP_i + FN_i}$$

Figure 3.4: mIOU formula calculated from a confusion matrix. C is the number of classes. TP_i, FP_i, FN_i represent the number of true positives, false positives, and false negatives for class i.

Chapter 4

Experiments

This chapter provides further experimental setup details, together with the results of the baseline and downstream experiments. A brief description of the results is included, together with some intuition on how they can be interpreted. All experiments were run on either Nvidia RTX 4090 or RTX 3080-Ti GPUs.

4.1 Baseline Experiments

The baseline experiments aim to answer **Research Question 0:** Is a fine-tuned encoder able to converge on the task of aligning its embeddings of incomplete images to the embeddings of the corresponding complete images, sourced from a pre-trained encoder?

The encoder fine-tuning was performed on the EuroSAT dataset for 15 epochs. When multiple bands were masked, the encoder was fine-tuned for 25 epochs instead. The batch size was 64, and $lr = 0.0001$, the default learning rate from the SatCLIP study [14], was used. Longer training runs were also attempted, with and without learning rate schedulers, but overfitting would occur if training for more than 15 epochs in single-band experiments, or 25 epochs in multi-band experiments. Training for 15 epochs took between 15-20 minutes on the aforementioned hardware.

Tables 4.1 and 4.2 feature validation loss values at different epochs of training, for various combinations of masked bands. The models trained during these baseline runs were subsequently used as encoders in the downstream tasks. The best and worst loss values measured on the test split, after 15 epochs of training (and 25 for the multi-band experiments) are highlighted using green and red text.

4.2 Downstream Experiments

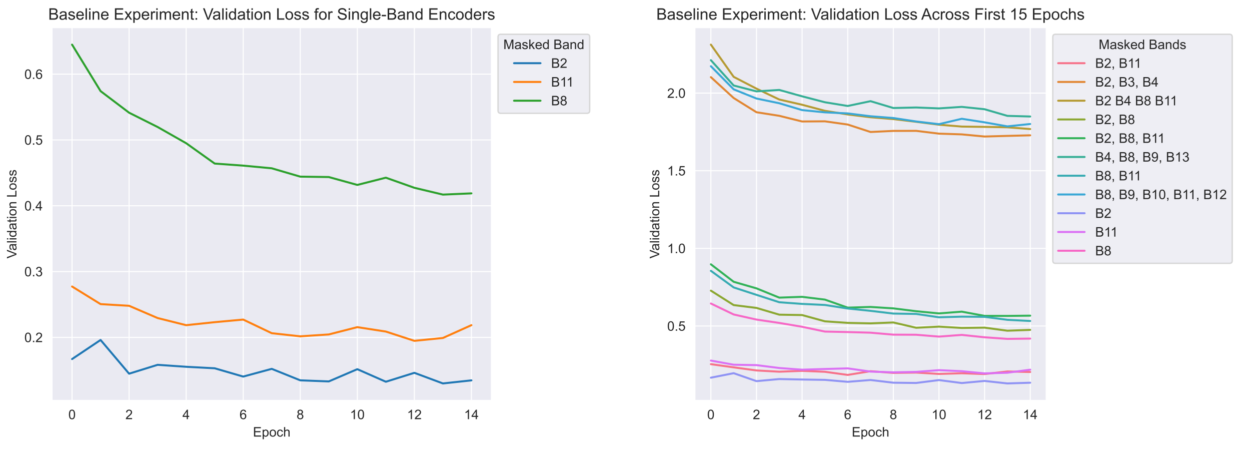
To assess the quality and applicability of the encoders' embeddings of the incomplete images, evaluating the performance on downstream tasks is required. This enables answering **Research**

Masked Band	Val Loss Ep 1 ↓	Val Loss Ep 8 ↓	Val Loss Ep 15 ↓	Test Loss ↓
none	0.065	0.060	0.055	0.057
B2	0.167	0.151	0.134	**0.132**
B8	0.6449	0.4436	0.4107	**0.4125**
B11	0.277	0.201	0.218	0.211

Table 4.1: Baseline experiment runs for single masked bands. Lower values mean better performance, as indicated by the arrows. Test loss was computed after 15 epochs of training.

Masked Band	Val Loss Ep 1 ↓	Val Loss Ep 13 ↓	Val Loss Ep 25 ↓	Test Loss ↓
none	0.065	0.060	0.055	0.057
B2 B11	0.254	0.207	0.186	**0.178**
B2 B8	0.727	0.522	0.475	0.469
B8 B11	0.855	0.577	0.537	0.533
B2 B3 B4	2.103	1.724	1.681	1.651
B2 B8 B11	0.897	0.564	0.530	0.532
B2 B4 B8 B11	2.105	1.784	1.709	1.722
B4 B8 B9 B13	2.212	1.911	1.818	**1.801**
B8 B9 B10 B11 B12	2.024	1.835	1.75	1.75

Table 4.2: Baseline experiment runs for multiple masked bands. Lower values mean better performance, as indicated by the arrows. Test loss was computed after 25 epochs of training.



(a) Single-Band Encoders

(b) All Encoders

Figure 4.1: Baseline experiment: validation loss for each band combination. The first 15 epochs are depicted, in order to visualize the steeper initial decline in all experiment runs.

Masked Band	Pre-Trained Encoder Accuracy ↑	Fine-Tuned Encoder Accuracy ↑
none	96.53 ± 0.32	-
B2	**95.96** ± 0.2	96.26 ± 0.35
B8	94.63 ± 0.15	95.52 ± 1.31
B11	95.6 ± 0.26	96.9 ± 0.1
B2 B11	95.16 ± 0.4	96.46 ± 0.32
B2 B8	94.76 ± 0.2	**95.65** ± 1.37
B8 B11	94.46 ± 0.41	96 ± 0.2
B2 B3 B4	78.3 ± 2.22	95.76 ± 1.44
B2 B8 B11	94.6± 0.004	96.16 ± 0.2
B4 B8 B9 B13	69.44 ± 1.41	95.85 ± 0.63
B2 B4 B8 B11	**67.52** ± 0.21	95.7 ± 0.28
B8 B9 B10 B11 B12	69.86 ± 3.21	**96.91** ± 0.021

Table 4.3: Test Accuracy after 20 epochs of training on the Scene Classification task. Results show mean and SD computed over three runs. Higher values mean better performance.

Questions 1, 2, 3 regarding the performance when only partial information is available. To this extent, the models trained during the baseline experiments were frozen and re-used. Then, the decoder networks described in Section 3 were deployed for each downstream task.

4.2.1 Scene Classification

Out of the two downstream tasks, scene classification is the easier one: only one prediction must be made for an entire image. The results observed in Table 4.3 show the performance of the MLP decoder with and without the fine-tuned encoder over 20 epochs of training, with batch size = 64, and learning rate = 0.01. Training for 20 epochs took roughly 30-35 minutes. The experiment was performed on the EuroSAT dataset. The results of this experiment offer an extra measurement of the encoder's ability to provide a useful input representation of the incomplete image to the task-specific decoder. Besides that, the results of this experiment can offer insight into which combinations of spectral bands play a more crucial role in classifying the scenery. The results are also visualized in Figure 4.2.

4.2.2 Image Segmentation

In the LULC segmentation, having access to many spectral bands can make more of a difference than in the scene classification task. Making the distinction between different regions within the image could require information that is only available in particular spectral bands. If the fine-tuned encoder can recover this information, it could significantly impact task performance.

After the modifications described in Section 3.2.2 were made to the DeepLabV3 architecture, the model was trained for 20 epochs. One such experiment run took approximately one hour on the hardware mentioned in the beginning of this chapter. Initial experimental runs revealed that task performance could further improve by running for an extra 10 epochs for some band combinations. However, the improvements were minimal and consisted of at most a $0.5 - 1.5\%$ increase in the average accuracy metric.

The number of training epochs was set to 20, after preliminary runs with 30 epochs. The DeepLabV3 model was able to converge on this experiment within 20 epochs. In the case of some masked bands, improvements in the evaluated metrics might occur after the 20th epoch. In other combinations of masked bands, the model would sometimes show signs of overfitting

Masked Bands	Pre-Trained Encoder		Fine-Tuned Encoder	
	Avg Acc (%) ↑	mIOU ↑	Avg Acc (%) ↑	mIOU ↑
none	74.17 ± 0.155	0.604 ± 0.002	-	-
B2	72.34 ± 0.695	0.578 ± 0.008	72.47 ± 0.195	0.578 ± 0.005
B8	73.51 ± 0.802	0.593 ± 0.005	71.22 ± 0.135	0.568 ± 0.002
B11	**79.18** ± 6.01	0.591 ± 0.005	72.34 ± 0.725	0.581 ± 0.004
B2 B11	73.33 ± 0.579	0.59 ±0.004	**76.02** ± 5.81	0.574 ± 0
B2 B8	77.38 ± 6.151	0.581 ± 0.005	72.105 ± 0.53	0.572 ± 0.003
B8 B11	77.56 ± 5.74	0.589 ± 0.002	71.31 ± 0.41	0.57 ±0.002
B2 B3 B4	72.87 ± 0.261	0.587 ± 0.004	74.86 ± 6.43	0.56 ±0.002
B2 B8 B11	72.75 ± 0.19	0.586 ± 0.002	71.49 ± 0.014	0.569 ± 0.007
B4 B8 B9 B13	**71.19** ± 0.94	0.561 ± 0.012	69.74 ± 0.537	0.549 ± 0.004
B2 B4 B8 B11	72.09 ± 0.926	0.572 ± 0.002	70.43 ± 0.622	0.56 ±0.004
B8 B9 B10 B11 B12	71.4 ±0.47	0.563 ± 0.003	**68.98** ± 0.273	0.547 ± 0.005

Table 4.4: Test Accuracy and mIOU after 20 epochs of training on the Image Segmentation task. Results show mean and SD computed over three runs. Higher values mean better performance.

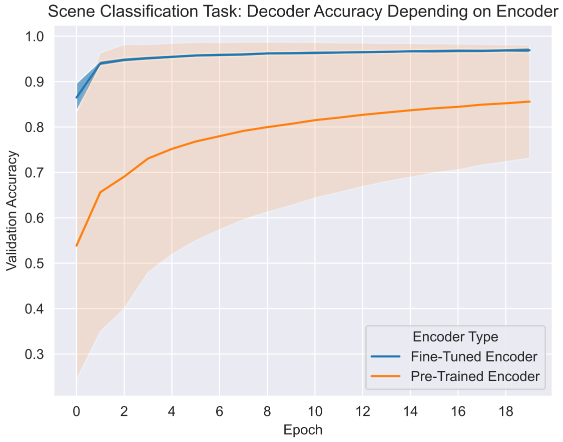


Figure 4.2: Validation accuracy of scene classification experiment, averaged across all experiment runs, using pre-trained and fine-tuned encoders. The shaded region shows one standard deviation above/below the mean.

past the 20 epoch. For this reason, together with the advantage of shortened training time, the decision was made to limit training to 20 epochs.

The learning rate was set to $lr = 0.01$, the same value that was used during the pre-training of the DeepLabV3 model. Cross-entropy loss was used as the training objective, as described in Section 3.3.2. To measure model performance, the average accuracy (AA) and mean intersection over union (mIOU) were computed. The formula of these metrics can be found in Equation 3.3 and 3.4. The performance results are displayed in Table 4.4. The results of this experiment are also visualized in Figure 4.3 (Average Accuracy) and Figure 4.4 (mIOU). The Appendix contains a number of visualized model outputs: Figures B.3, B.4, B.5, B.6.

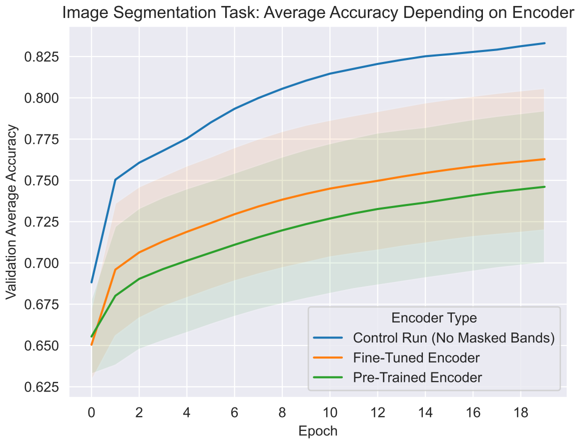


Figure 4.3: Validation average accuracy for land use-land cover experiment, averaged across all experiment runs, using pre-trained and fine-tuned encoders. A control run with pre-trained encoder and no masked bands is also depicted. The shaded region shows one standard deviation above/below the mean.

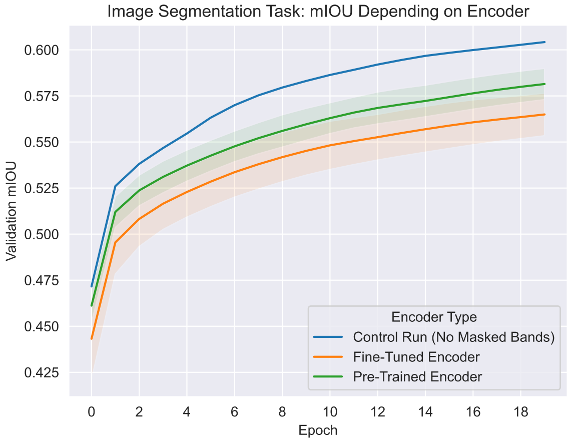


Figure 4.4: Validation mIOU values for land use-land cover experiment, averaged across all experiment runs, using pre-trained and fine-tuned encoders. A control run with pre-trained encoder and no masked bands is also depicted. The shaded region shows one standard deviation above/below the mean.

Chapter 5

Discussion

This chapter begins by interpreting the results from the experiments. First, different band masking settings are compared in order to determine the relationship across different runs of the same experiment. Then, each proposed research question is discussed by analyzing the experimental results. Lastly, limitations of this project are discussed, as well as potential future extensions.

5.1 Analysis of Results

5.1.1 Comparison between different sets of masked bands

Before interpreting results of experiment runs with different masked bands, it is important to take into consideration how the selected set of masked bands can affect the outcome. The following paragraphs discuss the impact of removing certain spectral bands in the context of each experiment.

Baseline Experiments

The loss values in the initial epoch can offer an approximate measurement of the extent to which each set of masked bands influences the output embedding. At this stage, the trainable parameters have gone through a single epoch of training. The absence of the masked bands strongly influences the loss values, since model convergence has not yet been reached. The loss values represent the distance between the complete and incomplete embeddings within the latent space. The *control run* of the baseline experiment, where no bands are masked, serves the purpose of contextualizing this measurement and the degree of stochasticity present in the results.

As Table 4.2 shows, the lowest loss values are reached, across all epochs, during the experiment with B2 (visible blue) masked. This suggests removing this band generates a small dissimilarity between the embeddings of the complete and corresponding incomplete images. One can interpret this as proof that B2 is not as important as other bands in the latent representation, meaning that it was not as crucial during SatCLIP's initial training regime. Conversely, B8 (Near-Infrared) seems to be the most important one out of the three bands included in the single-band baseline experiments, because masking it results in the largest loss values. Another important aspect to notice is how the loss value progresses through the training epochs. Even though the B8 experiment scores the highest loss values, it also shows the steepest decrease. This would also suggest an ability to adapt to the absence of this band.

The multi-band baseline experiments shown in Table 4.2 provide the same type of insight for combinations of masked bands. In line with the previous conclusion about each band's importance, the experiment where the less influential bands are missing (B2 and B11) reach the lowest loss values. A surprising result is that masking B8 and B11 results in almost identical loss values and convergence as masking B2, B8 and B11. This would point to either B2 having little to no influence over the latent representation, or to the information found within B2 being inferable from other spectral bands. This can also be noticed from Figure 4.1b, which noticeably groups the experiment runs in three clusters. The bottom cluster contains B2, B2 B11, and B11. The middle cluster contains B8, B2 B8 B11, B2 B8, and B8 B11.

The visible light bands (B2 B3 B4) appear to be important based on the loss values, seeing as the other experiment run with three masked bands scores much lower loss values. Masking B4, B8, B9, B13 results in the most difficult embedding alignment task: this experiment run scores the worst loss values across all stages of training.

Downstream Tasks

As previously established, removing band B2, and to a lesser extent B11, does not impact the embeddings as much as removing band B8. Consequently, masking these bands also results in the best performance in the downstream tasks. As seen in Tables 4.3, 4.4, both in the single- and multi-band experiments, the runs that achieve the best accuracy are those where band B2, and bands B2 and B11 for multi-band, are masked. The conclusions drawn from the baseline experiments about each band's importance are in-line with the conclusions drawn in the downstream tasks, which confirms the provided interpretation is not experiment-dependent.

Once again, it is important to note that the performance on the other experiment runs is not far from the best performance. This shows that enough information to achieve good results is still present within the image, regardless of the selection of masked bands. Similarly to the baseline experiments, the performance starts off worse (1-2% less accurate at epoch 1) when masking bands that appear to have more impact over performance. However, it does converge towards a value that is comparable to the best performing runs (within 0.5% accuracy across all runs).

As expected, the experiments where many bands are removed do not perform as well as the experiments where more information is available, but the difference in accuracy is small. Despite removing more information, most multi-band experiments converge to accuracy values comparable to the single-band experiments. In the well-performing multi-band experiments, both the test accuracy and the validation accuracy at different training stages show accuracy values within 0.5% of the best performing runs.

Two band combinations where performance is significantly worse when not using the fine-tuned encoder are the B2,B4,B8,B11 and the B8, B9, B10, B11, B12. In contrast to all other runs, which reach an accuracy above 90%, these two experiment runs score approximately 67% accuracy while using the pre-trained encoder. This shows that there is information relevant for the scene classification task within these bands.

5.1.2 Answering the research questions

Research Question 0

The main takeaway from the baseline experiment is that adapting to missing information is possible. All baseline experiments show a convergent trend of the loss value, which can be interpreted as the model being capable of (partly) adapting to a lack of information in all the bands included in the experiments. The fine-tuned encoder can converge on the task of aligning its embeddings to the embeddings of the complete images.

Research Question 1

This research question can be answered by comparing corresponding rows from the pre-trained and fine-tuned encoder, in both downstream experiments. In the scene classification experiment (Table 4.3), it can be noticed that the accuracy values are higher when using the fine-tuned encoder. Comparing Tables A.3 and A.4 from the appendix shows the fine-tuned encoder enables scoring higher accuracy values both before and after training the decoder. The single-band experiments only showing a slight difference between the encoders, of $0.3 - 0.5\%$ accuracy, but there is a significantly larger difference in the multi-band experiments. When masking two bands, the test accuracy in the experiments with the fine-tuned encoder are $1.1 - 1.6\%$ higher. If three or more bands are masked, the fine-tuned encoder manages to boost the accuracy levels by $3 - 6\%$. Figure 4.2 shows that the spread (standard deviation) of the results using the two encoders are very different, and proves that the fine-tuned encoder offers a robust method of adapting to missing information regardless of which bands are masked.

Whilst the experiments that do not use the fine-tuned encoder still obtain high accuracy levels of above 90%, it is important to keep in mind that the ResNet50 architecture is a well-performing image encoder for classification tasks [57]. It still has access to the majority of the spectral bands in their entirety, thus it does not come as a surprise that the produced embeddings still contain enough information to differentiate between the classes. Moreover, the EuroSAT dataset is not considered a difficult classification task, because the classes are quite distinctive. [58].

The accuracy scores from the first epoch can also be used to compare the two encoders. While the fine-tuned encoder allows the decoder to consistently score above 93% regardless of masked band, the first epoch's accuracy scores using the pre-trained encoder are below 90%, with the values being as low as 23% when bands B2, B4, B8, B11 are masked. As seen in row 11 of Table 4.3, this set of masked bands also scores the largest difference between using the fine-tuned and the pre-trained encoder. By the end of training, the test accuracy increases by 28% through the use of the fine-tuned encoder in this experiment run.

In the land use & land cover segmentation experiment, the effect of the fine-tuned encoder is not as visible. In fact, the results of this experiment are inconclusive, seeing as neither the pre-trained nor the fine-tuned encoder consistently outperforms the other. Out of 11 different band masking combinations, the pre-trained encoder outperforms the fine-tuned encoder in 8. Regardless of which encoder performs better, the difference in accuracy is at most 6.8%, with mean 2.93% and median 2.29%. The number of masked bands also does not appear to determine which encoder performs best.

One possible explanation for the inconclusive results of the image segmentation experiments could come from how the encoders were fine-tuned. The embedding alignment was performed using the final layer's output, which is the input to the MLP decoder of the scene classification task. However, the image segmentation decoder does not use this as input, but two intermediate layer outputs of the network as explained in Section 3.2.2. It is possible that these intermediate features were not as strongly affected by the proposed fine-tuning method as the final layer's output was. This could mean that the fine-tuned encoder would not be able to recover as much of the missing information within these intermediate outputs.

Another interpretation of the image segmentation results could be that this task requires too much information from the input to be able to mask some of it. It could be the case that the fine-tuned encoder *does* recover some missing information, but not enough to fulfill the requirements of the decoder and make a difference in the tracked performance metrics.

The fact that some of the scene classification experiment runs start off with a poor accuracy under the pre-trained encoder, but shrink the gap to the fine-tuned encoder runs, also requires further explanation. One possible explanation is that the fine-tuned encoder readily provides enough information to make the distinction between the classes, more so than the pre-trained encoder. While using the latter, the *decoder* must instead adapt to the missing information within the input, as provided by the pre-trained encoder's embeddings. Between having the fine-tuned encoder adapt to the missing information via the proposed method, and having the downstream task decoder adapt to the missing information that is encoded by a pre-trained encoder, there is a performance advantage in using the proposed fine-tuning method.

Research Question 2

The answer to this research question can be extracted by comparing the experiment runs that use the fine-tuned encoder to the *control runs*, where a decoder is trained on complete input using embeddings sourced from the pre-trained encoder. In the scene classification experiment, masking 1-2 bands gives inconclusive results compared to the control group: some experiment runs, such as masking B8 or masking B2 and B11, score lower accuracy values, but are still within 0.5% of the control group. The biggest performance difference is on the runs that mask B2, B4, B8, and B11: there is a 1% lower accuracy than the control group. Other runs, such as masking B11 or the visible light bands (B2,B3,B4), end up surpassing the control run performance by as much as 0.5%. Given that the state-of-the-art performance on the EuroSAT dataset, 99.17%, was reached using solely the visible light bands [50], this is an unexpected result. The best performance in the scene classification task is at 96.91% with bands B8 B9 B10 B11 B12 masked. This score is $<$ 2% within state-of-the-art accuracy on all 13 spectral bands, and 0.4% higher than the control run of this experiment with no masked bands. It is possible that some of the spectral bands might increase task difficulty, which would also explain why the literature scene classification state-of-the-art is higher on RGB than on multispectral imagery.

All test accuracy scores are within 1% of each other, whether the number of masked bands is 0, 1, or even 5. It is therefore difficult to state whether there is a difference between a decoder's performance on incomplete images, encoded by the fine-tuned encoder, and the decoder's performance on complete images, encoded by the pre-trained encoder. This conclusion contradicts the hypothesized outcome, which was that the pre-trained encoder that uses the entire range of available information would enable scoring the highest accuracy score. As mentioned under the Research Question 1 discussion in this section, a possible explanation of this outcome would be that the intermediate outputs of the network have not been directly trained to infer missing information. Instead, the back-propagation steps of the baseline experiments would have needed to cause this low-level feature alignment.

In the image segmentation experiment, the same inconclusive outcome occurred: when masking some of the bands, a lower accuracy than the control group is reached using the fine-tuned encoder. In other experiment runs, the fine-tuned encoder allows the model to surpass the control run's accuracy. This information would suggest no conclusive answer to RQ2 can be obtained. An unexpected outcome is that the pre-trained encoder also achieves an accuracy above the control run on three of the experiments: B11, B2 B8, B8 B11. The large standard deviation of the accuracy in these experiments might suggest that these are not reliable results. Another possible explanation could be that some of the information within the 13 spectral bands can hurt performance, and removing some of it is beneficial.

Research Question 3

Figures 4.2, 4.4, and 4.3 depict the progression of the recorded metrics. By analyzing these trends, one can identify how the learning curve of a decoder behaves based on the selected encoder and the amount of masked bands. In the scene classification experiment, the learning curve is visibly influenced by the encoder choice. The decoder converges within the first 5 epochs when using the fine-tuned encoder, and scores $>$ 90% accuracy after the first epoch.

Using the pre-trained encoder instead does not prevent the decoder from converging, but it does slow its rate of convergence. After converging, the decoder appears to remain on a lower plateau compared to the performance reached using the fine-tuned encoder.

In the image segmentation tasks, the encoder choice appears to have close to no impact on the convergence rate. In the average accuracy metric, the convergence rate appears to be identical regardless of encoder choice. In the mIOU graph, however, the fine-tuned encoder appears to cause a slightly quicker increase in the metric. In general, the decoders do exhibit very similar learning curves regardless of which encoder was used.

Main Research Question

Addressing the research questions offers an insight into the overall advantages of using the fine-tuned encoder. The experiments that were carried out in this study show that the fine-tuned encoder can positively influence downstream task performance. To a certain extent, information extrapolation under incomplete input conditions is possible using the proposed fine-tuning method. However, the encoder's influence on downstream task performance is largely dependent on the experiment setup.

5.2 Limitations

The potential improvements to this research must be acknowledged, in order to provide an objective context for the aforementioned results and takeaways. This section describes limitations of the presented research.

5.2.1 Proposed Method

Adjusting to the Downstream Task

As explained in Section 5.1.2, the proposed fine-tuning method might have been less effective in the image segmentation experiment due to only aligning the final layer's output. If the intermediate outputs that are fed to the DeepLabV3 decoder had also followed the same alignment procedure, one can hypothesize that the encoder could have also improved the downstream task performance on the image segmentation task.

Generalisability of the Proposed Method

This research was initially aimed at presenting a generalisable method for fine-tuning on incomplete multi-sensor data. However, due to time constraints imposed by the thesis duration, only the SatCLIP model was researched as an encoder, and only multispectral satellite imagery was used. It is therefore not possible to assess how generalisable the proposed method is to other architectures or domains. None of the proposed method's components are particular only to this architecture and domain, and related works presented in Section 2.1.3 suggest similar studies can be carried out using other architectures and training data. In spite of these aspects, it is not possible to claim the method is generalisable solely based on the results presented in this article.

5.2.2 Band Combinations

The choice of selected band combinations is explained in Section 3.1.2, and an analysis of the impact of masking particular bands is presented in Section 5.1.1. However, a more in-depth analysis of the effect of each individual band on the outcome of the experiment could have been provided if more experiments, exploring more sub-sets of the used band combinations, were carried out. To be able to fully explain the effects of masking a certain set of bands, experiment runs with all sub-sets of band combinations should have been performed and discussed.

5.2.3 Dataset Shortcomings

EuroSAT

The dataset used during the baseline experiments to obtain the fine-tuned encoders, as well as during the scene classification task, presents a number of issues that may have affected the results of this research. First, EuroSAT is quite a small dataset, containing $27,000$ images. These images are 64×64 pixels, meaning the input data had to be upscaled by a factor of 4 to match the image dimension required by the employed encoders, 256×256. While bilinear interpolation provides a reasonable solution to upscale the images, using a higher-resolution set of images could have potentially improved the performance of the presented models. The number of epochs for the baseline and scene classification experiments was also influenced by the dataset size, because further increasing the number of training steps resulted in overfitting. A larger dataset could have offered more opportunities for the models to learn to align the latent embeddings of the incomplete images to those of the complete images.

SEN12MS

Dataset size is not an issue in the case of SEN12MS. In fact, only a small subset (roughly 10%) of this dataset was used for the image segmentation task. However, the problematic part of this dataset are the MODIS labels. The spatial resolution of these labels is very low, $500m$. Moreover, as mentioned in 2.2.2, these labels are only 67% accurate.

The DeepLabV3 model that was used has previously been trained on higher-resolution labels, which resulted in the network's output often being more detailed than the ground truth of the SEN12MS dataset. An argument can be made, therefore, that the performance metrics presented in 4.4 are also inaccurate due to being computed using partly inaccurate labels as the absolute truth. As can be seen in Figure B.3, it has happened that the model's output appeared to be closer to the truth than the label.

5.3 Future Work

This section discusses how future research could build on top of the methods and findings presented in this study. It starts from adjustments that would improve the quality of the research, then pans out towards broader ideas to extend this study.

5.3.1 Alternative Hypermarameters

The most accessible improvement is to attempt further hyperparameter tuning. The following hyperparameter settings might bring improvements:

Masking ratio

The baseline experiments were performed using multiple values for the masking ratio, before selecting 0.9. Attempts using the following values were made: $\{0.1, 0.2, 0.5, 0.8, 0.9, 1.0\}$. The values below 0.9, which was ultimately used across all the presented experiments, did not seem to provide a difficult learning task. Within 7 epochs, the model was able to achieve loss values as close to 0 as the *control runs* from the experiments, where no bands are masked.

Using 1.0 as the masking ratio resulted in very low training loss, approximately 10 times smaller than the corresponding validation loss. As an unexpected outcome, the hypothesized explanation was that fully masking the band(s) caused the model to overfit on the training data. For a lack of arguments to back this explanation, as well as a general lack of understanding of this phenomenon, the experiments using $mask\_ratio = 1$ were omitted from this research. Looking further into this aspect would strengthen this study.

Masking schedule

As described in Section 3.1.2, *gradual* and *staircase* masking were also used as part of additional experiments. They entail changing the masking ratio as training progresses, and showed comparable results to the constant masking that was used during the experiments from this paper. Despite ultimately being excluded from the scope of this study, increasing the masking ratio as training progresses has shown potential, as can be seen in Table A.5 and Figure B.2. By starting with constant masking of some bands, and gradually increasing the masking ratio of additional bands, it could be possible to fine-tune an encoder on more missing bands than the experiments from this study.

Training Hyperparameters

While the learning rate for each task was selected by experimenting with multiple values, the other hyperparameters - weight decay, momentum, batch size - have not been fine-tuned. Improved task performance might be reached by fine-tuning these hyperparameters. Furthermore, alternatives to the used optimizers and schedulers were also not investigated.

5.3.2 Improving the Experiments

There are a few points of improvement to both the baseline and downstream experiments, which are covered in the following paragraphs.

Baseline Experiment

The presented experimental setup involves having one positive match for each element of the input batch. Research on contrastive learning objectives include variants of the contrastive loss that use multiple positive pairs per batch. Examples are the NTXent [59] and SimCLR loss [60]. In the context of this research, creating multiple positive pairs could be obtained by masking different band combinations of the same image. This might allow fine-tuning one encoder for multiple combinations of masked bands.

Scene Classification

As previously mentioned, the EuroSAT dataset is considered trivial for scene classification tasks [58]. Instead, a larger dataset with classes that are more difficult to differentiate, could be used. Moreover, using a dataset with a better resolution might be beneficial for both the encoder fine-tuning and the downstream task experiment.

Image Segmentation

The first point of improvement could be to make use of all available labels from the MODIS system. By using the full IGBP classification scheme, there is a potential to uncover weaknesses of the architecture at the level of specific classes. Using the other three classification systems that are included within the MODIS labels of SEN12MS would make an even stronger argument for the performance that the segmentation decoder achieves.

However, a more important improvement in this task would be acquiring higher-resolution labels. Running this experiment with labels of a better quality would remove any doubt over the legitimacy of the results, and allow for a clearer evaluation of the proposed method based on the performance on this task.

Additional Experiment: Input Reconstruction

Another assessment of how much information can be extrapolated by the fine-tuned encoder would be to perform an input reconstruction task. This would provide a visual method to observe which masked elements could be recovered by the encoder, and which could not. Furthermore, this experiment would pave the way towards a faulty sensor detection task.

Assuming the model resulting from this experiment can reliably predict missing values in the input, it could then be re-used to detect faulty sensors. By comparing the image predicted by the model to the image captured by one of the spectral imaging sensors, it would be possible to detect whether a spectral imaging sensor is malfunctioning. While this method would be vulnerable to the error rate of the model, it could prove to be useful in a scenario where the operating condition of the sensors needs to be automatically monitored.

Chapter 6

Conclusion

In this paper, a fine-tuning method has been proposed for creating a vision encoder that can operate on multispectral imagery with some masked spectral bands. A sub-set of the attempted experiments show this task is achievable. Under particular conditions, the proposed method can result in downstream task performance above the performance on complete input.

When the embedding alignment occurs directly on the encoder's output to be used in the downstream task, the method works as intended. However, downstream task performance can also remain unchanged or potentially decrease upon using the method. When an intermediate output that was not directly involved in the encoder fine-tuning process is used, the performance is less robust. Relying on back-propagation to shape an intermediate output did not offer a strong enough embedding alignment procedure. In light of these conclusions, the aspect to keep in mind when extending this research would be to carefully design the representation learning methodology.

Acknowledgements

I would like to thank my supervisors, Inske and Arnoud, for the guidance and patience that you have offered me. Your advice often proved to be crucial in advancing my thesis. The domain knowledge and good-practices that you shared with me allowed my project to shift towards the final version presented in this paper, and it has been a joy to research such an intriguing setting. I also want to thank the rest of the ASIS department at NLR, who have offered me a warm welcome, and who were always keen to share any insight that could be helpful.

To my friends, who were there to celebrate with me every small step forward, and to listen to my complaints and doubts on every misstep along the way. Thank you for making anywhere feel like home, no matter if you were a bike ride, a train ride, or a phone call away from me. I am grateful for having you in my life.

Special thanks goes to my dear friend and artist Bogdan Cașcaval for designing the cover image and abstract page artwork. I sincerely invite anyone who comes across this article and is in need of a graphic designer to reach out to him via email, as he possesses more talent than he allows himself to believe.

Lastly, my deepest gratitude goes to my family. Thank you for the undying love and support. I would not be where I am without you, I would not be who I am without you. The completion of this thesis, the determination and ambition that went into it, I owe to you.

Appendix A

Extra Tables

Band Name	Spatial Resolution	Central Wavelength
	(m)	(nm)
B01 - Aerosols	60	443
B02 - Blue	10	490
B03 - Green	10	560
B04 - Red	10	665
B05 - Red edge 1	20	705
B06 - Red edge 2	20	740
B07 - Red edge 3	20	783
B08 - NIR	10	842
B09 - Red edge 4	20	865
B10 - Water vapor	60	945
B11 - Cirrus	60	1375
B12 - SWIR 1	20	1610
B13 - SWIR 2	20	2190

Table A.1: The 13 spectral bands featured in the EuroSAT and SEN12MS datasets, sourced from the Sentinel 2 satellite. In some literature, B9 is known as B08A, which causes B10-13 to be indexed as B9-12.

	IGBP Class Name	Simplified IGBP	Notes
1	Evergreen needleleaf forests	Forest	All forest types share one category in the simplified IGBP scheme.
2	Evergreen broadleaf forests		
3	Deciduous needleleaf forests		
4	Deciduous broadleaf forests		
5	Mixed forests		
6	Closed Shrublands	Shrublands	
7	Closed Shrublands		
8	Woody savannas	Unused	Classes discarded due to low presence.
9	Savannas		
10	Grasslands	Grasslands	
11	Permanent wetlands	Permanent Wetlands	
12	Croplands	Croplands	No distinction is made between different types of croplands in the simplified IGBP scheme.
14	Cropland/natural vegetation mosaics		
13	Urban and built-up lands	Urban and built-up lands	
15	Snow and ice	Snow and ice	
16	Barren	Barren	
17	Water bodies	Water Bodies	

Table A.2: Overview of complete and simplified IGBP classification scheme

Masked Band	Val Acc Ep 1 ↑	Val Acc Ep 10 ↑	Val Acc Ep 20 ↑	Test Acc ↑
none	0.918± 0.008	0.953± 0.006	0.96± 0.005	0.965± 0.003
B2	94.29	97.02	97.73	96.62
B8	93.6	96.2	96.49	96
B11	94	96.24	97.02	96.11
B2 B11	94.19	96.89	97.28	96.81
B2 B8	93.26	95.91	96.39	96.70
B8 B11	93.2	95.72	96.18	96.07
B2 B3 B4	95.07	96.89	97.41	96.7
B2 B8 B11	93.71	96.18	96.71	96.81
B4 B8 B9 B13	94.05	95.8	96.47	95.14
B2 B4 B8 B11	94.23	96.09	96.43	95.14
B8 B9 B10 B11 B12	94.56	96.55	97.85	97.51

Table A.3: Scene classification accuracy at different training stages, using a fine-tuned encoder.

Masked Band	Val Acc Ep 1 ↑	Val Acc Ep 10 ↑	Val Acc Ep 20 ↑	Test Acc ↑
none	91.82	95.35	96.1	96.51
B2	91.18	95.5	96.15	95.99
B8	91.55	93.91	95.08	94.6
B11	90.92	95.14	95.31	95.59
B2 B11	28.48	62.17	70.95	71.51
B2 B8	89.4	94.18	94.93	94.51
B8 B11	88.6	93.5	94.25	94.48
B2 B8 B11	88.24	93.53	94.36	93.74
B2 B3 B4	34.93	72.6	78.39	77.18
B4 B8 B9 B13	80.15	89.7	91.07	91.51
B2 B4 B8 B11	23.12	57.2	65.45	67.37
B8 B9 B10 B11 B12	83.63	91.58	92.92	92.07

Table A.4: Scene classification accuracy at different training stages, using a non-fine-tuned encoder.

M. Band	Mask Strategy	Val Loss Ep 0	Val Loss Ep 8	Val Loss Ep 14	Test Loss
none	none	0.065	0.060	0.055	0.057
B2	constant, 0.9	0.167	0.151	0.134	0.132
B2	gradual, 0.2-0.8	0.163	0.16	0.132	0.134
B2	staircase, 0.2-0.8	0.193	0.142	0.162	0.155
B8	constant, 0.9	0.6449	0.4436	0.4107	0.4125
B8	gradual, 0.2-0.8	0.449	0.4289	0.4471	0.447
B8	staircase, 0.2-0.8	0.435	0.441	0.418	0.417

Table A.5: Comparison of implemented masking schedules. Mask Strategy column specifies masking schedule, together with mask ratio (initial and final mask ratios are specified in the case of gradual and staircase schedules).

M. Band	Mask Ratio	Val Loss Ep 0	Val Loss Mid-Training	Val Loss Post-Training	Test Loss
B8	1.0	0.365	0.257 (ep8)	0.238	0.245
B8	0.99	0.532	0.366 (ep8)	0.341	0.347
B8	0.6	0.636	0.442 (ep8)	0.408	0.408
B11	0.9	0.277	0.201 (ep8)	0.218	0.211
B8 B11	0.9	0.855	0.577 (ep8)	0.537	0.533
B2 B8	0.9	0.727	0.522 (ep8)	0.475	0.469
B2 B8 B11	0.9	0.905	0.612 (ep8)	0.575	0.577
B2 B8 B11	0.9	0.897	0.564 (ep13)	0.530 (ep25)	0.532
B2 B11	0.9	0.254	0.207 (ep13)	0.186 (ep25)	0.178
B2 B8 B11	1.0	0.571	0.360 (ep13)	0.308 (ep25)	0.323

Table A.6: Additional experiment runs with various mask ratios and numbers of epochs.

Appendix B

Extra Figures

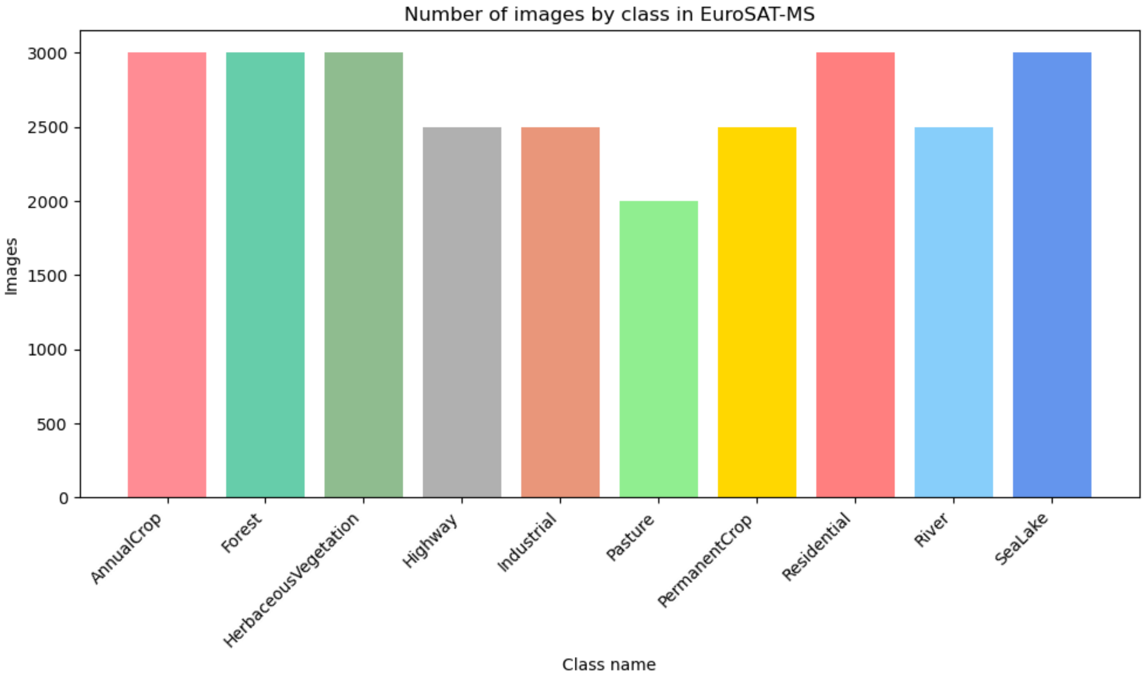


Figure B.1: Number of images per class in EuroSAT-MS dataset.

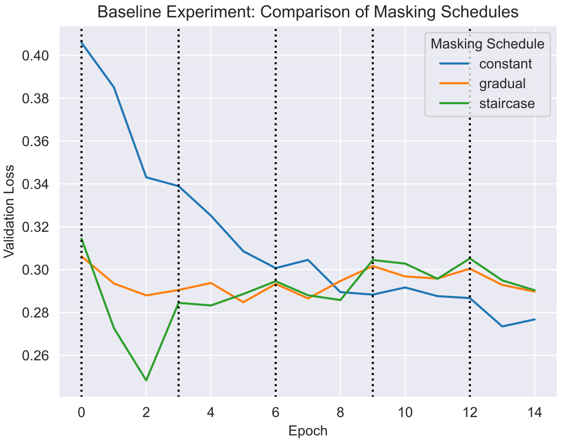


Figure B.2: Comparison of constant, gradual and staircase masking schedules. The vertical lines show when the staircase method increases the mask ratio. Constant masking appears to offer the best result; The other two schedules yield a similar final result despite their different convergence trends. Gradual and staircase masking show promising results, within a small margin of constant masking. This suggests that, using certain hyperparamters, it might be possible to achieve better results than the constant masking.

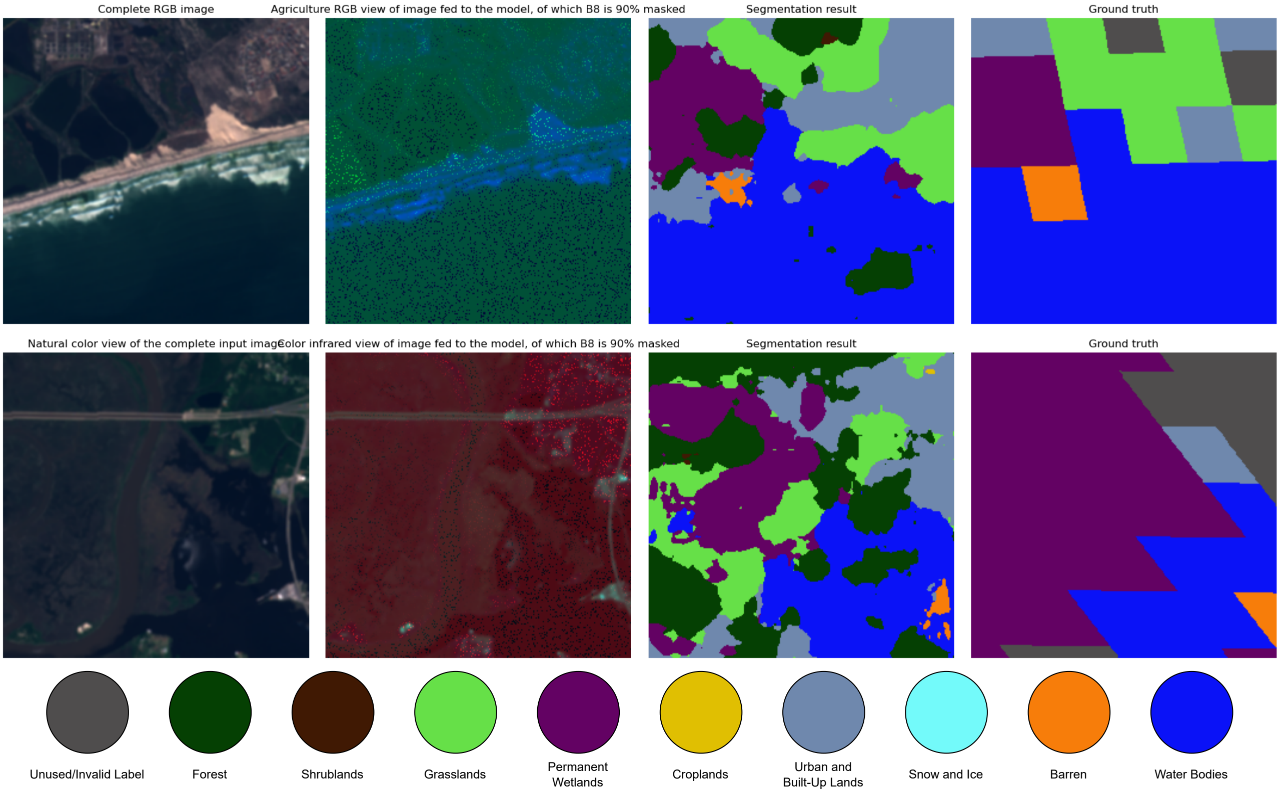


Figure B.3: Two examples where the DeepLabV3 model appears to produce a better output than the provided MODIS ground truth. In the top image, a number of elements occupying less space than one pixel of the MODIS label are present. The barren terrain in the left side of the image is segmented better by the model, and so is the top-right urban settlement. In the bottom image, a similar scenario occurs for barren terrain. Moreover, the shape of the detected wetland is more faithful to reality, which can be seen by the shape of the river in the image.

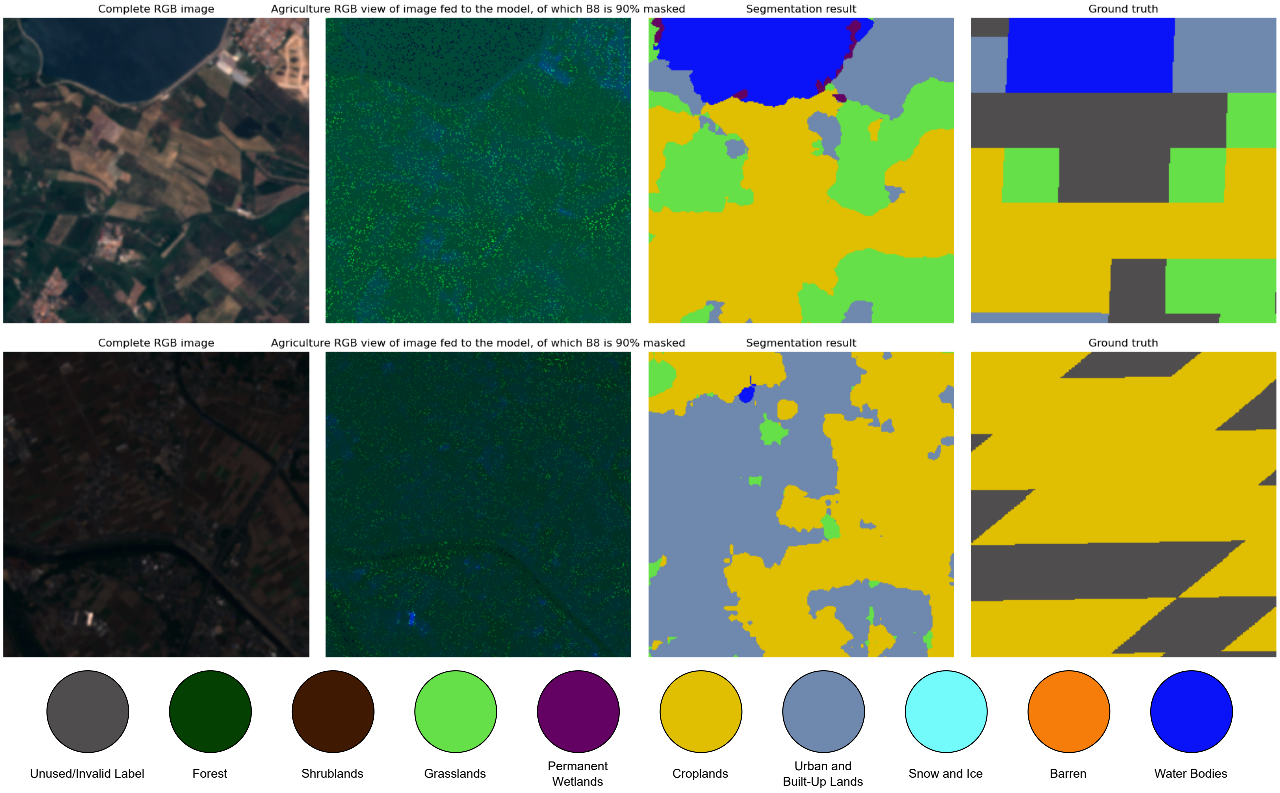


Figure B.4: Example segmentations when B8 is masked before passing the image to the pre-trained (top) and fine-tuned (bottom) encoder. Agricultural RGB view of image included, where the B8 masking adds artifacts that appear as green pixels.

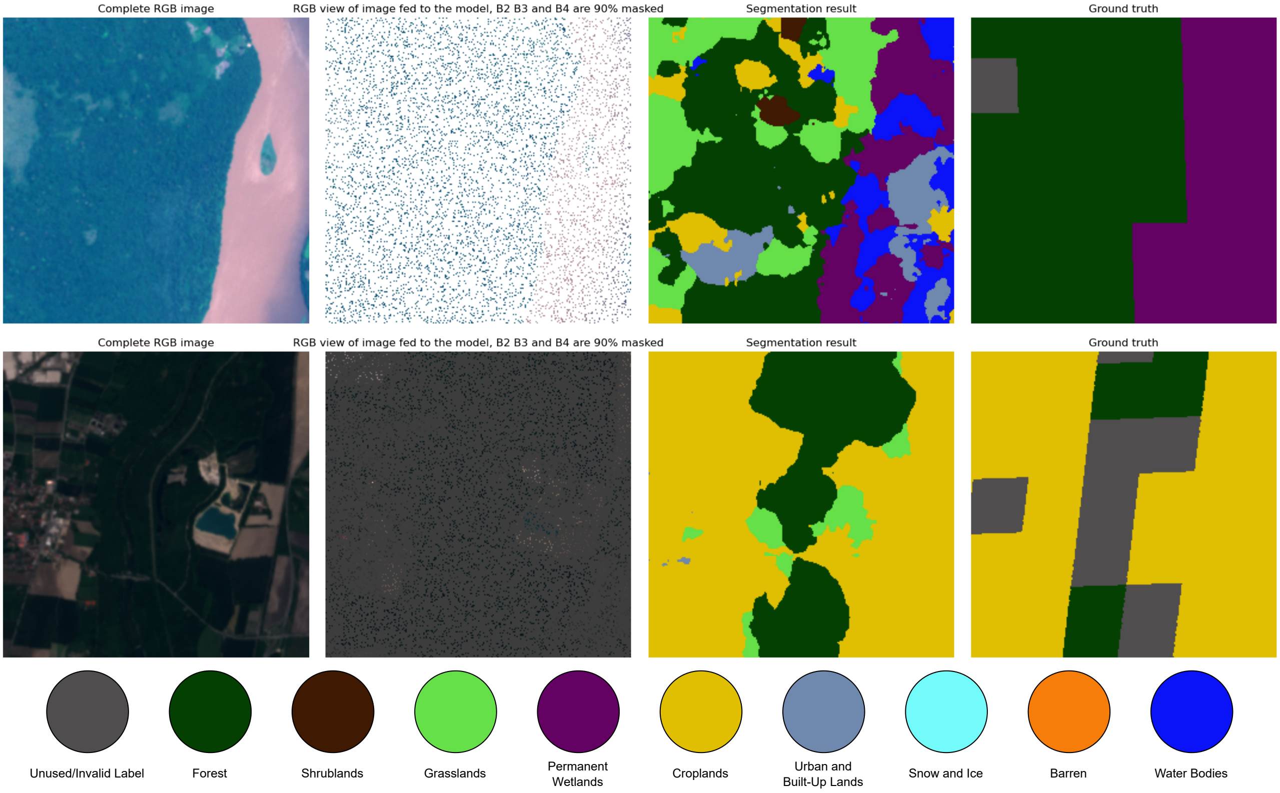


Figure B.5: Example segmentations when B2, B3, B4 are masked before passing the image to the pre-trained (top) and fine-tuned (bottom) encoder. Visible RGB view of image included, where all bands are 90% masked.

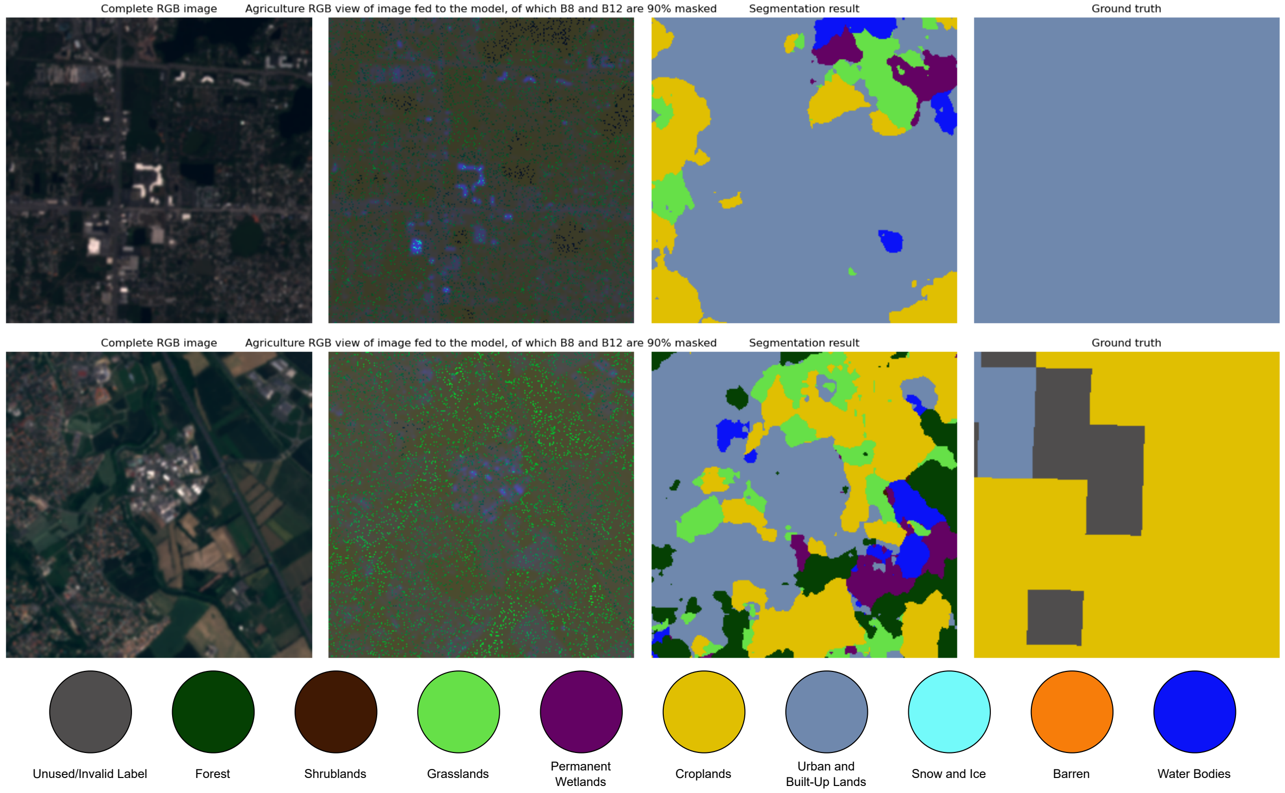


Figure B.6: Example segmentations when bands B8-12 are masked before passing the image to the pre-trained (top) and fine-tuned (bottom) encoder. Agricultural RGB view of image included.

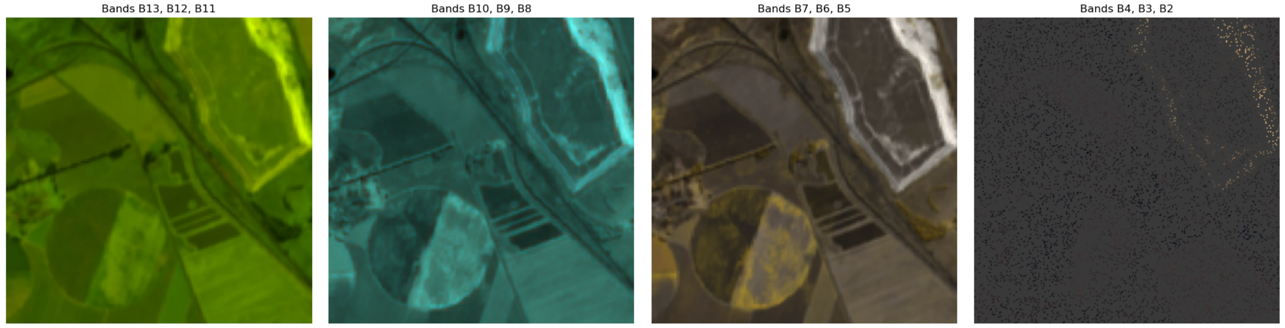


Figure B.7: Visualizing all bands of an image where bands B2, B3, B4 (visible RGB) were masked

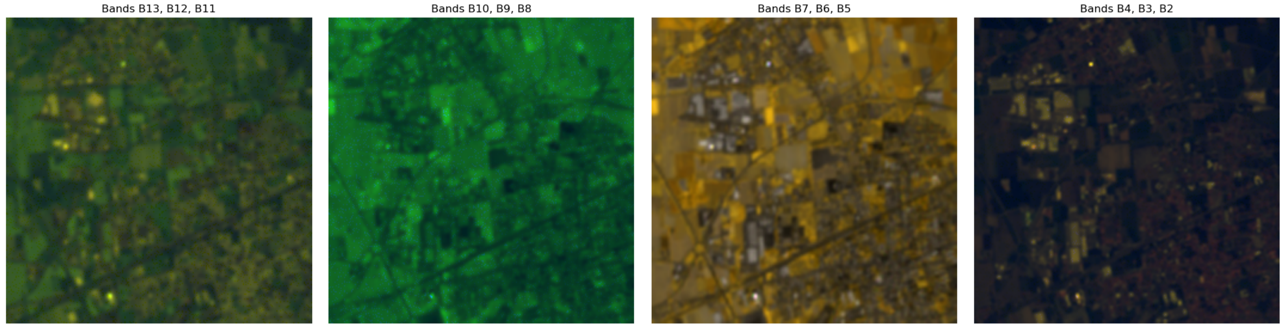


Figure B.8: Visualizing all bands of an image where bands B2, B8, B11 (Agricultural RGB) were masked

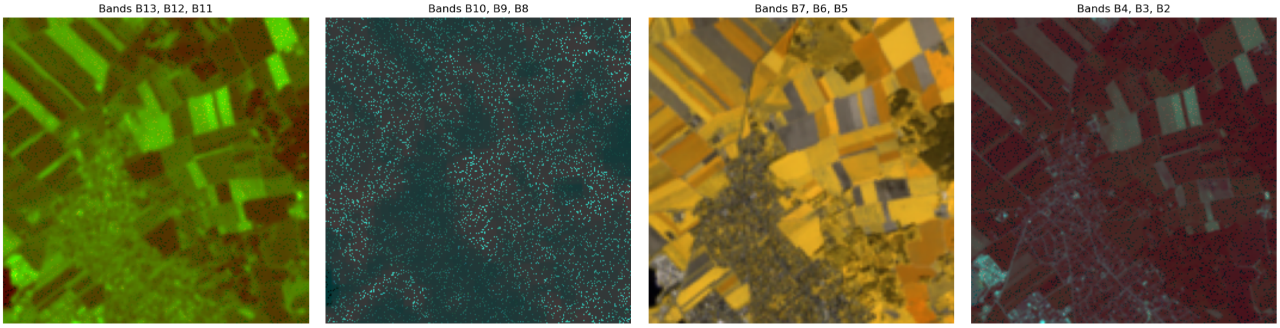


Figure B.9: Visualizing all bands of an image where bands B4, B8, B9, B13 were masked

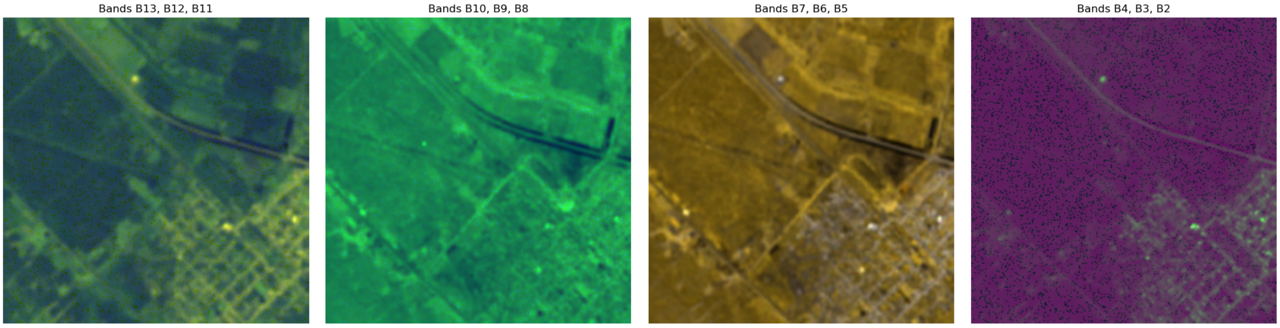


Figure B.10: Visualizing all bands of an image where bands B2, B4, B8, B11 were masked

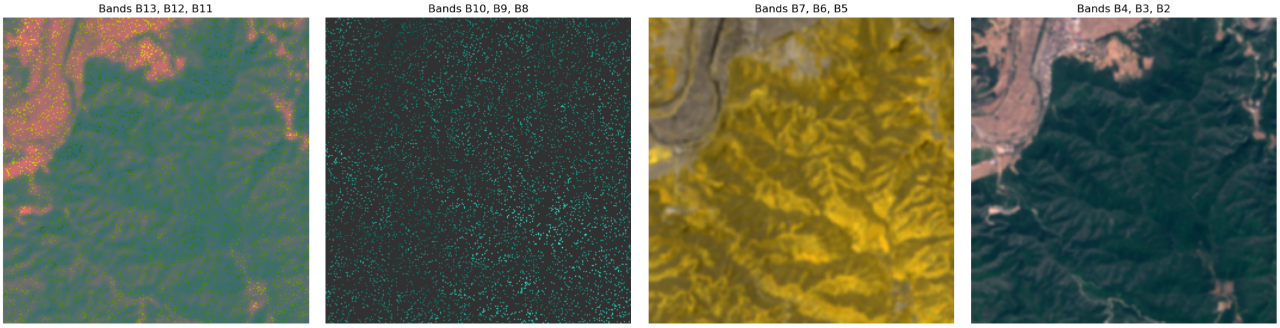


Figure B.11: Visualizing all bands of an image where bands B8, B9, B10, B11, B12 were masked

Bibliography

[1] Lingli Zhu, Juha Suomalainen, Jingbin Liu, Juha Hyyppä, Harri Kaartinen, Henrik Haggren, et al. A review: Remote sensing sensors. *Multi-purposeful application of geospatial data*, 19, 2018.

[2] Jon Christopherson, Shankar N Ramaseri Chandra, and Joel Q Quanbeck. 2019 joint agency commercial imagery evaluation—land remote sensing satellite compendium. Technical report, US Geological Survey, 2019.

[3] Lefei Zhang and Liangpei Zhang. Artificial intelligence for remote sensing data analysis: A review of challenges and opportunities. *IEEE Geoscience and Remote Sensing Magazine*, 10(2):270–294, 2022.

[4] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. 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): 2217–2226, 2019.

[5] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[6] Ce Zhou, Qian Li, Chen Li, Jun Yu, Yixin Liu, Guangjing Wang, Kai Zhang, Cheng Ji, Qiben Yan, Lifang He, et al. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. *arXiv preprint arXiv:2302.09419*, 2023.

[7] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[8] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[9] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9653–9663, 2022.

[10] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[11] Fan Liu, Delong Chen, Zhangqingyun Guan, Xiaocong Zhou, Jiale Zhu, Qiaolin Ye, Liyong Fu, and Jun Zhou. Remoteclip: A vision language foundation model for remote sensing. *IEEE Transactions on Geoscience and Remote Sensing*, 2024.

[12] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020.

[13] Johannes Jakubik, Sujit Roy, CE Phillips, Paolo Fraccaro, Denys Godwin, Bianca Zadrozny, Daniela Szwarcman, Carlos Gomes, Gabby Nyirjesy, Blair Edwards, et al. Foundation models for generalist geospatial artificial intelligence. *arXiv preprint arXiv:2310.18660*, 2023.

[14] Konstantin Klemmer, Esther Rolf, Caleb Robinson, Lester Mackey, and Marc Rußwurm. Satclip: Global, general-purpose location embeddings with satellite imagery. *arXiv preprint arXiv:2311.17179*, 2023.

[15] Yezhen Cong, Samar Khanna, Chenlin Meng, Patrick Liu, Erik Rozi, Yutong He, Marshall Burke, David Lobell, and Stefano Ermon. Satmae: Pre-training transformers for temporal and multi-spectral satellite imagery. *Advances in Neural Information Processing Systems*, 35:197–211, 2022.

[16] Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multi-modal fusion transformer for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7077–7087, 2021.

[17] Gabriel Tseng, Ruben Cartuyvels, Ivan Zvonkov, Mirali Purohit, David Rolnick, and Hannah Kerner. Lightweight, pre-trained transformers for remote sensing timeseries. *arXiv preprint arXiv:2304.14065*, 2023.

[18] Matt Allen, Francisco Dorr, Joseph A Gallego-Mejia, Laura Martínez-Ferrer, Anna Jungbluth, Freddie Kalaitzis, and Raúl Ramos-Pollán. Fewshot learning on global multimodal embeddings for earth observation tasks. *arXiv preprint arXiv:2310.00119*, 2023.

[19] Maofan Zhao, Qingyan Meng, Lifeng Wang, Linlin Zhang, Xinli Hu, and Wenxu Shi. Towards robust classification of multi-view remote sensing images with partial data availability. *Remote Sensing of Environment*, 306:114112, 2024.

[20] Robert A Schowengerdt. *Remote sensing: models and methods for image processing.* elsevier, 2006.

[21] John Wilson Rouse, Rüdiger H Haas, John A Schell, Donald W Deering, et al. Monitoring vegetation systems in the great plains with erts. *NASA Spec. Publ*, 351(1):309, 1974.

[22] Duccio Rocchini, Carlo Ricotta, and Alessandro Chiarucci. Using satellite imagery to assess plant species richness: The role of multispectral systems. *Applied Vegetation Science*, 10 (3):325–331, 2007.

[23] Ranga B Myneni, Forrest G Hall, Piers J Sellers, and Alexander L Marshak. The interpretation of spectral vegetation indexes. *IEEE Transactions on Geoscience and remote Sensing*, 33(2):481–486, 1995.

[24] Michel ED Chaves, Michelle CA Picoli, and Ieda D. Sanches. Recent applications of landsat 8/oli and sentinel-2/msi for land use and land cover mapping: A systematic review. *Remote Sensing*, 12(18):3062, 2020.

[25] Tianxiang Zhang, Jinya Su, Cunjia Liu, Wen-Hua Chen, Hui Liu, and Guohai Liu. Band selection in sentinel-2 satellite for agriculture applications. In *2017 23rd international conference on automation and computing (ICAC)*, pages 1–6. IEEE, 2017.

[26] James B Campbell and Randolph H Wynne. *Introduction to remote sensing.* Guilford press, 2011.

[27] Lei Ma, Yu Liu, Xueliang Zhang, Yuanxin Ye, Gaofei Yin, and Brian Alan Johnson. Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS journal of photogrammetry and remote sensing*, 152:166–177, 2019.

[28] Yi-Jen Shih, Hsuan-Fu Wang, Heng-Jui Chang, Layne Berry, Hung-yi Lee, and David Harwath. Speechclip: Integrating speech with pre-trained vision and language model. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 715–722. IEEE, 2023.

[29] Tianyu Huang, Bowen Dong, Yunhan Yang, Xiaoshui Huang, Rynson WH Lau, Wanli Ouyang, and Wangmeng Zuo. Clip2point: Transfer clip to point cloud classification with image-depth pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22157–22167, 2023.

[30] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 56(4):1–39, 2023.

[31] Yiwei Ma, Guohai Xu, Xiaoshuai Sun, Ming Yan, Ji Zhang, and Rongrong Ji. X-clip: End-to-end multi-grained contrastive learning for video-text retrieval. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 638–647, 2022.

[32] Angelos Zavras, Dimitrios Michail, Begüm Demir, and Ioannis Papoutsis. Mind the modality gap: Towards a remote sensing vision-language model via cross-modal alignment. *arXiv preprint arXiv:2402.09816*, 2024.

[33] Usha Bhalla, Alex Oesterling, Suraj Srinivas, Flavio P Calmon, and Himabindu Lakkaraju. Interpreting clip with sparse linear concept embeddings (splice). *arXiv preprint arXiv:2402.10376*, 2024.

[34] Rameen Abdal, Peihao Zhu, John Femiani, Niloy Mitra, and Peter Wonka. Clip2stylegan: Unsupervised extraction of stylegan edit directions. In *ACM SIGGRAPH 2022 conference proceedings*, pages 1–9, 2022.

[35] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[36] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. *OpenAI*, 2018.

[37] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[38] Siyuan Li, Luyuan Zhang, Zedong Wang, Di Wu, Lirong Wu, Zicheng Liu, Jun Xia, Cheng Tan, Yang Liu, Baigui Sun, et al. Masked modeling for self-supervised representation learning on vision and beyond. *arXiv preprint arXiv:2401.00897*, 2023.

[39] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International conference on machine learning*, pages 1691–1703. PMLR, 2020.

[40] Martina Zambelli, Antoine Cully, and Yiannis Demiris. Multimodal representation models for prediction and control from partial information. *Robotics and Autonomous Systems*, 123:103312, 2020.

[41] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European conference on computer vision (ECCV)*, pages 85–100, 2018.

[42] Krishna Kumar Singh, Hao Yu, Aron Sarmasi, Gautam Pradeep, and Yong Jae Lee. Hide-and-seek: A data augmentation technique for weakly-supervised localization and beyond. *arXiv preprint arXiv:1811.02545*, 2018.

[43] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

[44] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[45] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[46] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[47] Yaya Heryadi, Edy Irwansyah, Eka Miranda, Haryono Soeparno, Kiyota Hashimoto, et al. The effect of resnet model as feature extractor network to performance of deeplabv3 model for semantic satellite image segmentation. In *2020 IEEE Asia-Pacific Conference on Geoscience, Electronics and Remote Sensing Technology (AGERS)*, pages 74–77. IEEE, 2020.

[48] Michael Schmitt; Lloyd Hughes; Pedram Ghamisi; Naoto Yokoya; Ronny Hansch. 2020 ieee grss data fusion contest, 2019. URL https://dx.doi.org/10.21227/rha7-m332.

[49] Michael Schmitt, Lloyd Haydn Hughes, Chunping Qiu, and Xiao Xiang Zhu. Sen12ms–a curated dataset of georeferenced multi-spectral sentinel-1/2 imagery for deep learning and data fusion. *arXiv preprint arXiv:1906.07789*, 2019.

[50] Raoof Naushad, Tarunpreet Kaur, and Ebrahim Ghaderpour. Deep transfer learning for land use and land cover classification: A comparative study. *Sensors*, 21(23):8083, 2021.

[51] H Yassine, K Tout, and M Jaber. Improving lulc classification from satellite imagery using deep learning–eurosat dataset. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 43:369–376, 2021.

[52] Damien Sulla-Menashe, Mark A Friedl, et al. User guide to collection 6 modis land cover (mcd12q1 and mcd12c1) product. *Usgs: Reston, Va, Usa*, 1:18, 2018.

[53] Thomas R Loveland and AS Belward. The igbp-dis global 1km land cover data set, discover: First results. *International Journal of Remote Sensing*, 18(15):3289–3295, 1997.

[54] R Binet, E Bergsma, and V Poulain. Accurate sentinel-2 inter-band time delays. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 1:57–66, 2022.

[55] Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.

[56] DP Kingma. Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[57] Leiyu Chen, Shaobo Li, Qiang Bai, Jing Yang, Sanlong Jiang, and Yanming Miao. Review of image classification algorithms based on convolutional neural networks. *Remote Sensing*, 13(22):4712, 2021.

[58] Maxim Neumann, Andre Susano Pinto, Xiaohua Zhai, and Neil Houlsby. In-domain representation learning for remote sensing. *arXiv preprint arXiv:1911.06721*, 2019.

[59] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, 29, 2016.

[60] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.