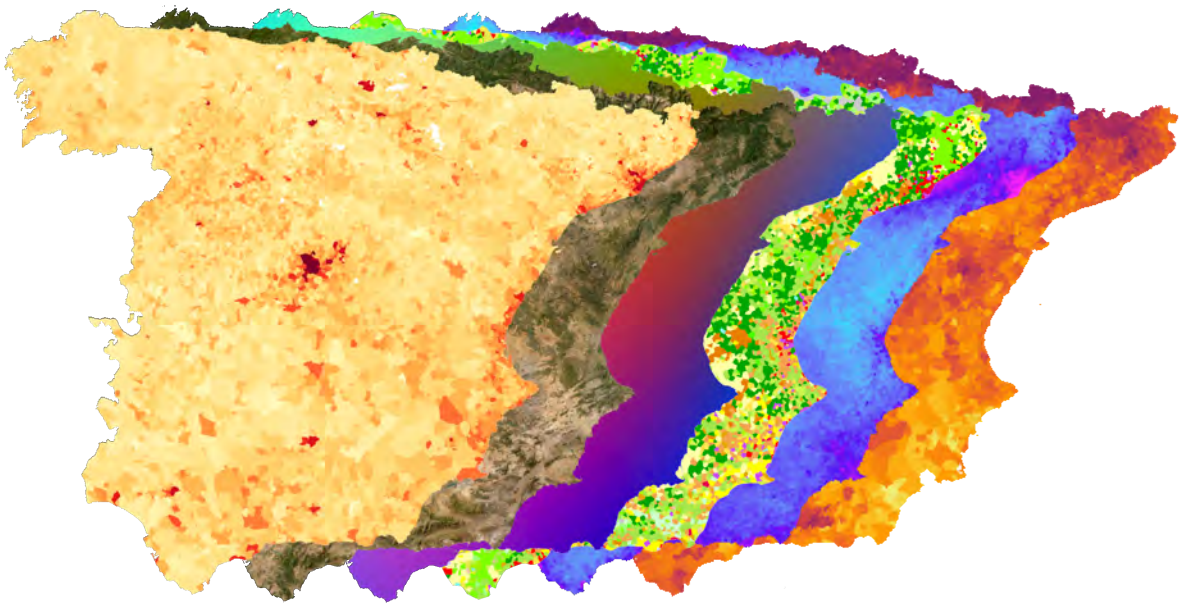
Geo-information Science and Remote Sensing

MSc thesis Geo-Information Science

---

# COMPARING IMAGE AND LOCATION ENCODER MODELS IN THE CONTEXT OF DISEASE MAPPING

Levien van Krieken



May 4, 2025

WAGENINGEN
UNIVERSITY & RESEARCH

# Comparing image and location encoder models in the context of disease mapping

Levien van Krieken

Registration number 1412671

Supervisors:

dr. MC (Marc) Rußwurm

dr.ir. S (Sytze) de Bruin

A thesis submitted in partial fulfilment of the degree of Master of Science

at Wageningen University and Research Centre,

The Netherlands.

May 4, 2025

Wageningen, The Netherlands

Report number : GIRS-2025-29

Thesis course code : GRS80436

Wageningen University and Research Centre

Laboratory of Geo-Information Science and Remote Sensing

# Contents

# 1. Introduction

Where we are and what defines our environment has impact on how we live and how we feel. It has influence on our happiness [1] and our health [2]. It is hard to define what makes one location different from another and there are thus many ways to abstract such locations. A common way to define the environment of a location is by how the space is used, for example for forest, as urban living space or for agriculture. Another is to try and define how much and what kind of vegetation is present at a location. Inherently all of these representations are an abstraction of the complex nature of all factors which represent our environment.

In spatial modelling of anything, the objective is to find how a location where something is known relates to a location where the same thing is unknown. The classical approach to relating known locations to unknown locations is often based on Tobler's First Law of Geography, which states that '*everything is related to everything else, but near things are more related than distant things*' [3]. This idea forms the basis of spatial interpolation methods like inverse distance weighting [4] and Kriging [5]. This idea is intuitive and allows us to compare two locations by their spatial distance.

For predicting something at an unknown location another assumption can be made: everything is related to everything else, but similar things are more related than dissimilar things. This is the concept of the semantic distance between two locations. The idea here is that while two locations could be spatially close, they do not have to be similar. Take the prediction of population density as an example. In an area there are two cities, with a bunch of rural land in between. Between the two cities there is a large spatial distance, but a small semantic distance. Each of the cities is only a walk away from rural land, but have a larger semantic distance to the rural location than to the other city. When predicting the population density of a new location, what if interpolation could be done semantically alongside spatially?

To be able to semantically define a new location, we can use remotely sensed imagery taken from space. One of the satellite image products is Sentinel-2 [6]. Sentinel-2 covers all land areas and is freely available at weekly time intervals. Furthermore, Sentinel-2 is multi-spectral and allows for capture of radiance outside the visible light spectrum. Multi-spectral satellite images are an accessible source of environmental data for any location on Earth and are commonly used for environmental indices like the NDVI [7] or the Temperature suitability [8]. These indices are the traditional way to attribute semantic information to a location by applying carefully engineered functions to the multi-spectral images.

A more recent strategy for attributing semantic meaning to a location is by analysing the corresponding satellite images with deep learning models. Computer vision has revolutionised image processing, enabling models to automatically extract meaningful patterns and features from visual data. Advancements in architectures which were initially made for RGB three-channel images [9, 10, 11], are able to be modified to allow multi-spectral images. As their artificial neural networks contain adjustable weights, deep learning models can be trained to extract the most relevant patterns in an image. The part of a deep learning model that is used for this feature extraction is known as an encoder. Encoders are models which encode their input, typically into a $n$-dimensional vector, called an **embedding**. Embeddings are an thus an interpretation of the input by an encoder, mapped to a set of representable numbers.

Locations can be encoded with **image encoders** by extracting the embedding of a satellite image at that location. In order for image encoders to extract meaningful information, the models are usually pre-trained to find the most discriminating features. Pre-training can be done in with supervision, by

learning from examples with labels. More recently, the advancements in **self-supervised learning** (SSL) have allowed image encoders to be pre-trained without the need for labels. Instead, these methods learn to extract relevant features from a representable image dataset by solving auxiliary tasks. The tasks use different strategies, such as predicting missing parts [12], contrasting similar and dissimilar samples [13, 14] or aligning representations [15]. With these objectives, SSL allows encoders to learn semantically rich representations from different kinds of image datasets.

Models can be pre-trained on satellite imagery to get encoders capable of extracting a semantic representation for a location. Taking satellite image embeddings for different locations, the similarity between these locations can be analysed. Furthermore, variables can be interpolated for new locations based on their semantic similarity. Apart from (satellite) image encoders, a different kind of strategy for embedding locations was recently introduced. They use SSL to learn semantic representations directly from coordinates [16, 17, 18]. These **location encoders** can be pre-trained with images to learn image-based representations without the need for any images to encode locations later.

Representing locations and environmental factors is an important part in the context of **disease mapping**. Disease mapping models use incidence or mortality data on specific diseases and apply spatial Bayesian methods to predict localised disease risk [19, 20, 21]. These models can use random effects to smooth out the variation in risk spatially, following Tobler's first law by assuming neighbouring areas have more similar risks. However, this variation can alternatively be partially or completely explained with relevant semantic covariates like pollution levels, demographics or land cover [22].

Instead of known covariates, more abstract semantic representation could help with approximating relative disease risk. The embeddings of locations from deep learning encoders could help with representing for the disease mapping. The pre-trained image and location encoders extract different kinds of semantic features depending on architecture and pre-training data and objective. What kind of encoder model could be best suited for disease mapping?

### Research questions

1. What distinguishing visual patterns emerge on maps when comparing spatial embeddings?
2. What kinds of embeddings have strongest correlation with selected disease predictor tasks?
3. What is the effect of taking embeddings of satellite imagery from an intermediate layer of a convolution-based encoder?
4. What is the effect of sampling disease prediction variables spatially?

**Research objective** The goal of this thesis is to compare different strategies for encoding locations with deep learning models. These models differ in their neural network architecture and their pre-training strategies, where most of the models will rely on extracting location-specific image features from Sentinel-2 satellite imagery. Evaluating the models is done both qualitatively and quantitively. For the qualitative part, the embeddings are visualised on maps to find out what distinguishing semantic patterns are encoded spatially. Quantitatively, the embeddings are tested by correlating them to various downstream prediction tasks related to human diseases. The research further explores the use of embeddings from intermediate layers in convolutional neural networks and the effect of spatial sampling on the downstream tasks. Ultimately, the main goal of this thesis is to contribute to assessing the kind of encoding model that provides the most relevant semantic environmental features for use in disease mapping.

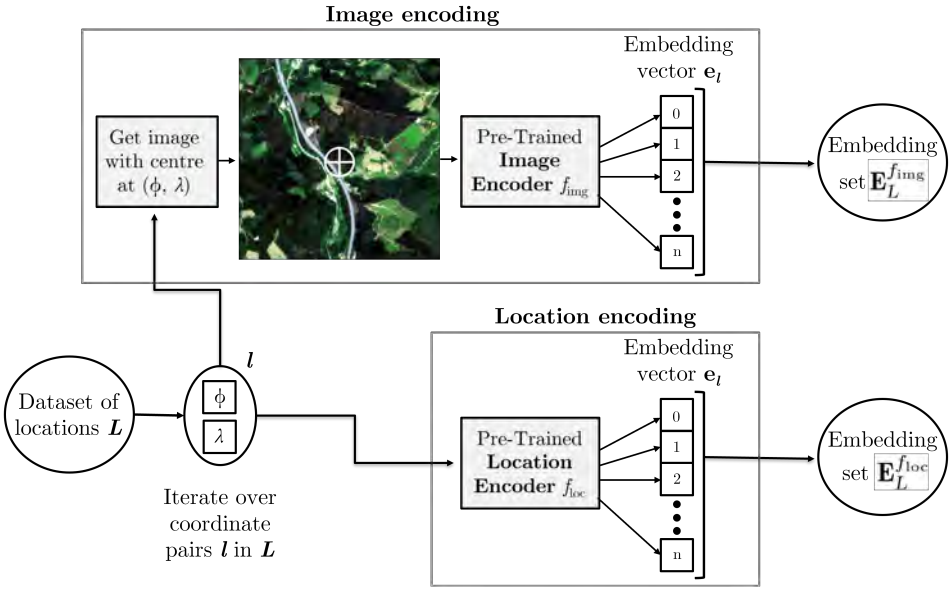



**Figure 1.** Schematic overview of how a set of locations $L$ gets encoded. Each location $\boldsymbol{l} \in L$ is defined by a coordinate pair $(\lambda, \phi)$ and gets encoded into an embedding vector $\mathbf{e}_{\boldsymbol{l}}$. Encoding $L$ with an image encoder model $f_{\text{img}}$ results in the embedding set $\mathbf{E}_L^{f_{\text{img}}}$ and encoding with a location encoder model $f_{\text{loc}}$ results in $\mathbf{E}_L^{f_{\text{loc}}}$.

## 2. Encoder models

To determine which encoder models are best suited for disease mapping, this section defines spatial embeddings and explains how they are generated by encoder models. For this research, a location $\boldsymbol{l}$ is defined as a point on the $S^2$ sphere of Earth with a corresponding latitude-longitude coordinate pair $(\lambda, \phi)$. To get a spatial embedding vector $\mathbf{e}$, this location needs to be encoded with a spatial encoding model $f$, where $\mathbf{e} = f(\boldsymbol{l})$. Given a set of locations $L = \{\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_n\}$, $f$ can be used to generate the spatial embeddings $\mathbf{E}_L^{\text{f}} = \{f(\boldsymbol{l}) \mid \boldsymbol{l} \in L\} = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n\}$.

The embeddings of the location $\boldsymbol{l}_i$ come from models that are broadly separated in three categories by their neural network (NN) architecture. **Image encoder** models require images for their embedding. Encoding $\boldsymbol{l}_i$ with an image encoder $f_{\text{img}}$ thus entails getting an image $\mathbf{I}_i$ at $\boldsymbol{l}_i$. The spatial embeddings for $L$ using $f_{\text{img}}$ are then generated by $\mathbf{E}_L^{f_{\text{img}}} = \{f_{\text{img}}(\mathbf{I}) \mid \mathbf{I} \text{ at } \boldsymbol{l},\ \boldsymbol{l} \in L\}$. **Location encoders** do not require images to embed locations. These models generate embeddings from just the coordinates. An location encoder $f_{\text{loc}}$ generates the spatial embeddings $\mathbf{E}_L^{\text{f}_{\text{loc}}} = \{f_{\text{loc}}(\boldsymbol{l}) \mid \boldsymbol{l} \in L\}$.[1]

A schematic overview of how image and locations encoders extract embeddings is in Figure 1. The architectures and pre-training methods for these two categories of models will be explained in this section. The third category of models is the **baselines**, which contain models that are not based on NN's and are used to compare the encoder models against. The baseline models are explained in section 4.

[1] The models can also be explained as Remote Sensing Foundation Models, following the terminology used by Guo et al. [23]. The image encoders are then known as Remote Sensing Vision Foundation Models, and the location encoders as Remote Sensing Vision-Location Foundation Models.

## 2.1. Image Encoders

### 2.1.1. Architecture

This category of models uses trainable weights to extract features from images. In modern deep learning, two main functions are used to extract these features, either with convolutions or self-attention.

**Convolutions** are done by sliding a weighted kernel window over the input tensor to find localised patterns in the input image and result in feature layers for the image. By putting multiple convolutions in succession, features of different levels of complexity can be embedded. A residual network (**RN** or **ResNet**)[10] has five blocks of convolutional layers and improved image processing by including skip connections between each block and the next. A ResNet architecture typically does a global pooling operation after the last block of convolutions followed by a final fully connected layer without convolutions. This last layer is typically used to relate patterns of features to targets relevant for specific tasks.

A **transformer** [24] is an architecture which uses **self-attention** functions on a sequence of token-vectors to learn relations. It was originally developed for natural language processing. A Vision Transformer (**ViT**)[11] is an adapted version of this idea, using images as input. To use images with this architecture, the image is first reshaped into a sequence of flattened patches before being mapped with a linear projection. This projection is used to encode patches of $16 \times 16$ pixels as tokens. To make up for the loss of spatial relations within the image, each of the embedded patch tokens receives a learnable positional encoding as well. These patches then operate as the tokens for the transformer part of the ViT. ViT adds a learnable [CLASS] token, whose representation is used with a fully connected layers as the last function of the architecture. This serves a similar purpose to the global pooling and fully connected layer of a ResNet in that it considers the entire image and uses the extracted features for task-specific purposes.

Convolutions force a local receptive field where a pixel can only be influenced by neighbours within the kernel window. The attention mechanism present within a transformer instead allows a ViT to attend all other patches globally, enabling long-range dependencies.

### 2.1.2. Pre-training

Methods which use **self-supervised learning** allow models to be trained on a dataset without the need for any of the samples to be annotated. For image-based models, self-supervised methods like SimCLR [14] instead use a contrastive objective to find the most defining features in an image. This works by providing two augmented positive views of an image and comparing the output embeddings. The models are then trained by rewarding similarity between the positive pair while penalising similarity with negative samples.**MoCo** [13] uses a similar strategy, but trains two encoders separately. The query encoder is trained with back-propagation and gradient descent. The other is instead updated with a slow-moving average of the query encoder's weights, called momentum. This momentum encoder uses a queue of negative samples, which allows for consistency across batches.

**DINO** [15] also uses a momentum update, but does not require negative samples to find the most defining features. Instead this uses knowledge **di**stilation [25] with **no** labels. It's loss function still matches augmented views, but instead of trying to minimise the similarity to negative samples, DINO tries to create a cluster of outputs specific to that image. Not only does this mean that DINO does not require the queue MoCo does, it also means that if the dataset includes some similar images, DINO would not penalise the model for generating similar outputs.

Models pre-trained with MoCo or DINO are very useful for taking generalised image embeddings, as they are by definition trained to find the most distinguishing features in their training dataset. For Sentinel-2 satellite imagery, one such dataset is **SSL4EO**-S12 [26]. The dataset contains Sentinel-1 and Sentinel-2 images for over 250,000 locations around the world. Moreover, pre-trained image encoders which have been trained on this dataset are freely available.

A different way to pre-train image encoders is with Masked Autoencoding (MAE) [12]. Instead of learning to match augmentations, MAE masks part of the input before encoding. The objective of MAE is then to reconstruct the original image using a decoder. Multi-pretext MAE (MP-MAE) does not just encode and decode images, but also other relevant information at the same location. MP-MAE is used with an ConvNeXt V2 architecture [27] on the **MMEarth** [28] dataset to pre-train an image encoder. This dataset contains modalities for 1.2 million locations which includes satellite imagery from Sentinel-1 and Sentinel-2, as well as other pixel and image-level modalities like land cover, elevation and temperature. While both an encoder and decoder are necessary for the pre-training, just the pre-trained encoder is used to extract the embeddings.

## 2.2. Location encoders

### 2.2.1. Architecture

A different strategy to encode locations is with geographic **location encoders**. These encoders are typically a combination of a non-parametric positional encoding $\mathrm{PE}(\cdot)$ for $\boldsymbol{l}$ and a trainable neural network $\mathrm{NN}(\cdot)$. The embedding vector is then obtained with $\mathbf{e} = \mathrm{NN}(\mathrm{PE}(\boldsymbol{l}))$. $\mathrm{PE}(\boldsymbol{l})$ is a deterministic function that transforms the coordinates into a positional embedding vector, while $\mathrm{NN}(\cdot)$ provides a learnable component to the location encoder. These location encoders can then be trained with supervision to learn the interaction between input coordinates and target labels [16].

**SatCLIP** [18] uses spherical harmonics(SH) [29] for its positional encoding and a SIREN neural network [30]. The spherical harmonics have a history in Earth sciences and are particularly suited for coordinates on the surface of spheres. Their spatial smoothness is controlled by the number of Legendre polynomials $L$. The SIREN network uses periodic activation functions which are particularly suited for implicit neural representations of complex natural signals. This makes it a logical choice for SatCLIP, as its goal is to encode satellite image features implicitly into coordinate pairs.

**GeoCLIP** [17] has a positional encoder which transforms $\boldsymbol{l}$ into the Equal Earth projection (EEP) [31] before extracting features using Random Fourier Features (RFF) [32]. RFF allow for high-frequency embeddings from low-dimensional inputs. The frequency in RFF is varied to capture features at three different spatial scales and for each scale a MLP is pre-trained on the image model to get a hierarchical representation. The embeddings from different levels are then summed element-wise to obtain the location embedding for $\boldsymbol{l}$.

### 2.2.2. Pre-training

One way to train a location encoder for generalised spatial embeddings is by providing image embeddings. This subset of location encoders use a contrastive objective to learn to match location embeddings to the embeddings of an image at the same location. The idea for this is that by pre-training with an image encoder, the location encoder can infer image features without the need for the images themselves. This Contrastive Location-Image Pre-training (CLIP) is based on the Contrastive Language-Image Pre-training [33]. Both of the models mentioned in the previous section are trained with this CLIP objective.

SatCLIP is pre-trained on satellite imagery samples from around the globe, allowing for particularly good results in areas that traditionally have less sample coverage. SatCLIP models are pre-trained on their own dataset of Sentinel-2 imagery, S2-100K. The location encoder is then trained contrastively against the image encoder. The MoCo pre-trained ResNet and ViT image encoders from SSL4EO [26], explained in section 2.1.2, were used for this purpose.

GeoCLIP has instead been pre-trained on a dataset of geo-tagged terrestrial RGB images from Flickr created for the 2016 MediaEval workshop [34]. This model also has another difference to the SatCLIP models. The ViT image encoder to learn from was pre-trained with the Contrastive Language-Image Pre-training [33], effectively embedding language, image and location-based features for each $\boldsymbol{l}$.


**Figure 2.** Boundaries of the Spanish municipalities [35] used in this thesis.

# 3. Data

## 3.1. Study area

For this thesis, researchers at the Public University of Navarre (UPNA) provided data consisting of the modelled expected and truly observed cases for three rare diseases. The data was provided aggregated at municipal level in Spain and the research area comprises all municipalities of peninsular (continental) Spain. Accordingly, the research does not consider the Canary Islands, the Balearic Islands nor the autonomous regions at the coast of Africa. Furthermore, some small regions of peninsular Spain are too sparsely populated to provide meaningful data, and are also not considered. The municipal dataset [35] has 7954 municipalities, which are mapped out in Figure 2. The municipalities vary significantly in population size and area. According to 2022 population data [36], the most populous municipality (Madrid) has over 3.2 million inhabitants while the least populous municipality (Illán de Vacas) has only three. Similarly, the largest municipality (Cáceres) in this dataset has an area of just over 1750km$^2$ and the smallest (Emperador) an area of about 0.025km$^2$[2].

[2] A 2022 review of the municipal boundaries by the Valencian Cartographic Institute measured a different smallest municipality (Llocnou de la Corona), but this change was not yet in the data for this thesis.

**(a)** $L_{centroid}$ **(b)** $L_{SRS20k}$

**Figure 3.** Sampling locations for the satellite image datasets.

### Sentinel-2 satellite imagery

For the thesis, two distinct sets of locations were used. These locations are mapped out in Figure 3. For every location in both dataset, multi-spectral (13-band) 256×256 Sentinel-2 L1C images were downloaded using the Google Earth Engine [37], such that the location is at the centre of the Sentinel-2 image. For the first dataset, $L_{centroid}$, the locations are the centroids of each municipality resulting in 7954 locations and images.

For the second dataset, locations were sampled using a stratified random sampling strategy where the number of locations sampled in each municipality was depended on the area of the municipality. This dataset has 20,000 locations in total. To ensure compatibility with municipal level data, at least one location was sampled for each municipality. The remaining 12,046 locations were distributed proportionally to the area and sampled randomly within those municipalities. The resulting *Stratified Random Sample* of locations with corresponding images is the $L_{SRS20k}$ dataset.

## 3.2. Data for downstream tasks

The municipal data for the diseases was provided by researchers at the UPNA. There are three different diseases in this dataset. For confidentiality reasons, it is not revealed which diseases are in the dataset, which is not a problem for the scope of this thesis. Following the naming convention of the original data, these diseases are referred to as $D_1$, $D_2$ and $D_3$. For each of the diseases, both the observed and the expected counts are provided, O and E. This expected count is based on the age distribution of the population within each municipality and takes into account the relative disease risk for each age group [38]. To normalise E, the rate per 100,000 inhabitants was calculated based on municipal population data [36], resulting in $\text{E100k}_1$, $\text{E100k}_2$ and $\text{E100k}_3$ respectively.

This thesis uses four disease predictors along with the disease data for the downstream tasks. The first is the most recent CORINE Land Cover (**CLC**) dataset [39]. This dataset from 2018 offers 44 thematic land cover classes. From a global temperature dataset [40] the minimum ($\mathbf{T_{min}}$) and maximum temperature ($\mathbf{T_{max}}$) are used. The temperature data is from June of 2019 and was converted to °C. The last dataset[41] has been created for the particulate matter concentration smaller than 2.5µm (**PM2.5**), averaged over 2022.

# 4. Experimental setup

## Models

Following the objective for the thesis, different kinds of encoding models were compared both qualitatively and quantitatively. An overview of the encoder models used is in Table 1. As mentioned in the encoders section, image encoders extracted embeddings with a feed-forward encoding of the satellite images, while the location encoders extracted embeddings from the location coordinate pair.

All of the **image encoder models** apart from MMEarth [28] were pre-trained on SSL4EO-S12[26]. The SSL4EO models vary in model architecture and pre-training methodology and were implemented using TorchGeo [42]. To get the embeddings from the MMEarth model, an average pooling operation was added after the encoding of the Sentinel-2 imagery.

While the **location encoder models** from SatCLIP [18] all use the same architecture and pre-training method, these vary in two other pre-training parameters. The first is the image encoder they were pre-trained with, either a ResNet18, a ResNet50 or a Visual Transformer [10, 11]. Each of these models also come in two levels of smoothness, $L = 10$ or $L = 40$.

Along with the pre-trained encoders, two **baseline models** were used. These models have no encoding architecture and are used as a baseline evaluation to compare the encoders to. 'Random' contains embeddings with 10 uniformly random sampled values between 0 and 1. 'Mean reflection' is a baseline embedding of length 13 which is calculated by taking the mean average reflectance per band for the Sentinel-2 image at each location.

For each model, three sets of embeddings were extracted. The first two are obtained by passing the coordinates or images from the $L_{centroid}$ and $L_{SRS20k}$ datasets directly through the models. Encoding locations $L$ with $f$ thus gives the embedding matrix $\mathbf{E}_L^f \in \mathbb{R}^{n \times \mathbf{D}}$ where M has an output embedding size of $\mathbf{D}$ dimensions for $n$ locations in $L$. For example, running the images of $L_{centroid}$ through the SSL4EO ViT-DINO model gives embeddings $\mathbf{E}_{L_{SRS20k}}^{\text{ViT-DINO}} \in \mathbb{R}^{20000 \times 1000}$. The third set of embeddings $\mathbf{E}_{avg\text{-}muni}^{\text{M}}$ was calculated by averaging the embeddings from $L_{SRS20k}$ per municipality to be able to use them for tasks on municipal level. Averaging the example embedding $\mathbf{E}_{L_{SRS20k}}^{\text{ViT-DINO}}$ then results in $\mathbf{E}_{avg\text{-}muni}^{\text{ViT-DINO}} \in \mathbb{R}^{7954 \times 1000}$.

**Table 1.** Encoder models used and the baselines (BL). Includes the encoder architecture, the self-supervised learning (SSL) method, the data used and the output embedding dimension size.

| | Model | Encoder architecture | SSL method | Pre-training dataset | Data type | Dimension size |
|---|---|---|---|---|---|---|
| **BL** | Mean reflection | - | - | - | L1C Sentinel-2 | 13 |
| | Random | - | - | - | - | 10 |
| **Image Enc.** | RN18-MoCo$_{fc}$ | ResNet18 | MoCo | SSL4EO-S12 | L1C Sentinel-2 | 1000 |
| | RN50-MoCo$_{fc}$ | ResNet50 | MoCo | SSL4EO-S12 | L1C Sentinel-2 | 1000 |
| | RN50-DINO$_{fc}$ | ResNet50 | DINO | SSL4EO-S12 | L1C Sentinel-2 | 1000 |
| | ViT-DINO | ViT-16 | DINO | SSL4EO-S12 | L1C Sentinel-2 | 1000 |
| | ViT-MoCo | ViT-16 | MoCo | SSL4EO-S12 | L1C Sentinel-2 | 1000 |
| | MMEarth | ConvNeXt V2 | MP-MAE | MMEarth | Multi-modal | 320 |
| **Location Enc.** | SatCLIP-RN18$_{L=10}$ | SirenNet(SH) | CLIP | S2-100K | L2A Sentinel-2 | 256 |
| | SatCLIP-RN18$_{L=40}$ | SirenNet(SH) | CLIP | S2-100K | L2A Sentinel-2 | 256 |
| | SatCLIP-RN50$_{L=10}$ | SirenNet(SH) | CLIP | S2-100K | L2A Sentinel-2 | 256 |
| | SatCLIP-RN50$_{L=40}$ | SirenNet(SH) | CLIP | S2-100K | L2A Sentinel-2 | 256 |
| | SatCLIP-ViT$_{L=10}$ | SirenNet(SH) | CLIP | S2-100K | L2A Sentinel-2 | 256 |
| | SatCLIP-ViT$_{L=40}$ | SirenNet(SH) | CLIP | S2-100K | L2A Sentinel-2 | 256 |
| | GeoCLIP | RFF with MLP's | CLIP | MP-16 | Flickr images | 512 |

## 4.1. Embedding pattern visualisation

To answer the first research question regarding the distinguishing visual patterns which emerge on maps when comparing spatial embeddings, the embedded locations were visualised spatially and compared. Two locations $\boldsymbol{l}_1$ and $\boldsymbol{l}_2$ are compared by the relation between the geographic distance and the embedded semantic distance. Patterns between embeddings can then be analysed by relating spatial closeness to semantic closeness. This was done visually on maps that show a semantic comparison. Maps show geographic closeness inherently and by visualising semantic distance, spatial embedding patterns can be compared. The semantic distance was approximated with two different methods which show different semantic properties.

The first method relies on principal component analysis (**PCA**), which is a dimensionality reduction technique used to capture the largest variance within a dataset in $p$ principal components. These principal component vectors can then be used for a transformation, reducing the embedding dimension to $p$. By choosing $p = 3$ and normalising the values to integers between 0 and 255, PCA was utilised as a RGB-transformation, mapping the largest variance within the embeddings to red, green and blue values. This RGB transformation was done on $\mathbf{E}^f_{SRS20k}$ and the resulting RGB points were mapped onto Thiessen polygons (geographical Voronoi diagram) for improved visual readability.

The second method involves the **cosine similarity** (COS_SIM) metric. This metric is used to calculate the similarity between a specific embedding and all others within the same embedding set. The cosine similarity between the embeddings of two locations is then calculated with $\text{COS\_SIM}(\mathbf{e}_1, \mathbf{e}_2) = \frac{\mathbf{e}_1 \cdot \mathbf{e}_2}{\|\mathbf{e}_1\| \|\mathbf{e}_2\|}$ The semantic distance between two locations is then the inverse of their similarity. While the PCA-transformation shows how embeddings all compare to one another, cosine similarity can be used to visualise how a specific location $\boldsymbol{l}$ compares to all other. This is done by calculating $\text{COS\_SIM}(f(\boldsymbol{l}), f(\boldsymbol{l}_i))$ for all $\boldsymbol{l}_i \in L$. To visualise the cosine similarity, the municipal locations from $L_{centroid}$ were used to compare all municipal embeddings to the embedding of Madrid. The municipality of Madrid was chosen as a target municipality to compare against to find whether other theoretically semantically similar urban areas show high similarity to it.

## 4.2. Downstream tasks

To answer the second research question, the encoder models were compared on their ability to correlate to disease predictor variables. The embeddings were validated on several downstream regression tasks, based on the disease predictor variables mentioned in section 3.2. The objective here is to find out which pre-calculated embeddings are best able to correlate to these disease predictors, and not necessarily to find the best correlation or approximation.

The disease predictor variables were sampled at the locations from both $L_{centroid}$ and $L_{SRS20k}$ dataset. For $\text{T}_{\text{min}}$, $\text{T}_{\text{max}}$, PM2.5 ,$\text{E100k}_1$, $\text{E100k}_2$ and $\text{E100k}_3$, predictions were fit with a linear regression. A linear regression fits slope parameters $\boldsymbol{\beta}$ and an intercept $\alpha$ such that the predictions $\widehat{\mathbf{y}} = \mathbf{E}^f_L \cdot \boldsymbol{\beta} + \alpha$ result in the lowest residual sum of squares ($SS_{\text{res}}$) compared to the predictor values $\mathbf{y}$. A linear regression thus tries to minimise $SS_{\text{res}} = \sum_i^n (y_i - \hat{y}_i)^2$, for the $n$ locations in the training set. Half of the available samples (50%) of a predictor variable were selected randomly for training. The other half of the samples were used for testing.

The $R^2$ coefficient was calculated over the test set and used for evaluation of the linear fit. This coefficient is calculated as shown in Equation 1 by dividing the sum of the residual sum of squares ($SS_{\text{res}}$), with the total variance in the downstream task dataset known as the total sum of squares($SS_{\text{tot}}$). This fraction is subtracted from one to get the proportion of variance in the dataset which can be

predicted from the embeddings. Note that $R^2$ has an upper bound of 1, but no lower bound. An $R^2$ of 0 is often thought of as a lower bound, implying no correlation. A constant prediction of $\bar{y}$ for all samples would give this $R^2$. An $R^2$ lower than 0 implies negative correlation. This can for example happen when the training sample is not representable for the entire dataset.

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}} = 1 - \frac{\sum_i^n (y_i - \hat{y}_i)^2}{\sum_i^n (y_i - \bar{y})^2} \quad \text{where} \quad \begin{aligned} n &= \text{number of evaluated locations} \\ y_i &= \text{dataset sample value for } \boldsymbol{l}_i \\ \bar{y} &= \text{mean value of samples} \\ \hat{y}_i &= \text{predicted value for } \boldsymbol{l}_i = f(\boldsymbol{l}_i) \cdot \boldsymbol{\beta} + \alpha \\ \boldsymbol{\beta} &= \text{fitted slope vector of the regression} \\ \alpha &= \text{intercept of the regression} \end{aligned} \tag{1}$$

For the **CLC** prediction, the same 50-50 test-train split was used, but its nominal data was fit using a logistic regression instead. A detailed explanation of the logistic regression objective is in Appendix C. For learning this regression, a maximum of 100 iterations were used. After the training, the fit was used with the test set and the % top-one accuracy is calculated as evaluation metric.

### 4.3. ResNet intermediate layer embeddings

Residual networks encode low and high-level features depending on the depth of the model. A residual network is made up of 5 blocks of convolutions with residual connections between them. A schematic visualisation for the ResNet50 architecture is shown in Figure 4. The early blocks encode low-level features, e.g. corners, edges and simple relations between bands. The deeper layers encode more high-level features, which are usually complex patterns and objects. For research question 3, embeddings were taken directly from intermediate layers of the ResNet model. This was done by removing deeper layers and applying a global average pooling function over the feature maps after the residual connection. The embeddings from the different convolutional layers were evaluated on their disease predictor correlation ability by repeating the experimental setup from subsection 4.2. These deeper embeddings were also compared to the embeddings from the final fully connected (fc) layer.

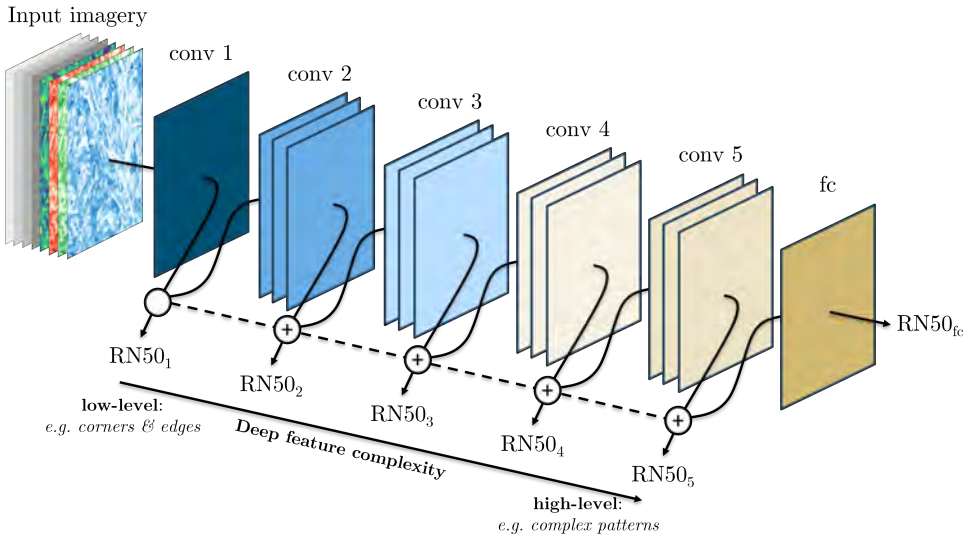
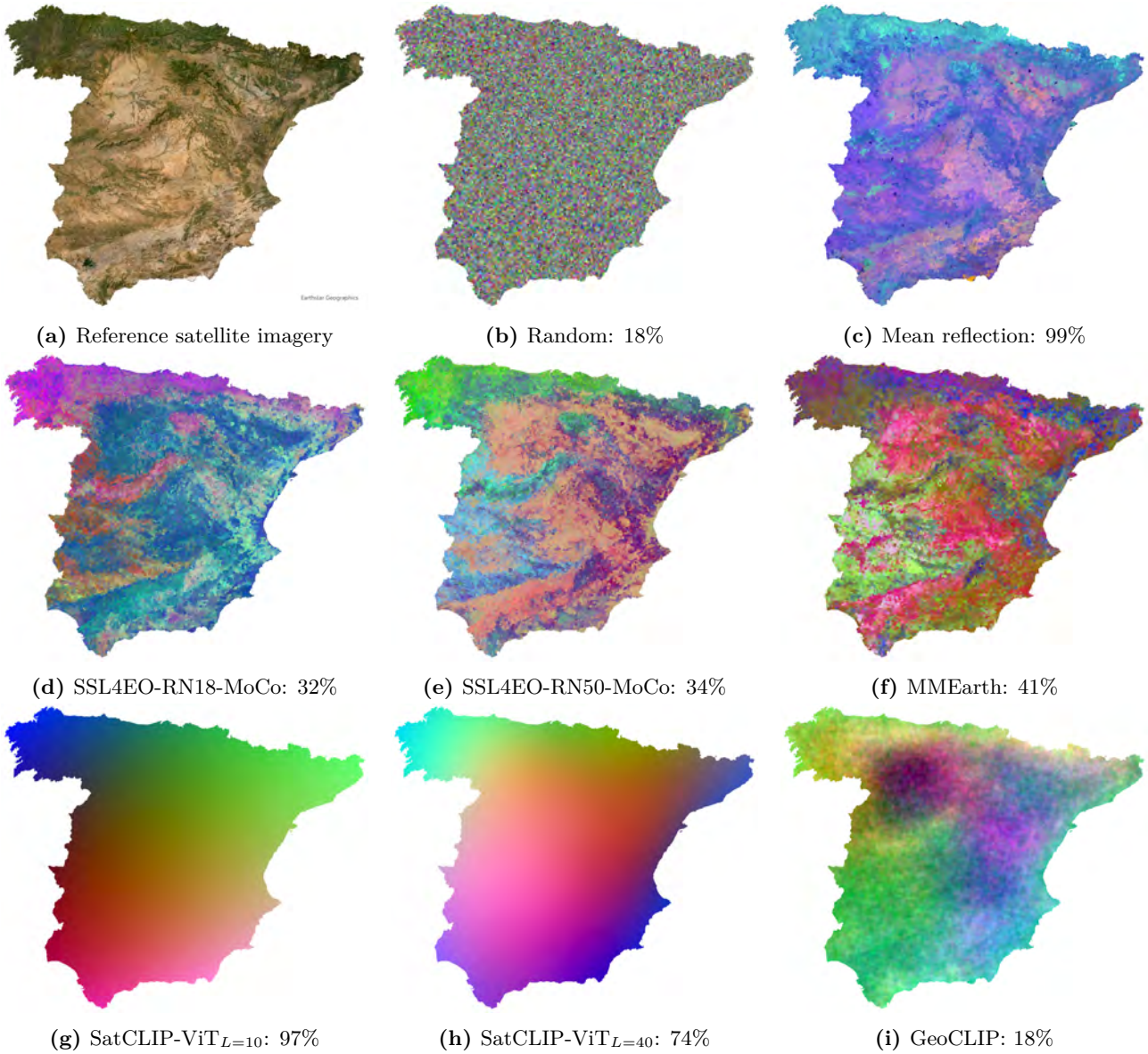



**Figure 4.** Schematic description of a ResNet50 image encoding model. Embeddings were taken after the residual connection (+) from each of the convolutional (conv) layers.

### 4.4. West-East split sampling

To further test each the ability of each encoder to generalise locations, the downstream tasks were repeated on a spatially sampled training set. Instead of sampling the test and training split randomly from $L$, sampling was done purely spatially. The 50% most western samples (by longitude) were used for the training set and the 50% most eastern samples for testing. This creates a spatially clustered sample which is by design no longer representable for the entire dataset. The rest of the experiment followed the downstream task evaluation as mentioned in subsection 4.2.



**(a)** Reference satellite imagery **(b)** Random: 18% **(c)** Mean reflection: 99%

**(d)** SSL4EO-RN18-MoCo: 32% **(e)** SSL4EO-RN50-MoCo: 34% **(f)** MMEarth: 41%

**(g)** SatCLIP-ViT$_{L=10}$: 97% **(h)** SatCLIP-ViT$_{L=40}$: 74% **(i)** GeoCLIP: 18%

**Figure 5.** The first three principal components of an embedding mapped to red, green and blue respectively. Includes the percentage of the total variance explained by these components.

# 5. Results

## 5.1. Embedding pattern visualisation

**PCA maps** In Figure 5 the results of the principal component transformations are mapped to RGB. These are shown along with the percentage of the total variance within each embedding dataset which can be explained by these components. In this figure, map (a) is a reference satellite image. The other maps ((b)-(i)) are the output of the PCA transformation on the embeddings from selected models. In these maps, locations with similar colours have similar embeddings. Maps (b) and (c) visualise the embedding space of the baseline embeddings. As expected, the random model (b) shows noise, indicating no spatial awareness and no ability to distinguish locations meaningfully.

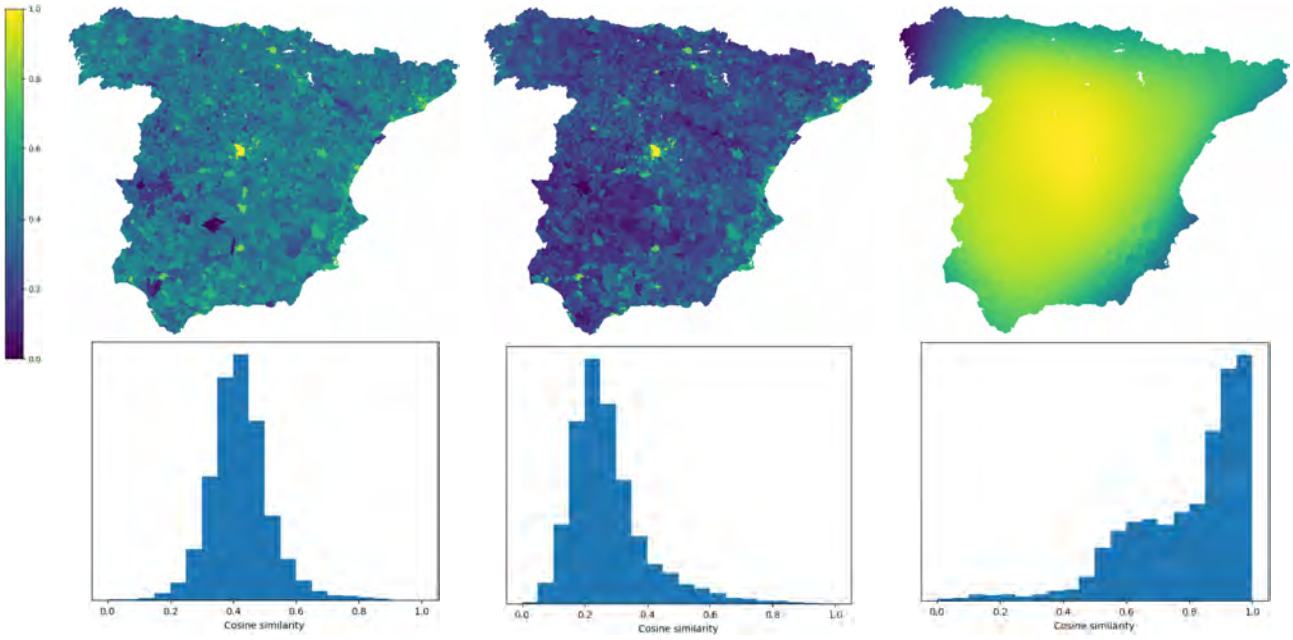
All of the models which were run with the multispectral satellite imagery (c - f) follow the main geographic features of Spain, as can be seen by comparing to the reference satellite image (a). In map for the mean reflection model (c) this generalised pattern is recognizable, but there is little shade variance within the large areas of blue indicating that the model shows low capability to distinguish locations semantically. This compared to the embeddings from the image encoders (d - f), that show more variation in colour. This difference in colours indicates that the image encoder models are more specific in distinguishing certain semantic types of areas. For example, map (c) shows no distinction between the border area with Portugal on the left side of the map and the more land inward area around Barcelona.

The maps (g-i) resulting from the location encoders do not follow the main geographical features of Spain; instead these maps show a colour gradient rather than distinct sections of colour. There was a very high spatial correlation in the location encoder maps, seen by large smooth areas of similar colours. This difference is especially noticeable when compared to the image encoder maps around the mountain area in the south. All image encoder models show a 'line' in the south-west area of Spain, there is no similar line visible in the location encoder maps. There is a clear visual difference between the SatCLIP PCA maps (g, h) and the GeoCLIP map (i). This was due to the distinct strategies these models employ for embedding a location from the location's coordinates. The differences between SatCLIP and GeoCLIP within the context of this thesis will be discussed in more detail in the discussion section.

**Cosine similarity maps** The cosine similarity with respect to Madrid were calculated for the different kinds of models. In Figure 6, these results are shown for three encoder models, all of which had their features pre-trained based on visual transformers. The cosine similarity maps are shown above with matching histograms of the values below. The maps show how similar the embedding of the centroid of the municipality of Madrid is to all municipalities within the $L_{centroid}$ dataset. Cosine similarity is between 0 and 1, with Madrid having a perfect similarity of 1 with itself. The histograms below show the distribution of similarity values across the entire dataset.

The location encoder model from SatCLIP (c) shows very strong spatial similarity, where the embeddings of the geographically closest municipalities also have a high cosine similarity to Madrid. Looking at the histogram, this also results in a high average cosine similarity in the embedding space with a mean cosine similarity of 0.79. This shows that the global SatCLIP model is not able to make highly localised semantic inferences. In other words, as the model was pre-trained for a global covering objective, it's not able to encode different urban areas on the scale of Spain.

The maps and histograms for the image encoders (a-b) show a larger discrepancy between the embed-




**(a)** SSL4EO-ViT-MoCo **(b)** SSL4EO-ViT-DINO **(c)** SatCLIP-ViT$_{L=40}$

**Figure 6.** Cosine similarity of the embedding of Madrid and all other embedded municipalities in $\mathbf{E}_{L_{centroid}}$. For each model, the spatial distribution is mapped out on top and a histogram of the cosine similarity values is shown below.

ding of Madrid and the other municipalities. For both, very few other municipalities are considered close in the embedding space. The municipalities which are the closest are often other urban areas, showing an ability of the image encoders to encode semantic similarity. Take for example the larger Barcelona area, which 'lights up' on these maps. This shows that these pre-trained image encoders encode the the satellite images of Madrid and Barcelona to a similar place in the embedding space, giving them a strong semantic similarity.

Comparing the SSL4EO MoCo model to the DINO model, the average similarity to Madrid differs quite a bit between these models. The MoCo model has the mean average cosine similarity of 0.42 with respect to Madrid, while this is 0.27 for DINO. However, the 99% quantile, which encompasses the 80 most similar municipalities, starts at a comparable value of 0.73 for the MoCo pre-trained model and 0.71 for the DINO pre-trained model. These differences in representation between the MoCo and DINO are further discussed in section 6.1.

## 5.2. Downstream tasks

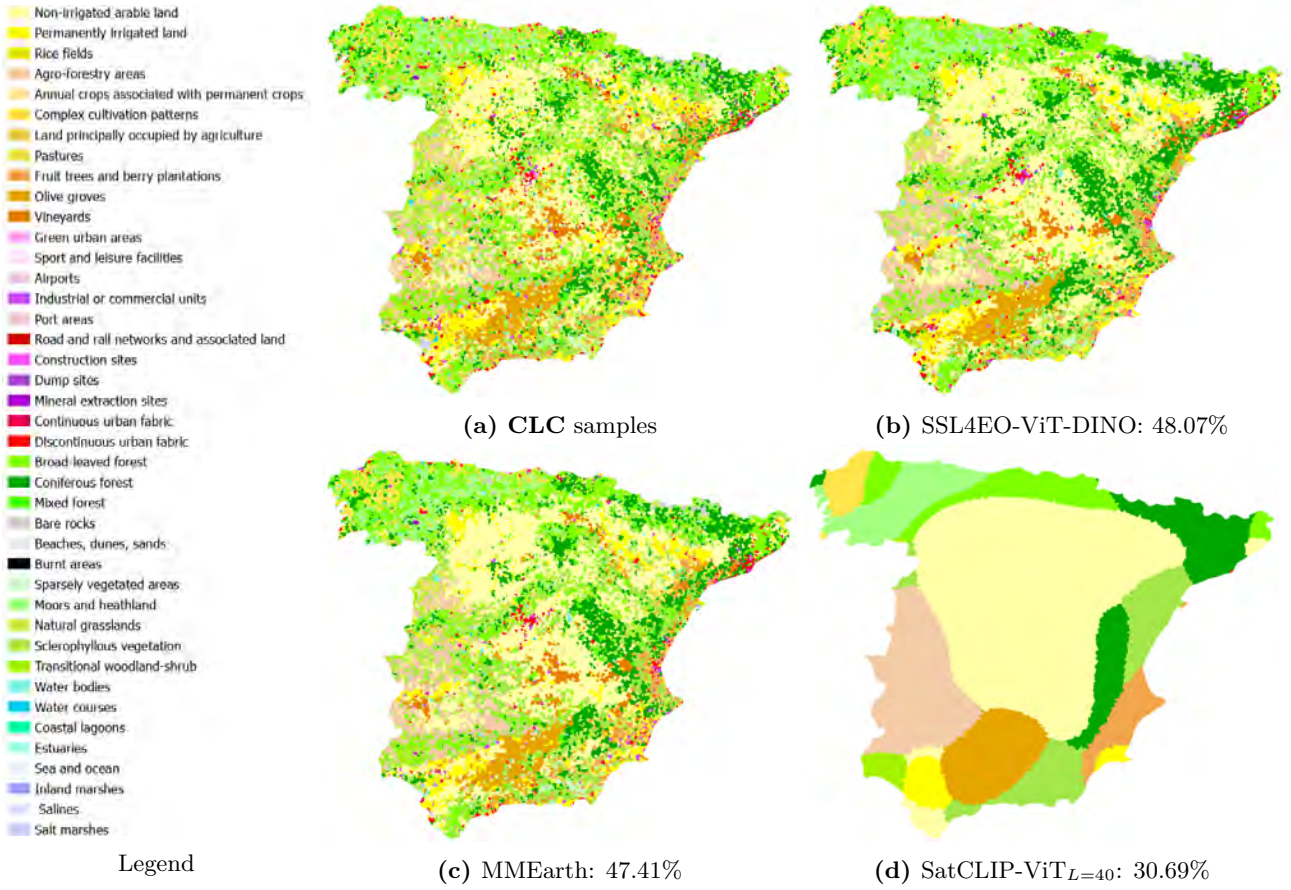
The results for the downstream tasks are in Table 2 for $\mathbf{E}^{\mathrm{M}}_{L_{centroid}}$. The results for the downstream task evaluation on $\mathbf{E}^{\mathrm{M}}_{L_{SRS20k}}$ and $\mathbf{E}^{\mathrm{M}}_{avg\text{-}muni}$ can be found in Appendix D with Table 5 and Table 6 respectively. The evaluation on the disease predictor correlation tasks show that there was no single model which performed best on all of these tasks. Which of the encoding models had the strongest correlation thus depended on the type of disease predictor variable.

The results of the CORINE Land Cover (**CLC**) classification task show that the image encoder models significantly outperformed the location encoder models for all embedding sets. For the embeddings from the centroids of the municipalities as seen in Table 2, there was a difference of at least 10% accuracy between the best image encoder and the worst location encoder. Looking at the prediction maps shown in Figure 7, the SatCLIP location encoder model (d) was unable to model the fine-grained land cover differences. Instead, the fit seems to have predicted the most common land cover type within a larger area. The image encoder models (b-c) predicted a wider variety of different land cover classes and were able to model differences in land cover on small spatial scales.

On the rest of the linear regression tasks, the embedded locations from SatCLIP's models with higher smoothness (L=40) generally had the strongest correlation. In Table 2, an image encoder only finds a higher correlation on one of the 6 other tasks, with the RN50-DINO$_{fc}$ model on the maximum temperature ($\mathbf{T_{max}}$). This model is also notable for showing the strongest correlations among the different image encoder models. This result is consistent across the different embedding sets.

**Table 2.** Downstream task performance for baseline (BL), image and location embeddings on the locations in the $E_{L_{centroid}}$ embeddings. The scores are averaged over 10 independently initialised runs and standard deviation is given. The best performing embeddings per task are highlighted.

| | **Model ↓ Task →** | **CLC** % Accuracy | $\mathbf{T_{min}}$ $R^2$ | $\mathbf{T_{max}}$ $R^2$ | **PM2.5** $R^2$ | $\mathbf{E100k_1}$ $R^2$ | $\mathbf{E100k_2}$ $R^2$ | $\mathbf{E100k_3}$ $R^2$ |
|---|---|---|---|---|---|---|---|---|
| **BL** | Mean reflection | 33.53 ± 0.66 | 0.69 ± 0.01 | 0.71 ± 0.01 | 0.38 ± 0.01 | 0.29 ± 0.01 | 0.22 ± 0.01 | 0.32 ± 0.01 |
| | Random | 24.60 ± 0.38 | 0.00 ± 0.00 | 0.00 ± 0.00 | 0.00 ± 0.00 | 0.00 ± 0.00 | 0.00 ± 0.00 | 0.00 ± 0.00 |
| **Image Enc.** | RN18-MoCo$_{fc}$ | 46.93 ± 0.68 | 0.65 ± 0.01 | 0.77 ± 0.00 | 0.44 ± 0.01 | 0.28 ± 0.01 | 0.13 ± 0.01 | 0.28 ± 0.02 |
| | RN50-DINO$_{fc}$ | 47.50 ± 0.56 | 0.80 ± 0.00 | **0.87** ± 0.00 | 0.54 ± 0.02 | 0.37 ± 0.01 | 0.21 ± 0.02 | 0.37 ± 0.02 |
| | RN50-MoCo$_{fc}$ | 46.21 ± 0.61 | 0.73 ± 0.01 | 0.83 ± 0.00 | 0.49 ± 0.01 | 0.32 ± 0.02 | 0.16 ± 0.02 | 0.33 ± 0.01 |
| | ViT-DINO | 44.83 ± 0.66 | 0.79 ± 0.00 | 0.85 ± 0.00 | 0.49 ± 0.02 | 0.32 ± 0.02 | 0.19 ± 0.01 | 0.33 ± 0.01 |
| | ViT-MoCo | **49.69** ± 0.37 | 0.76 ± 0.01 | 0.81 ± 0.00 | 0.42 ± 0.02 | 0.29 ± 0.01 | 0.15 ± 0.01 | 0.29 ± 0.01 |
| | MMEarth | 44.70 ± 0.69 | 0.71 ± 0.01 | 0.80 ± 0.00 | 0.44 ± 0.02 | 0.33 ± 0.02 | 0.24 ± 0.01 | 0.34 ± 0.01 |
| **Location Enc.** | SatCLIP-RN18$_{L=10}$ | 30.30 ± 0.38 | 0.75 ± 0.22 | 0.73 ± 0.23 | 0.43 ± 0.45 | 0.44 ± 0.05 | 0.35 ± 0.05 | 0.45 ± 0.02 |
| | SatCLIP-RN18$_{L=40}$ | 31.91 ± 0.90 | **0.86** ± 0.02 | 0.84 ± 0.02 | **0.62** ± 0.03 | **0.50** ± 0.01 | 0.39 ± 0.02 | **0.50** ± 0.02 |
| | SatCLIP-RN50$_{L=10}$ | 30.25 ± 0.45 | 0.82 ± 0.06 | 0.80 ± 0.03 | 0.48 ± 0.18 | 0.43 ± 0.05 | 0.34 ± 0.05 | 0.35 ± 0.17 |
| | SatCLIP-RN50$_{L=40}$ | 32.25 ± 0.64 | **0.87** ± 0.01 | 0.84 ± 0.01 | 0.53 ± 0.40 | **0.51** ± 0.01 | 0.40 ± 0.02 | **0.51** ± 0.01 |
| | SatCLIP-ViT$_{L=10}$ | 30.46 ± 0.81 | 0.69 ± 0.16 | 0.77 ± 0.10 | -0.07 ± 1.84 | 0.19 ± 0.65 | 0.18 ± 0.46 | 0.38 ± 0.10 |
| | SatCLIP-ViT$_{L=40}$ | 32.25 ± 0.49 | **0.87** ± 0.00 | 0.84 ± 0.01 | **0.63** ± 0.04 | **0.50** ± 0.03 | **0.42** ± 0.01 | **0.51** ± 0.01 |
| | GeoCLIP | 30.29 ± 0.82 | 0.83 ± 0.00 | 0.82 ± 0.00 | 0.59 ± 0.01 | 0.46 ± 0.01 | 0.34 ± 0.01 | 0.46 ± 0.01 |




Legend

**(a)** **CLC** samples

**(b)** SSL4EO-ViT-DINO: 48.07%

**(c)** MMEarth: 47.41%

**(d)** SatCLIP-ViT$_{L=40}$: 30.69%

**Figure 7.** Prediction maps for the CORINE Land Cover (**CLC**) classification task. Map (a) contains the true classes for the task. The other maps were fit on $\mathbf{E}^{\mathrm{M}}_{SRS20k}$ with logistic regression. Includes the top-1 % accuracy.

## 5.3. ResNet intermediate layer embeddings

The disease predictor tasks were repeated on embeddings from different layers of ResNet models as explained in subsection 4.3. The results of these regression fits with the embeddings on $L_{centroid}$ are in Table 3. For the other embedding sets, the results are in Table 7 and Table 8 in Appendix D. These tables show for each of the SSL4EO models with a ResNet architecture the downstream task performance. Here, not only the best performing embedding per task is highlighted, but also the best performing embedding from different layers in each model.

Across the three tables and for every task, the final fully connected ($fc$) layer was never the best performing for a model. This is quite unexpected, as the established practice for the use of pre-trained models involves including these final-layer embeddings. The results further show that it is depended on the task which convolutional level has the strongest correlation. In general, the land cover classification (**CLC**) shows stronger correlation with the features from convolutional layers 4 and 5 while the linear regression fits have a stronger correlation to the earlier layers (2, 3 and 4). The land cover prediction thus benefits from using high-level features. Similarly, the linear regression is able to correlate better to lower level features. Comparing Table 3 to Table 2 shows that while the embeddings from the intermediate layers outperformed the embeddings from the fully connected, the models from SatCLIP still show a stronger correlation on $\mathbf{T_{min}}$, **PM2.5**, $\mathbf{E100k}_1$, $\mathbf{E100k}_2$ and $\mathbf{E100k}_3$.

**Table 3.** Scores for the downstream tasks on the *centroid* dataset when taking embeddings from deeper parts of a pre-trained ResNet model. Best performing per pre-trained model and task are highlighted. All ResNet models were pre-trained on the SSL4EO [26] dataset.

| **Model ↓ Task →** | **CLC** % Accuracy | **$T_{min}$** $R^2$ | **$T_{max}$** $R^2$ | **PM2.5** $R^2$ | **E100k$_1$** $R^2$ | **E100k$_2$** $R^2$ | **E100k$_3$** $R^2$ |
|---|---|---|---|---|---|---|---|
| RN18-MoCo$_{fc}$ | 46.93 ± 0.68 | 0.65 ± 0.01 | 0.77 ± 0.00 | 0.44 ± 0.01 | 0.28 ± 0.01 | 0.13 ± 0.01 | 0.28 ± 0.02 |
| RN18-MoCo$_5$ | **47.68** ± 0.52 | 0.70 ± 0.01 | 0.80 ± 0.00 | 0.51 ± 0.01 | 0.37 ± 0.01 | 0.26 ± 0.01 | 0.39 ± 0.01 |
| RN18-MoCo$_4$ | 45.70 ± 0.54 | 0.76 ± 0.00 | 0.83 ± 0.00 | 0.52 ± 0.01 | 0.39 ± 0.01 | **0.30** ± 0.01 | **0.42** ± 0.01 |
| RN18-MoCo$_3$ | 41.93 ± 0.49 | **0.79** ± 0.00 | **0.84** ± 0.00 | **0.53** ± 0.01 | **0.40** ± 0.01 | **0.30** ± 0.01 | 0.41 ± 0.01 |
| RN18-MoCo$_2$ | 40.43 ± 0.51 | 0.77 ± 0.00 | 0.83 ± 0.00 | 0.48 ± 0.01 | 0.36 ± 0.01 | 0.27 ± 0.01 | 0.38 ± 0.01 |
| RN18-MoCo$_1$ | 39.33 ± 0.74 | 0.78 ± 0.00 | 0.83 ± 0.01 | 0.50 ± 0.02 | 0.36 ± 0.01 | 0.28 ± 0.01 | 0.39 ± 0.01 |
| RN50-MoCo$_{fc}$ | 46.21 ± 0.61 | 0.73 ± 0.01 | 0.83 ± 0.00 | 0.49 ± 0.01 | 0.32 ± 0.02 | 0.16 ± 0.02 | 0.33 ± 0.01 |
| RN50-MoCo$_5$ | **49.34** ± 0.67 | 0.64 ± 0.01 | 0.77 ± 0.01 | 0.28 ± 0.03 | -0.01 ± 0.03 | -0.28 ± 0.05 | -0.03 ± 0.02 |
| RN50-MoCo$_4$ | 46.93 ± 0.52 | 0.80 ± 0.00 | 0.86 ± 0.00 | 0.51 ± 0.01 | 0.31 ± 0.01 | 0.16 ± 0.02 | 0.32 ± 0.02 |
| RN50-MoCo$_3$ | 44.39 ± 0.75 | **0.83** ± 0.02 | **0.89** ± 0.01 | **0.55** ± 0.03 | 0.38 ± 0.03 | 0.26 ± 0.02 | 0.42 ± 0.02 |
| RN50-MoCo$_2$ | 41.71 ± 0.60 | 0.82 ± 0.01 | 0.87 ± 0.01 | 0.54 ± 0.02 | **0.40** ± 0.05 | **0.30** ± 0.01 | **0.43** ± 0.02 |
| RN50-MoCo$_1$ | 38.40 ± 0.48 | 0.77 ± 0.00 | 0.81 ± 0.00 | 0.50 ± 0.01 | 0.36 ± 0.01 | 0.28 ± 0.01 | 0.38 ± 0.01 |
| RN50-DINO$_{fc}$ | 47.50 ± 0.56 | 0.80 ± 0.00 | 0.87 ± 0.00 | 0.54 ± 0.02 | 0.37 ± 0.01 | 0.21 ± 0.02 | 0.37 ± 0.02 |
| RN50-DINO$_5$ | 47.43 ± 0.59 | 0.74 ± 0.01 | 0.83 ± 0.01 | 0.34 ± 0.02 | 0.04 ± 0.03 | -0.25 ± 0.04 | 0.06 ± 0.03 |
| RN50-DINO$_4$ | **49.05** ± 0.70 | 0.82 ± 0.00 | 0.87 ± 0.00 | 0.54 ± 0.02 | 0.37 ± 0.02 | 0.20 ± 0.01 | 0.38 ± 0.01 |
| RN50-DINO$_3$ | 47.36 ± 0.70 | 0.83 ± 0.00 | **0.88** ± 0.00 | **0.57** ± 0.01 | 0.41 ± 0.01 | 0.27 ± 0.02 | 0.42 ± 0.01 |
| RN50-DINO$_2$ | 44.96 ± 0.50 | **0.84** ± 0.00 | **0.88** ± 0.00 | **0.57** ± 0.02 | **0.43** ± 0.01 | **0.31** ± 0.01 | **0.44** ± 0.01 |
| RN50-DINO$_1$ | 41.09 ± 0.48 | 0.77 ± 0.03 | 0.82 ± 0.01 | 0.52 ± 0.02 | 0.35 ± 0.07 | 0.27 ± 0.05 | 0.40 ± 0.02 |

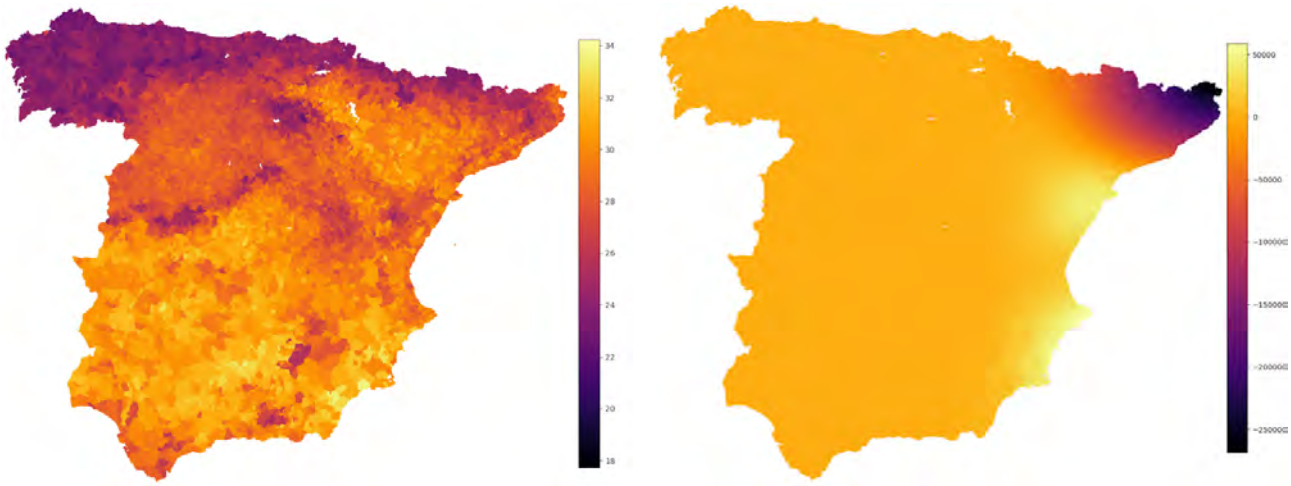
## 5.4. West-East split sampling

The results of the experiment outlined in subsection 4.4 are in Table 4. Since the training and test samples are consistent by design (west-most samples are for training, east-most for testing), no repetition done and scores are shown for a single run. For all models and all tasks, the evaluation scores are lower than with the random sampling. For many of the regressions, the fit to the spatial training data led to a negative correlation ($R^2 < 0$). The only model which was able to fit a positive correlation ($R^2 > 0$) for all tasks is the mean reflection baseline model.

What is mainly striking in Table 4 is the consistent very strong negative correlation ($R^2 \ll 0$) fits with the various SatCLIP models. These models show an extreme extrapolation when subjected to a spatially disjunct training and test sample. For example, the SatCLIP-ViT$_{L=40}$ fit on $\mathbf{T_{max}}$ has an $R^2$ of $-6.33 \times 10^8$, using this spatial split. This $R^2$ is able to be this low as the predicted temperatures are extrapolated.

**Table 4.** Scores for the downstream tasks using east-west train-test split on the *centroid* dataset.

| | **Model ↓ Task →** | **CLC** % Accuracy | **$T_{min}$** $R^2$ | **$T_{max}$** $R^2$ | **PM2.5** $R^2$ | **E100k$_1$** $R^2$ | **E100k$_2$** $R^2$ | **E100k$_3$** $R^2$ |
|---|---|---|---|---|---|---|---|---|
| **BL** | Mean reflection | 27.96 | 0.54 | 0.53 | 0.14 | **0.11** | **0.02** | **0.21** |
| | Random | 20.29 | -0.30 | -0.05 | -0.07 | -0.11 | -0.18 | -0.05 |
| **Image Enc.** | RN18-MoCo$_{fc}$ | 39.07 | 0.36 | 0.46 | 0.12 | -0.11 | -0.25 | 0.00 |
| | RN50-DINO$_{fc}$ | 30.70 | 0.46 | 0.58 | 0.03 | -0.26 | -0.37 | -0.12 |
| | RN50-MoCo$_{fc}$ | 39.25 | 0.50 | 0.61 | 0.04 | 0.01 | -0.13 | 0.10 |
| | ViT-DINO | 29.47 | **0.65** | **0.67** | **0.20** | -0.01 | -0.22 | 0.12 |
| | ViT-MoCo | **40.18** | 0.49 | 0.61 | 0.14 | 0.04 | -0.12 | 0.14 |
| | MMEarth | 29.62 | 0.42 | 0.46 | -0.02 | -0.04 | -0.15 | 0.09 |
| **Location Enc.** | SatCLIP-RN18$_{L=10}$ | 19.41 | $-1.01\times10^{18}$ | $-5.54\times10^{17}$ | $-5.09\times10^{17}$ | $-2.60\times10^{18}$ | $-2.13\times10^{18}$ | $-1.66\times10^{18}$ |
| | SatCLIP-RN18$_{L=40}$ | 9.00 | $-9.03\times10^{8}$ | $-1.11\times10^{10}$ | $-1.25\times10^{10}$ | $-1.97\times10^{9}$ | $-6.36\times10^{9}$ | $-4.88\times10^{8}$ |
| | SatCLIP-RN50$_{L=10}$ | 17.22 | $-3.58\times10^{17}$ | $-5.19\times10^{14}$ | $-3.61\times10^{17}$ | $-9.40\times10^{15}$ | $-2.46\times10^{17}$ | $-8.64\times10^{15}$ |
| | SatCLIP-RN50$_{L=40}$ | 16.34 | $-4.94\times10^{9}$ | $-1.86\times10^{9}$ | $-1.35\times10^{8}$ | $-3.96\times10^{9}$ | $-4.48\times10^{9}$ | $-6.35\times10^{9}$ |
| | SatCLIP-ViT$_{L=10}$ | 19.69 | $-4.31\times10^{17}$ | $-1.09\times10^{17}$ | $-1.73\times10^{17}$ | $-2.69\times10^{18}$ | $-5.31\times10^{18}$ | $-1.50\times10^{18}$ |
| | SatCLIP-ViT$_{L=40}$ | 12.87 | $-6.00\times10^{7}$ | $-6.33\times10^{8}$ | $-4.48\times10^{8}$ | $-3.95\times10^{8}$ | $-2.49\times10^{8}$ | $-4.53\times10^{8}$ |
| | GeoCLIP | 23.61 | 0.58 | 0.47 | 0.12 | -0.24 | -0.34 | -0.14 |



**(a)** SSL4EO ViT-DINO: $R^2 = 0.67$ **(b)** SatCLIP-ViT$_{L=40}$: $R^2 = -6.33 \times 10^8$

**Figure 8.** Prediction maps of $\mathbf{T_{max}}$ in °C for two encoding models. The 50% westernmost samples were used for training and the easternmost were used testing. $R^2$ score for the fit on the test set is given.

The predictions for this example are mapped in Figure 8, with the SatCLIP model in map (b) and an image encoder in map (Figure 8a). The SatCLIP model's predicted maximum temperature values are between $\approx -250,000$ and $\approx 50,000$ °C. These values are far from the actual data range of the samples, and the discrepancy seems to increase if the geometric distance to the training samples increases. Similar extrapolation happens for the linear regression fits of each of the different SatCLIP models. The extrapolation is even stronger for the SatCLIP models with $L = 10$. Following the example, the SatCLIP-ViT$L = 10$ model had an $R^2$ of -1.09×$10^{17}$ on this task instead.

The best performing model for this maximum temperature example, mapped in (Figure 8a), does not have the extrapolation problem, though also suffered in correlation compared to random sampling. The value range of these $\mathbf{T_{max}}$ predictions are between $\approx 18$ and $\approx 34$ °C, which is within expectations. With a reasonable $R^2$ of 0.67, this map shows that this model was still able to correlate the embedding of a location to the maximum temperature even if the training data contained no training samples which were geometrically close.

# 6. Discussion

## 6.1. Embedding pattern visualisation

The results from the embedding pattern visualisation show clear differences between the image encoders and location encoders. These differences were expected and were visualised mainly to show how the embedded features are related in space. The relation between locations can be purely spatial, following Tobler's First law [3], purely semantic or a combination of both. Both the embeddings from the image encoders and the embeddings from the location encoder show a combination of both.

The image encoders embed a similarity which is mostly semantic in nature as the embeddings are just an interpretation of an image. In both the PCA maps and the cosine similarity maps, these embeddings indeed show ability to differentiate between areas based on what's visible from space at that location. These models still show spatial patterns (patches of clustered similar colours) in the PCA maps when visualising the embedding space however, showing that often locations close in space end up close in semantic similarity. This means mainly that the encoders interpret the satellite images of locations which are close geometrically as containing most similar information, which is not unexpected. The fact that image encoders have no inherent spatial awareness still allows locations to be close semantically while far away geometrically. Similarly, it also allows locations which are close geographically to be dissimilar semantically.

The location encoders do have an inherent spatial awareness, as they're encoding based on the main formal identifier of the location in space, the coordinates. The SatCLIP models show this clearly in the resulting PCA and cosine similarity maps which are very smooth. In these maps, the spatial relation is very strong, meaning that locations which are close geometrically are also close semantically. While this shows the location encoders can infer spatial relation with ease, it does not allow for large semantic changes between to geometrically close locations. There is an obvious explanation for this. The SatCLIP models are pre-trained globally and have only had 600 training points on the scale of peninsular Spain.

**Comparing SatCLIP and GeoCLIP** Like SatCLIP, GeoCLIP shows large sections of similar colours in the PCA maps. However, the map (i) shows a sort of speckle, with localised differences among greater patterns in colour. I assume three reasons for this, based on the main differences between SatCLIP and GeoCLIP. Firstly, GeoCLIP uses a hierarchical strategy for the location encoding, which considers features on different levels of spatial resolution. The larger areas of similar colour are thus the result of similar encoding on a high spatial level. The localised differences are then the result of the lower-level encoding. The other difference is in training size. On the scale of Spain, GeoCLIP had significantly more images to pre-train with. This allows for more fine-grained features. Finally, GeoCLIP is pre-trained on terrestrial images to include language features, which would explain why the PCA would not exactly follow the geographical features of the reference satellite imagery (a). This means that compared to all other models in this thesis, GeoCLIP should mainly have an edge when embedding locations with a high-population density.

**Comparing MoCo and DINO** Figure 6 compared the cosine similarity of embeddings to that of Madrid. There is an interesting difference between result for the MoCo [13] pre-trained model and the [15] pre-trained model. This difference in how the histograms look can be explained by the difference in pre-training objective. MoCo uses a form of contrastive learning, meaning the objective is to maximise the semantic distance of the target embedding (in this case Madrid) to all other embeddings in the dataset. In other words, MoCo needs negative samples to identify specific semantic features. DINOs

objective is not contrastive. Instead, it's goal is to learn semantic features from augmented positive images, allowing semantically similar images to cluster naturally.

This difference between embeddings from models pre-trained on MoCo and embeddings pre-trained on DINO can be seen in the histograms of 6a and 6b. The cosine similarity for the MoCo model follows a normal (Gaussian) distribution, centred around a moderate similarity value. This is expected considering the pre-training objective is focussed around having each embedding be as far apart as possible in the embedding space. In contrast, DINO cosine similarity has a lower average similarity which is positively skewed. This positive skew shows that the pre-training was successful in encoding Madrid's most representative features without penalising other municipalities which are close in semantic similarity.

### 6.2. Downstream tasks

**CLC task** The poor result of the SatCLIP model on the CLC task is expected given the patterns explained in the previous section. When using the globally trained location encoder on the relatively small scale of Spain, the encoded semantic features are to coarse to be able to differentiate between areas enough to predict the land cover. Looking at the prediction maps shown in Figure 7, the pre-trained location encoder is unable to predict fine-grained land cover classes and its best fit is found by spatially extrapolating the most common class in a region. This comparison between image and location encoders on a land cover prediction task is interesting, but ultimately unfair. Not only was the CLC dataset created by utilising the Sentinel-2 imagery, which the image encoders have been given to but the location encoders. Furthermore, logistic regression has by design 44 times as many parameters to tune and might therefore inherently work better with the more semantically complex image encoding models. Similarly, as the SatCLIP model was trained on only about 600 images in Spain, the Land Cover task with all 44 classes is too spatially specific for the globally trained SatCLIP even with the higher smoothness.

Another problem is the spatial resolution of the CLC, which is 100m. It raises the question what the location entails. The CLC samples were taken from a 100 by 100 patch at the centre of the Sentinel 2 image, which is 2560 by 2560 meter. This also means that only 100 of the 65536 pixels within the image are certain to contain the sampled land cover class. This could have been considered during sampling, for example by choosing the reference land cover class for a location as the most common CLC class within the image receptive field. The importance of having the relevant class in the receptive field of an image encoder can also be seen by comparing the CLC scores for $\mathbf{E}^f_{L_{centroid}}$ (Table 2) against the scores for $\mathbf{E}^f_{L_{avg\text{-}muni}}$ (Table 6). In the latter case, the location embeddings were averaged over all embeddings within the municipality, which means the CLC patch was never directly, and often not at all in the receptive field.

**Linear tasks** Considering the coarse patterns of SatCLIP's embeddings for these tasks, it was somewhat unexpected it showed stronger correlation for $T_{min}$, PM2.5, and all of the expected disease cases tasks than the image encoders. However, this result might be related to the nature of these tasks. In all of these tasks, the goal is to find correlation between the semantic interpretation of a location and an aspect of a location which can not be detected visually.

Especially for the expected disease case tasks, the image encoder found weaker correlation than the location encoder. The features embedded by the image encoders are never enough to be able to represent each and every aspect of a location which are related to relative disease counts. The image encoders used for this part all encode high-level, complex patterns which might be important for

detecting urban areas but offer too specific semantic representations for explaining relative disease risk. And in the cases where the semantic representation does not suffice, it is important to have some of Tobler's First Law in your model to ensure that '*near things are more related than distance things*' [3].

### 6.3. ResNet intermediate layer embeddings

From the tables which compare embeddings from different layers of a ResNet model (Table 3, 7 and 8), it's clear that using the features of intermediate layers is beneficial for correlating to disease predictors. This is unexpected as taking embeddings from the final fully connected layer, using the entire model, is the de facto standard. However, it is important to note that the matter in which pre-trained convolutional neural networks are used for this thesis, is not the standard. For most use-cases, these pre-trained models are used with fine-tuning or transfer learning. This means that while generally the feature extraction weights are frozen, the models are still trained to a specific task. With these methods, the fully connected layer is usually modified to fit a specific task. But since fully connected layer is practically the same as a regression, fine-tuning with a pre-trained ResNet is comparable to taking the embeddings from the last convolution layer and fitting them separately.

While the result that embeddings from lower-level convolutional layers is somewhat unexpected, previous studies have highlighted the effectiveness of pooling from intermediate ResNet layers [43]. Interestingly, this study also shows that combining pooled features from "various CNN layers is effective in collecting evidences from both low and high level descriptors". Finding out whether combining embeddings from different layers helps with the correlation to disease predictors could be interesting for future research. Another study shows that the deep level features are less depended on pre-training data and that they could provide a general knowledge representation without fine-tuning [44].

It was explained in subsection 4.3 and shown in Figure 4 that the earlier convolutional layers encode low-level features like corners and edges, while the later convolutional layers encode more complex patterns. However, the ResNet models which are used were trained on 13-band multispectral data. This means that along with corners and edges, relevant low-level features can also encode relations between the different bands in the data, akin to traditional remote sensing indices.

The more generalisable semantic image embeddings might be beneficial for the disease mapping. Using embeddings for the fixed effects in the Bayesian disease mapping model scales heavy with embedding output dimension. For this reason, the embedding was planned to be reduced in dimension (PCA) anyway, potentially losing out on relevant semantic features. A recent study has created a selective principal component layer, which incorporates PCA into ResNet Conv block to remove redundant features [45]. This study shows that the intermediate PCA over the layers can improve a model, while doing a PCA at the end usually limits it.

### 6.4. West-East split sampling

This sampling strategy led to a situation where the training set was not representable for the entire dataset. This is proven by looking at the evaluation of the Random embedding on the linear regression tasks. The $R^2$ score for the linear fits of $\mathbf{E}^{\text{Random}}_{SRS20k}$ with randomly selected training samples (Table 2) is consistently 0.00. In Table 4 the correlation of the Random embedding with the predictor variables is consistently negative, showing that the spatial clustering has also led to a semantic clustering. This explains why all of the models perform worse on this task than on the original experiment.

The main goal of this experiment was to fit a relation between semantic features and the disease predictors on one side of the research area and find out how well this relation transfers to the other side. The fact that the image encoders are still able to find a correlation can be attributed to the location independent features which these models encode. In contrast, the embeddings from the location encoders are inherently related to the coordinates. The extrapolation of SatCLIP shows that the features of this model are so strongly correlated to the locations that it over-fits when subjected to the experiment.

The GeoCLIP location encoder does not produce this extrapolation. It's hierarchical location encoder allows for feature embedding on different spatial scales. The reason GeoCLIP was likely able to produce these localised features, whereas SatCLIP struggles is likely due to the higher training sample density in Spain to learn from. The original SatCLIP paper stated that the pre-trained model has '*limited spatial scales, dictated by the L parameter of the location encoder*' [18]. On the scale of Spain, the global model is not suitable for the kinds of spatial comparisons done by this experiment. A SatCLIP model would either require a model to be trained with either more Legendre polynomials and more data or with a different kind of location encoder on a smaller scale.

I believe that while the discrepancy in spatial training sample density can explain why GeoCLIP outperforms SatCLIP on this task, the results might show a limitation of the SatCLIP model. SatCLIP's embeddings are presented as general purpose and globally representable. From this, one could assume that a relation learned in one area of the globe would be transferable to a new area. As far as I can tell, the results show that SatCLIP can have a tendency to over-fit spatially, making it susceptible to extrapolation when predicting a variable in this new area.

## 6.5. Recommendations on encoding locations for disease mapping

A model for the disease mapping objective needs to have a general applicability and be able to be correlated to various disease predictors. Looking at the results, the perfect embedding of a location has spatial awareness while keeping the ability to encode fine-grained differences to geographically close other locations. This is why a combination of a satellite image encoder and a location encoder would be recommended. The most obvious strategy to combine these is to pre-train a Spain specific SatCLIP or other location-image model. This thesis originally planned to do this, but due to time constraints this model was never trained. There are however other ways of combining location and image encoders which would not require any new pre-training.

The embeddings from an image encoder and a location encoder could for example be concatenated. This concatenated embedding should technically encode both location encoder and image encoder features to a single location. As a proof-of-concept, **PM2.5** was fit and predicted with this concatenated embedding with the test setup from subsection 4.2. The prediction maps of the SSL4EO pre-trained image encoder, the SatCLIP encoder and the concatenated embedding is in Appendix E. This concatenated embedding shows a strong correlation to **PM2.5**, with an $R^2$ of 0.85, 0.11 higher than the best performing model in Table 5. Since pre-training new models is computationally costly, it might be interesting to look into this and other strategies for combining the embeddings from existing models in future research.

# 7. Conclusion

To conclude this thesis, the following section revisits the research questions outlined in the introduction and summarizes the key findings.

**What distinguishing visual patterns emerge on maps when comparing spatial embeddings?** The embeddings which have been extracted from locations allowed for comparing a semantic distance along with the geographical distance. The maps have shown that satellite image encoders are able to encode fine-grained feature-rich semantic representations. The embeddings from the location encoders showed high spatial correlation, meaning that locations that are close geographically share semantic similarities. On the scale of Spain the cosine similarity maps showed that the image encoders are suited to detect high semantic similarity even if geographically relatively distant.

**What kinds of embeddings have strongest correlation with selected disease predictor tasks?** Evaluation of the correlation tasks showed that the image encoders were better at predicting land cover classes than the location encoders, which was expected given the complex image features encoded by image encoders. On the other disease predictor tasks the location encoders and specifically the high-smoothness SatCLIP model (L=40) had stronger correlation to these tasks than the image encoders. This was relatively unexpected and accentuated the ability of this location encoder to correlate to disease predictors without the need for high-complexity localised representation.

**What is the effect of taking embeddings of satellite imagery from an intermediate layer of a convolution-based encoder?** As expected, the high-level embeddings from the later convolutional layers of a pre-trained convolutional image encoder were useful for extracting features relating to the classification of land cover classes. However, more interesting is that embeddings from the earlier convolutional layers proved more effective for extracting features which can correlated to the other disease tasks. Specifically, embeddings from final convoluted layers are thus not necessarily most suitable for disease correlation tasks and low-level features might provide representations which are better suited.

**What is the effect of sampling disease prediction variables spatially?** As expected, having a complete east-west split hurts correlation between all embeddings and the disease predictor variables. Still, the image-based features fit to one area could still showed correlation to disease predictors if tested in a spatially separate area. In contrast, the embeddings from SatCLIP's pre-trained models made extremely extrapolated predictions on the test samples, hinting at a potential vulnerability to spatial over-fitting within the model.

In general, pre-trained models allow for extracting semantic information from locations without the need for any costly training. Satellite imagery is becoming increasingly easy to access and the image encoders allow for semantically-rich representation which is able to correlate to various disease predictors. Extracting embeddings of locations from pre-trained location encoders is even easier, as they do not require any resources apart from the coordinates of a location. These embeddings are able to encode spatially aware semantic features, which makes them particularly suited for location representation in existing models.

In the context of disease mapping, the encoding of a location should be able to represent the environmental conditions which affect the relative disease risk. From the results in this thesis, the expectation is that a model which encodes both location as image based features would be most suitable.

## Acknowledgements

I would like to thank Aritz Adin, Carlos Echegoyen and María Dolores Ugarte from the Public University of Navarre for the disease data provided and the help on the disease mapping portion of the thesis. Further thanks go out to Vishal Nedungadi for help with explaining and running the MMEarth model.

I want to also thank the GRS group for the provided amenities. The thesis room provided a good place for focus and nice large screens. My gratitude also goes out to the other students in this room for the fun times and help. A special thanks is for Dainius Masilinas for providing a server to run my experiments remotely.

In addition, I would like to thank Bart Waterreus for his encouragement and for offering to drive me to Wageningen on several occasions. For the times that I did work from home, I want to thank my girlfriend Isolde for supporting me and for listing to my thoughts if I got stressed.

Last and most importantly, I'd like to thank my supervisors Marc and Sytze. Marc has had an inexhaustible enthusiasm for the research and the Deep Learning behind it, and has kept me motivated through the thesis. While Marc and I could get lost in detail, Sytze made sure the whole thesis didn't get too complicated and has kept me sane if I got stuck in my head too much.

## Use of generative AI

Generative AI was used in various stages in the thesis analysis and writing. Automatic completion was used to write faster code documentation. I occasionally asked LLM-based chat models with help writing code, and with coming up with solutions to bugs. While these chat models have given suggestions for the LaTeX formatting of the report, no text was directly written with the help of AI. Some examples of prompts asked to the generative pre-trained large-language models:

- *I'm using LaTeX and have an generated table of contents. I have some loose sections and appendices at the end. How do I reduce the whitespace in the TOC for a section without subsections?*
- *I'm formatting some maps in LaTeX. I have 4 maps and the legend, each a separate image. I would like to have the legend on the left, taking up 0.32 textwidth and the 4 maps on the right 2 by two with each map taking up also 0.32 textwidth. How would I do this?*
- *What does it mean if L2 regularisation is used as penalty?*
- *I'm using ArcGIS. I have a map with predictions. I also have the ground truth for the predictions in a different column in the same shapefile. How do I show the predictions with the lower and upper limit set to the min and max value of the ground truth?*

# References


[1] Sjerp de Vries et al. "In which natural environments are people happiest? Large-scale experience sampling in the Netherlands". In: *Landscape and Urban Planning* 205 (2021), p. 103972. ISSN: 0169-2046. DOI: 10.1016/j.landurbplan.2020.103972.

[2] Ana V. Diez Roux and Christina Mair. "Neighborhoods and health". In: *Annals of the New York Academy of Sciences* 1186.1 (2010), pp. 125–145. DOI: 10.1111/j.1749-6632.2009.05333.x.

[3] W. R. Tobler. "A Computer Movie Simulating Urban Growth in the Detroit Region". In: *Economic Geography* 46 (1970), pp. 234–240. ISSN: 00130095, 19448287.

[4] Donald Shepard. "A two-dimensional interpolation function for irregularly-spaced data". In: *Proceedings of the 1968 23rd ACM National Conference*. ACM '68. New York, NY, USA: Association for Computing Machinery, 1968, pp. 517–524. ISBN: 9781450374866. DOI: 10.1145/800186.810616.

[5] Georges Matheron. "Principles of geostatistics". In: *Economic Geology* 58.8 (Dec. 1963), pp. 1246–1266. ISSN: 0361-0128. DOI: 10.2113/gsecongeo.58.8.1246.

[6] M. Drusch et al. "Sentinel-2: ESA's Optical High-Resolution Mission for GMES Operational Services". In: *Remote Sensing of Environment* 120 (May 2012), pp. 25–36. ISSN: 0034-4257. DOI: 10.1016/j.rse.2011.11.026.

[7] Nathalie Pettorelli et al. "Using the satellite-derived NDVI to assess ecological responses to environmental change". In: *Trends in Ecology & Evolution* 20.9 (2005), pp. 503–510. ISSN: 0169-5347. DOI: 10.1016/j.tree.2005.05.011.

[8] Daniel J Weiss et al. "Air temperature suitability for Plasmodium falciparum malaria transmission in Africa 2000-2012: a high-resolution spatiotemporal prediction". In: *Malaria Journal* 13.1 (May 2014). ISSN: 1475-2875. DOI: 10.1186/1475-2875-13-171.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc., 2012. DOI: 10.1145/3065386.

[10] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[11] Alexander Kolesnikov et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition a Scale". In: *2021 International Conference on Learning Representations (ICLR)*. 2021. arXiv: 2010.11929.

[12] Kaiming He et al. "Masked Autoencoders Are Scalable Vision Learners". In: *CoRR* abs/2111.06377 (2021). arXiv: 2111.06377.

[13] Kaiming He et al. "Momentum Contrast for Unsupervised Visual Representation Learning". In: *CoRR* abs/1911.05722 (2019). arXiv: 1911.05722.

[14] Ting Chen et al. "A simple framework for contrastive learning of visual representations". In: *International conference on machine learning*. PmLR. 2020, pp. 1597–1607.

[15] Mathilde Caron et al. *Emerging Properties in Self-Supervised Vision Transformers*. May 2021. arXiv: 2104.14294.

[16] Gengchen Mai et al. "A Review of Location Encoding for GeoAI: Methods and Applications". en. In: *International Journal of Geographical Information Science* 36.4 (Apr. 2022), pp. 639–673. ISSN: 1365-8816, 1362-3087. arXiv: 2111.04006.

[17] Vicente Vivanco Cepeda, Gaurav Kumar Nayak, and Mubarak Shah. *GeoCLIP: Clip-Inspired Alignment between Locations and Images for Effective Worldwide Geo-localization*. Nov. 2023. arXiv: 2309.16020.

[18] Konstantin Klemmer et al. *SatCLIP: Global, General-Purpose Location Embeddings with Satellite Imagery*. Apr. 2024. arXiv: 2311.17179.

[19] María Dolores Ugarte et al. "On fitting spatio-temporal disease mapping models using approximate Bayesian inference". In: *Statistical Methods in Medical Research* 23.6 (Apr. 2014), pp. 507–530. ISSN: 1477-0334. DOI: 10.1177/0962280214527528.

[20] Erick Orozco-Acosta, Aritz Adin, and María Dolores Ugarte. "Big problems in spatio-temporal disease mapping: Methods and software". In: *Computer Methods and Programs in Biomedicine* 231 (Apr. 2023), p. 107403. ISSN: 0169-2607. DOI: 10.1016/j.cmpb.2023.107403.

[21] Garazi Retegui, Jaione Etxeberria, and María Dolores Ugarte. "Multivariate Bayesian models with flexible shared interactions for analyzing spatio-temporal patterns of rare cancers". en. In: *Environmental and Ecological Statistics* (July 2024). ISSN: 1573-3009. DOI: 10.1007/s10651-024-00630-w.

[22] Aritz Adin et al. "Alleviating confounding in spatio-temporal areal models with an application on crimes against women in India". In: *Statistical Modelling* 23.1 (2023), pp. 9–30. DOI: 10.1177/1471082X211015452.

[23] Xin Guo et al. "Skysense: A multi-modal remote sensing foundation model towards universal interpretation for earth observation imagery". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 27672–27683.

[24] Ashish Vaswani et al. "Attention Is All You Need". In: *CoRR* abs/1706.03762 (2017). arXiv: 1706.03762.

[25] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. *Distilling the Knowledge in a Neural Network*. 2015. arXiv: 1503.02531.

[26] Yi Wang et al. *SSL4EO-S12: A Large-Scale Multi-Modal, Multi-Temporal Dataset for Self-Supervised Learning in Earth Observation*. May 2023. arXiv: 2211.07044.

[27] Sanghyun Woo et al. *ConvNeXt V2: Co-designing and Scaling ConvNets with Masked Autoencoders*. 2023. arXiv: 2301.00808 [cs.CV].

[28] Vishal Nedungadi et al. *MMEarth: Exploring Multi-Modal Pretext Tasks For Geospatial Representation Learning*. arXiv:2405.02771 [cs]. July 2024. DOI: 10.48550/arXiv.2405.02771.

[29] Marc RuSSwurm et al. *Geographic Location Encoding with Spherical Harmonics and Sinusoidal Representation Networks*. Apr. 2024. arXiv: 2310.06743.

[30] Vincent Sitzmann et al. "Implicit Neural Representations with Periodic Activation Functions". In: *CoRR* abs/2006.09661 (2020). arXiv: 2006.09661.

[31] Bojan avri, Tom Patterson, and Bernhard Jenny. "The Equal Earth map projection". In: *International Journal of Geographical Information Science* 33 (Mar. 2019), pp. 454–465. DOI: 10.1080/13658816.2018.1504949.

[32] Matthew Tancik et al. *Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains*. June 2020. arXiv: 2006.10739.

[33] Alec Radford et al. *Learning Transferable Visual Models From Natural Language Supervision*. 2021. arXiv: 2103.00020 [cs.CV].

[34] Martha Larson et al. "The Benchmarking Initiative for Multimedia Evaluation: MediaEval 2016". In: *IEEE MultiMedia* 24.1 (2017), pp. 93–96. DOI: 10.1109/MMUL.2017.9.

[35] Instituto Geográfico Nacional. *Nomenclátor Geográfico de Municipios y Entidades de Población*. Dataset. Provided on September 26 2024 by Aritz Adin from the Universidad Pública de Navarra. 2021. URL: https://www.ign.es/web/rcc-nomenclator-nacional.

[36] Instituto Nacional de Estadística. *Population Figures (Population of Spain's Municipalities. Municipal Register Revision)*. Dataset. 2022. URL: https://www.ine.es/dynt3/metadatos/en/RespuestaDatos.html?oe=30245.

[37] Noel Gorelick et al. "Google Earth Engine: Planetary-scale geospatial analysis for everyone". In: *Remote Sensing of Environment* (2017). DOI: 10.1016/j.rse.2017.06.031.

[38] Gonzalo Vicente et al. "High-dimensional order-free multivariate spatial disease mapping". In: *Statistics and Computing* 33.5 (July 2023). ISSN: 1573-1375. DOI: 10.1007/s11222-023-10263-x.

[39] European Environment Agency. *CORINE Land Cover 2018 (raster 100 m), Europe, 6-yearly - version 2020u1*. en. May 2020. DOI: 10.2909/71c95a07-e296-44fc-b22b-415f42acfdf0.

[40] Rui Yao et al. "Global seamless and high-resolution temperature dataset (GSHTD), 20012020". In: *Remote Sensing of Environment* 286 (2023), p. 113422. ISSN: 0034-4257. DOI: https://doi.org/10.1016/j.rse.2022.113422.

[41] Jing Wei et al. "First close insight into global daily gapless 1km PM2.5 pollution, variability, and health impact". In: *Nature Communications* 14.1 (Dec. 2023). ISSN: 2041-1723. DOI: 10.1038/s41467-023-43862-3.

[42] Adam J. Stewart et al. "TorchGeo: deep learning with geospatial data". In: *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*. SIGSPATIAL 22. ACM, Nov. 2022, pp. 1–12. DOI: 10.1145/3557915.3560953.

[43] Liang Zheng et al. "Good practice in CNN feature transfer". In: *arXiv preprint arXiv:1604.00133* (2016).

[44] Dario Garcia-Gasulla et al. "On the behavior of convolutional nets for feature extraction". In: *Journal of Artificial Intelligence Research* 61 (2018), pp. 563–592.

[45] Jing Liao et al. "A machine learning-based feature extraction method for image classification using ResNet architecture". In: *Digital Signal Processing* 160 (2025), p. 105036. ISSN: 1051-2004. DOI: https://doi.org/10.1016/j.dsp.2025.105036.

[46] Dong C. Liu and Jorge Nocedal. "On the limited memory BFGS method for large scale optimization". In: *Mathematical Programming* 45.1 (Aug. 1989), pp. 503–528. ISSN: 1436-4646. DOI: 10.1007/BF01589116.

# A. Overview of the methodology

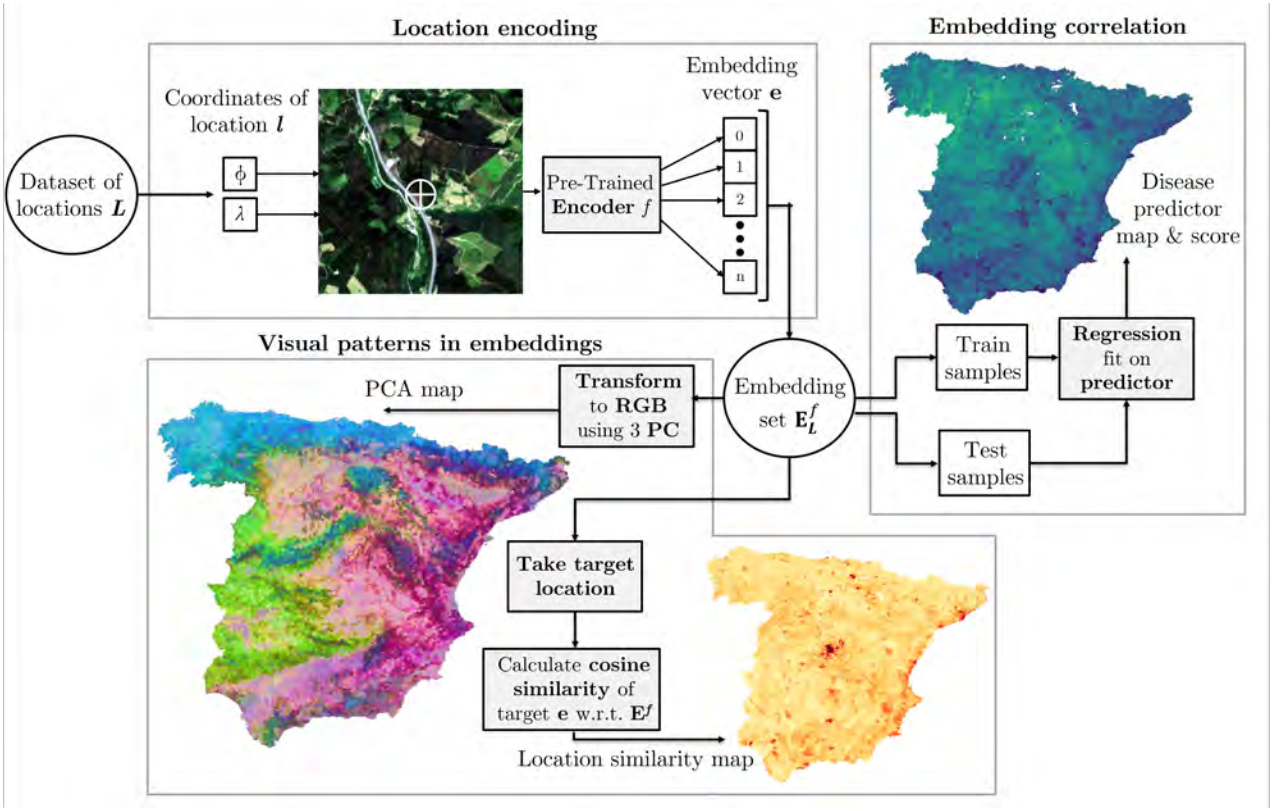



**Figure 9.** An simplified overview of the main data flows in the methodology.

# B. Explanation of the research data

The zip-file provided with this thesis contains the following:

- The thesis report.
- This documentation on file structure.
- Two posters created for the thesis, the midterm poster and the poster for the NWO-NAC.
- The slides for the colloquium.
- A repository folder, which is a modified version of my git repository.

This repository further contains the following:

- A readme file further explaining how to use the repository.
- The required relevant data used in the Data folder.
- A python package called disease_mapping, which contains python modules.
- Jupyter notebooks with scripts and explanations of the methods used.
- An environment file with the dependencies to run some of the notebooks.

## C. Logistic regression

The logistic regression was trained to fit the 44 CORINE Land Cover classes. A location $l_i$ from $L$ has a corresponding land cover class $\boldsymbol{LC}(l_i)$. Logistic regression uses a softmax function to calculate the probability that a location $l_i$ has class $c$ given $\mathbf{e}_i$ as follows:

$$P(\boldsymbol{LC}(l_i) = c \mid \mathbf{e}_i) = \frac{e^{\mathbf{e}_i \cdot \mathbf{w}_c}}{\sum_{j=1}^{C} e^{\mathbf{e}_i \cdot \mathbf{w}_j}} \tag{2}$$

The model is trained with the optimiser algorithm **L-BFGS** [46]. The class-specific weights $\mathbf{w}_c$ are updated by calculating the gradient of the loss function $\mathcal{L}$ with respect to $\mathbf{w}_c$. Logistic regression uses a negative log-likelihood (NLL) as $\mathcal{L}$ defined by:

$$\mathcal{L} = -\sum_{i=1}^{n} \sum_{c=1}^{44} \Delta(\boldsymbol{LC}(l_i) = c) \ln P(\boldsymbol{LC}(l_i) = c \mid \mathbf{e}_i) \tag{3}$$

where $\Delta(\boldsymbol{LC}(l_i) = c)$ is an indicator function that gives 1 if the true land cover class at $l_i$ is $c$ and 0 otherwise. For the test set, the land cover class at $l_i$ is predicted by calculating the class with the highest probability.

$$\hat{\boldsymbol{LC}}(l_i) = \arg\max_{c} P(\boldsymbol{LC}(l_i) = c \mid \mathbf{e}_i) \tag{4}$$

# D. Additional tables

## Downstream task scores on $E_{L_{SRS20k}}$

**Table 5.** Downstream task performance of $E_{L_{SRS20k}}$. The scores are averaged over 10 independently initialised runs and the best performing embeddings per task are highlighted.

| | Model ↓ Task → | Land Cover % Accuracy | $T_{min}$ $R^2$ | $T_{max}$ $R^2$ | PM2.5 $R^2$ |
|---|---|---|---|---|---|
| BL | Mean reflection | 33.71 ± 0.68 | 0.63 ± 0.01 | 0.72 ± 0.01 | 0.41 ± 0.01 |
| | Random | 20.29 ± 0.41 | 0.00 ± 0.00 | 0.00 ± 0.00 | 0.00 ± 0.00 |
| Image Enc. | RN18-MoCo$_{fc}$ | 48.57 ± 0.71 | 0.68 ± 0.01 | 0.83 ± 0.00 | 0.56 ± 0.01 |
| | RN50-DINO$_{fc}$ | 49.50 ± 0.38 | 0.83 ± 0.01 | **0.91** ± 0.00 | 0.67 ± 0.01 |
| | RN50-MoCo$_{fc}$ | 48.52 ± 0.59 | 0.77 ± 0.00 | 0.88 ± 0.00 | 0.60 ± 0.01 |
| | ViT-DINO | **50.45** ± 0.94 | 0.82 ± 0.00 | 0.89 ± 0.00 | 0.60 ± 0.01 |
| | ViT-MoCo | 48.86 ± 0.38 | 0.78 ± 0.00 | 0.87 ± 0.00 | 0.54 ± 0.01 |
| | MMEarth | 48.28 ± 0.50 | 0.70 ± 0.01 | 0.84 ± 0.00 | 0.50 ± 0.01 |
| Location Enc. | SatCLIP-RN18$_{L=10}$ | 26.69 ± 0.49 | 0.85 ± 0.01 | 0.84 ± 0.00 | 0.69 ± 0.06 |
| | SatCLIP-RN18$_{L=40}$ | 29.92 ± 0.52 | **0.86** ± 0.00 | 0.86 ± 0.00 | **0.74** ± 0.01 |
| | SatCLIP-RN50$_{L=10}$ | 26.87 ± 0.47 | 0.84 ± 0.03 | 0.83 ± 0.04 | 0.71 ± 0.02 |
| | SatCLIP-RN50$_{L=40}$ | 29.53 ± 0.44 | **0.86** ± 0.00 | 0.86 ± 0.00 | 0.73 ± 0.01 |
| | SatCLIP-ViT$_{L=10}$ | 26.73 ± 0.58 | 0.85 ± 0.00 | 0.85 ± 0.00 | 0.72 ± 0.01 |
| | SatCLIP-ViT$_{L=40}$ | 29.74 ± 0.69 | **0.86** ± 0.00 | 0.86 ± 0.00 | **0.74** ± 0.01 |
| | GeoCLIP | 32.28 ± 0.87 | 0.82 ± 0.00 | 0.84 ± 0.00 | 0.64 ± 0.01 |

## Downstream task scores on $E_{L_{avg\text{-}muni}}$

**Table 6.** Downstream task performance of $E_{L_{avg\text{-}muni}}$. The scores are averaged over 10 independently initialised runs and the best performing embeddings per task are highlighted.

| | Model ↓ Task → | CLC % Accuracy | $T_{min}$ $R^2$ | $T_{max}$ $R^2$ | PM2.5 $R^2$ | E100k$_1$ $R^2$ | E100k$_2$ $R^2$ | E100k$_3$ $R^2$ |
|---|---|---|---|---|---|---|---|---|
| BL | Mean reflection | 31.49 ± 1.00 | 0.67 ± 0.01 | 0.66 ± 0.01 | 0.37 ± 0.01 | 0.29 ± 0.02 | 0.22 ± 0.01 | 0.32 ± 0.01 |
| | Random | 24.30 ± 0.62 | 0.00 ± 0.00 | 0.00 ± 0.00 | 0.00 ± 0.00 | 0.00 ± 0.00 | 0.00 ± 0.00 | 0.00 ± 0.00 |
| Image Enc. | RN18-MoCo$_{fc}$ | 40.77 ± 1.18 | 0.67 ± 0.01 | 0.77 ± 0.01 | 0.44 ± 0.02 | 0.31 ± 0.02 | 0.19 ± 0.01 | 0.31 ± 0.02 |
| | RN50-DINO$_{fc}$ | **42.65** ± 0.59 | 0.83 ± 0.01 | **0.87** ± 0.00 | 0.57 ± 0.02 | 0.43 ± 0.01 | 0.28 ± 0.02 | 0.42 ± 0.01 |
| | RN50-MoCo$_{fc}$ | 40.57 ± 1.06 | 0.77 ± 0.01 | 0.83 ± 0.00 | 0.50 ± 0.03 | 0.39 ± 0.03 | 0.25 ± 0.01 | 0.38 ± 0.01 |
| | ViT-DINO | 39.25 ± 0.77 | 0.80 ± 0.01 | 0.84 ± 0.01 | 0.52 ± 0.02 | 0.37 ± 0.02 | 0.22 ± 0.01 | 0.39 ± 0.02 |
| | ViT-MoCo | 39.84 ± 1.16 | 0.78 ± 0.01 | 0.82 ± 0.01 | 0.44 ± 0.03 | 0.31 ± 0.02 | 0.21 ± 0.02 | 0.33 ± 0.02 |
| | MMEarth | 38.20 ± 0.96 | 0.73 ± 0.01 | 0.79 ± 0.01 | 0.47 ± 0.02 | 0.38 ± 0.02 | 0.29 ± 0.02 | 0.37 ± 0.02 |
| Location Enc. | SatCLIP-RN18$_{L=10}$ | 30.23 ± 0.71 | 0.85 ± 0.03 | 0.81 ± 0.03 | 0.58 ± 0.18 | 0.48 ± 0.03 | 0.39 ± 0.02 | 0.48 ± 0.05 |
| | SatCLIP-RN18$_{L=40}$ | 31.87 ± 0.59 | **0.88** ± 0.00 | 0.85 ± 0.01 | **0.68** ± 0.02 | 0.52 ± 0.01 | 0.41 ± 0.02 | 0.52 ± 0.02 |
| | SatCLIP-RN50$_{L=10}$ | 30.28 ± 1.00 | 0.84 ± 0.03 | 0.83 ± 0.02 | 0.60 ± 0.10 | 0.49 ± 0.02 | 0.38 ± 0.04 | 0.46 ± 0.05 |
| | SatCLIP-RN50$_{L=40}$ | 32.34 ± 0.82 | **0.88** ± 0.01 | 0.85 ± 0.01 | 0.66 ± 0.03 | 0.52 ± 0.01 | 0.41 ± 0.02 | 0.51 ± 0.02 |
| | SatCLIP-ViT$_{L=10}$ | 30.11 ± 0.67 | 0.84 ± 0.05 | 0.78 ± 0.09 | 0.45 ± 0.30 | 0.47 ± 0.04 | 0.40 ± 0.01 | 0.46 ± 0.07 |
| | SatCLIP-ViT$_{L=40}$ | 32.81 ± 0.48 | 0.87 ± 0.01 | 0.85 ± 0.01 | 0.66 ± 0.02 | **0.53** ± 0.02 | **0.43** ± 0.01 | **0.53** ± 0.01 |
| | GeoCLIP | 32.08 ± 0.77 | 0.84 ± 0.01 | 0.83 ± 0.01 | 0.62 ± 0.03 | 0.50 ± 0.02 | 0.38 ± 0.02 | 0.50 ± 0.02 |

## Downstream task scores on intermediate ResNet embeddings for $L_{SRS20k}$

**Table 7.** Scores for the downstream tasks on the dataset when taking embeddings from deeper parts of a pre-trained ResNet model. Best performing per pre-trained model and task are highlighted.

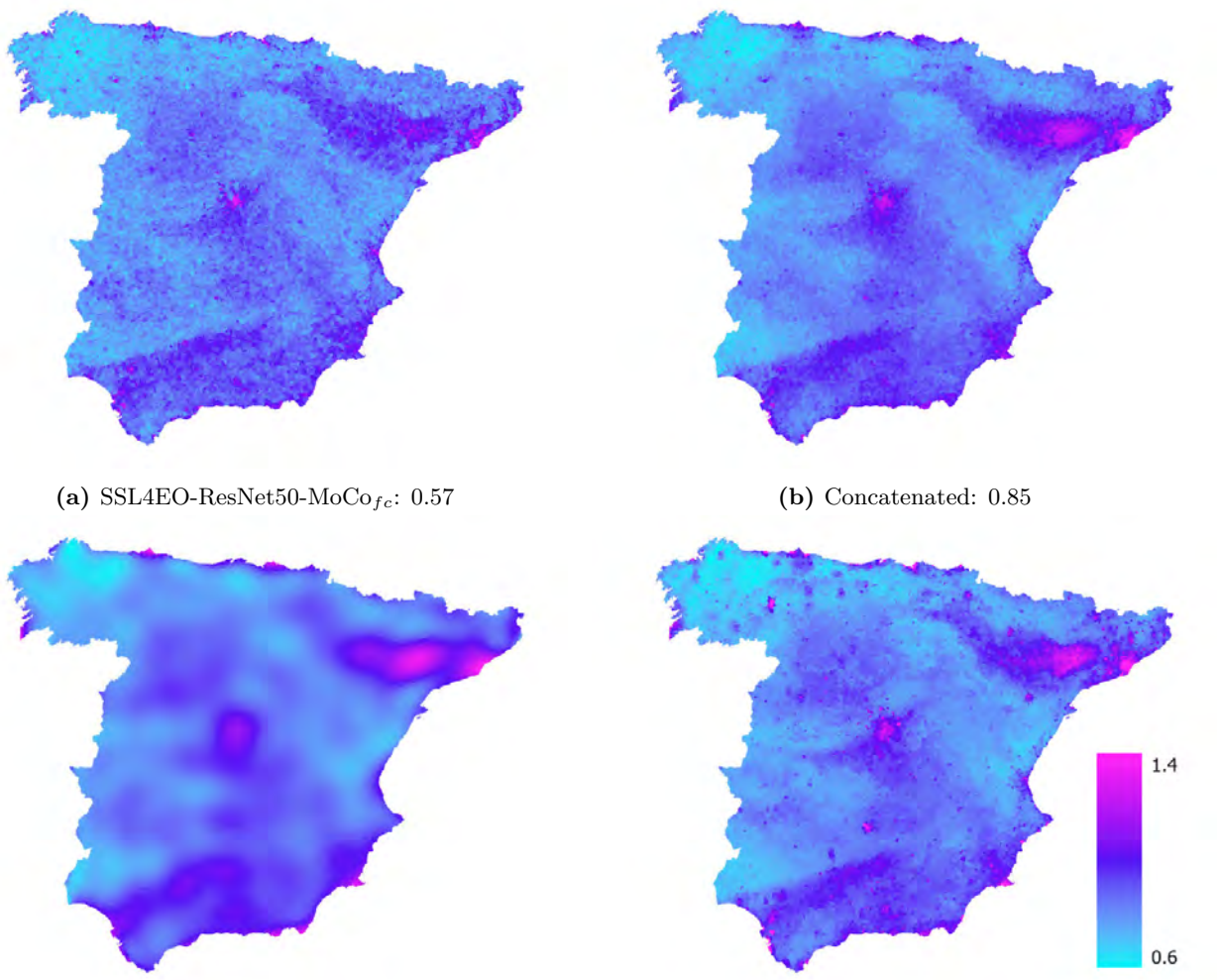
| Model | Land Cover | $T_{min}$ | $T_{max}$ | PM2.5 |
|---|---|---|---|---|
| RN18-MoCo$_{fc}$ | 48.34 ± 0.42 | 0.67 ± 0.00 | 0.82 ± 0.00 | 0.54 ± 0.01 |
| RN18-MoCo$_5$ | **48.98** ± 0.23 | 0.69 ± 0.00 | 0.83 ± 0.00 | **0.56** ± 0.01 |
| RN18-MoCo$_4$ | 47.82 ± 0.33 | 0.74 ± 0.00 | 0.85 ± 0.00 | 0.55 ± 0.01 |
| RN18-MoCo$_3$ | 45.52 ± 0.42 | **0.77** ± 0.00 | **0.86** ± 0.00 | 0.55 ± 0.01 |
| RN18-MoCo$_2$ | 44.12 ± 0.49 | 0.73 ± 0.00 | 0.85 ± 0.00 | 0.52 ± 0.01 |
| RN18-MoCo$_1$ | 42.62 ± 0.46 | 0.73 ± 0.00 | 0.84 ± 0.00 | 0.52 ± 0.01 |
| RN50-MoCo$_{fc}$ | 48.61 ± 0.45 | 0.76 ± 0.00 | 0.87 ± 0.00 | 0.58 ± 0.01 |
| RN50-MoCo$_5$ | **50.69** ± 0.27 | 0.77 ± 0.01 | 0.88 ± 0.00 | 0.57 ± 0.01 |
| RN50-MoCo$_4$ | 49.21 ± 0.37 | 0.83 ± 0.00 | 0.91 ± 0.00 | 0.60 ± 0.01 |
| RN50-MoCo$_3$ | 47.12 ± 0.26 | **0.84** ± 0.00 | **0.92** ± 0.00 | **0.63** ± 0.01 |
| RN50-MoCo$_2$ | 44.55 ± 0.44 | 0.82 ± 0.00 | 0.90 ± 0.00 | 0.60 ± 0.01 |
| RN50-MoCo$_1$ | 41.31 ± 0.32 | 0.72 ± 0.00 | 0.83 ± 0.00 | 0.52 ± 0.01 |
| RN50-DINO$_{fc}$ | 48.74 ± 0.54 | 0.82 ± 0.00 | 0.91 ± 0.00 | 0.64 ± 0.00 |
| RN50-DINO$_5$ | 49.04 ± 0.46 | 0.84 ± 0.00 | 0.91 ± 0.00 | **0.65** ± 0.01 |
| RN50-DINO$_4$ | **50.05** ± 0.62 | **0.85** ± 0.00 | **0.92** ± 0.00 | **0.65** ± 0.01 |
| RN50-DINO$_3$ | 49.38 ± 0.52 | 0.84 ± 0.00 | 0.91 ± 0.00 | 0.61 ± 0.00 |
| RN50-DINO$_2$ | 47.81 ± 0.39 | 0.82 ± 0.00 | 0.90 ± 0.00 | 0.60 ± 0.00 |
| RN50-DINO$_1$ | 43.97 ± 0.43 | 0.74 ± 0.00 | 0.84 ± 0.00 | 0.54 ± 0.00 |

## Downstream task scores on intermediate ResNet embeddings on $E_{L_{avg\text{-}muni}}$

**Table 8.** Scores for the downstream tasks on the $E_{L_{avg\text{-}muni}}$ when taking embeddings from deeper parts of a pre-trained ResNet model. Best performing per pre-trained model and task are highlighted.

| Model | Land Cover | $T_{min}$ | $T_{max}$ | PM2.5 | E100k$_1$ | E100k$_2$ | E100k$_3$ |
|---|---|---|---|---|---|---|---|
| RN18-MoCo$_{fc}$ | 40.13 ± 0.73 | 0.64 ± 0.01 | 0.74 ± 0.01 | 0.38 ± 0.02 | 0.23 ± 0.01 | 0.08 ± 0.02 | 0.23 ± 0.02 |
| RN18-MoCo$_5$ | **40.56** ± 0.34 | 0.70 ± 0.00 | 0.78 ± 0.00 | 0.46 ± 0.02 | 0.36 ± 0.01 | 0.23 ± 0.02 | 0.36 ± 0.01 |
| RN18-MoCo$_4$ | 39.24 ± 0.79 | 0.76 ± 0.00 | 0.81 ± 0.00 | 0.49 ± 0.01 | 0.39 ± 0.01 | 0.29 ± 0.01 | 0.40 ± 0.01 |
| RN18-MoCo$_3$ | 37.85 ± 0.45 | **0.79** ± 0.00 | **0.82** ± 0.00 | **0.50** ± 0.01 | **0.40** ± 0.01 | **0.30** ± 0.01 | **0.42** ± 0.01 |
| RN18-MoCo$_2$ | 36.41 ± 0.48 | 0.76 ± 0.01 | 0.81 ± 0.00 | 0.46 ± 0.01 | 0.37 ± 0.01 | 0.28 ± 0.01 | 0.38 ± 0.01 |
| RN18-MoCo$_1$ | 36.46 ± 0.36 | 0.76 ± 0.00 | 0.80 ± 0.00 | 0.46 ± 0.01 | 0.36 ± 0.01 | 0.28 ± 0.01 | 0.38 ± 0.01 |
| RN50-MoCo$_{fc}$ | 39.16 ± 0.36 | 0.74 ± 0.01 | 0.81 ± 0.00 | 0.43 ± 0.01 | 0.31 ± 0.01 | 0.14 ± 0.03 | 0.32 ± 0.01 |
| RN50-MoCo$_5$ | **41.28** ± 0.58 | 0.64 ± 0.01 | 0.72 ± 0.02 | 0.15 ± 0.05 | -0.09 ± 0.05 | -0.40 ± 0.06 | -0.08 ± 0.03 |
| RN50-MoCo$_4$ | 40.38 ± 0.42 | 0.80 ± 0.00 | 0.84 ± 0.00 | 0.44 ± 0.03 | 0.29 ± 0.01 | 0.13 ± 0.02 | 0.30 ± 0.02 |
| RN50-MoCo$_3$ | 39.43 ± 0.56 | **0.83** ± 0.00 | **0.86** ± 0.00 | **0.54** ± 0.01 | **0.41** ± 0.01 | 0.28 ± 0.02 | **0.42** ± 0.01 |
| RN50-MoCo$_2$ | 37.12 ± 0.71 | 0.82 ± 0.00 | 0.85 ± 0.00 | 0.53 ± 0.02 | **0.41** ± 0.01 | **0.31** ± 0.01 | **0.42** ± 0.01 |
| RN50-MoCo$_1$ | 35.80 ± 0.67 | 0.75 ± 0.00 | 0.78 ± 0.01 | 0.47 ± 0.01 | 0.36 ± 0.01 | 0.28 ± 0.01 | 0.37 ± 0.01 |
| RN50-DINO$_{fc}$ | 37.75 ± 1.05 | 0.80 ± 0.01 | 0.84 ± 0.00 | 0.51 ± 0.01 | 0.36 ± 0.02 | 0.19 ± 0.03 | 0.35 ± 0.02 |
| RN50-DINO$_5$ | 37.78 ± 0.84 | 0.73 ± 0.01 | 0.78 ± 0.01 | 0.29 ± 0.03 | 0.00 ± 0.02 | -0.27 ± 0.02 | 0.00 ± 0.02 |
| RN50-DINO$_4$ | 39.94 ± 0.58 | 0.82 ± 0.00 | 0.85 ± 0.00 | 0.51 ± 0.02 | 0.34 ± 0.02 | 0.19 ± 0.02 | 0.35 ± 0.01 |
| RN50-DINO$_3$ | **41.44** ± 0.51 | **0.83** ± 0.00 | **0.86** ± 0.00 | **0.53** ± 0.01 | 0.41 ± 0.00 | 0.28 ± 0.01 | 0.39 ± 0.01 |
| RN50-DINO$_2$ | 40.37 ± 0.48 | **0.83** ± 0.00 | **0.86** ± 0.00 | **0.53** ± 0.01 | **0.43** ± 0.01 | **0.32** ± 0.01 | **0.43** ± 0.01 |
| RN50-DINO$_1$ | 37.57 ± 0.64 | 0.76 ± 0.00 | 0.80 ± 0.01 | 0.48 ± 0.02 | 0.39 ± 0.01 | 0.29 ± 0.01 | 0.41 ± 0.01 |

# E. Additional maps




**(a)** SSL4EO-ResNet50-MoCo$_{fc}$: 0.57

**(b)** Concatenated: 0.85

**(c)** SatCLIP-ViT$_{L=40}$: 0.72

**(d)** Reference PM2.5

**Figure 10.** Predicting PM2.5 values for $L_{SRS20k}$ with an image encoder (a), a location encoder (c) and with the concatenated embedding of both (b). Predictions include $R^2$. The reference PM2.5 map is in (d).

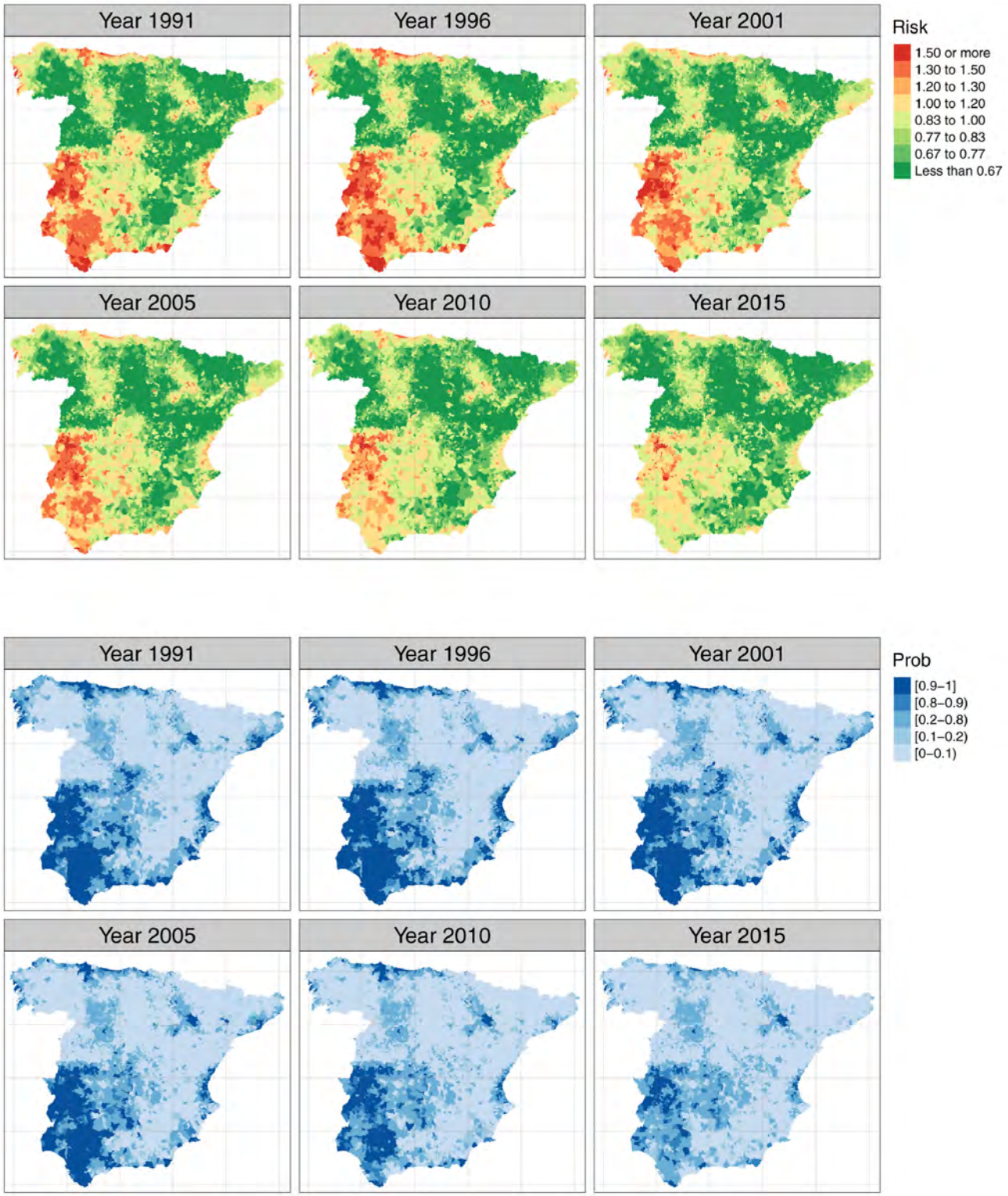
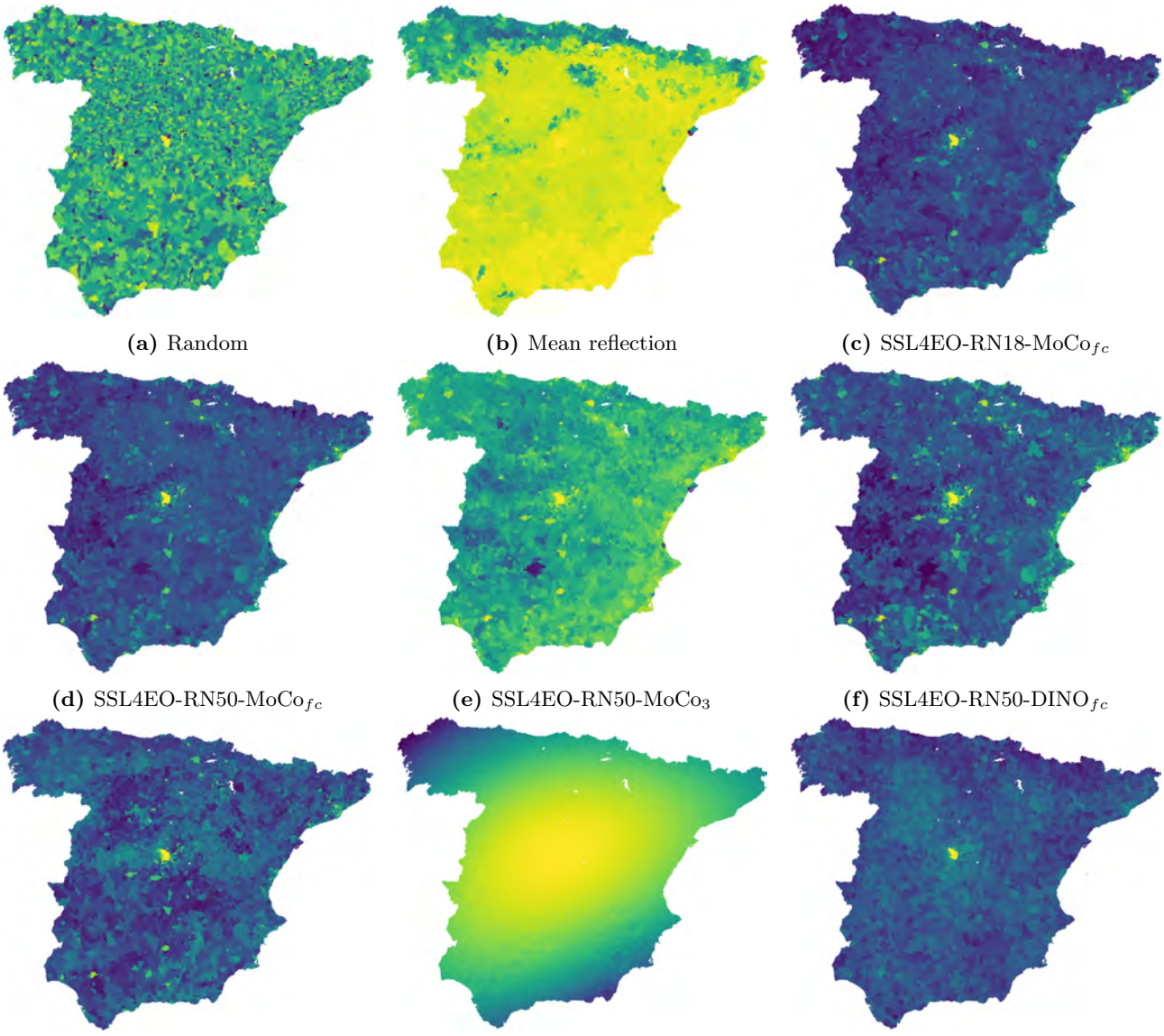



**Figure 11.** Example of disease mapping maps from the 2023 article by Orozco-Acosta et al. [20]. Shows maps of posterior median estimates of relative risks (top) and posterior exceedence probabilities (bottom) for the 1st-order neighbourhood model considering a BYM2 conditional autoregressive prior for space, RW1 prior for time and Type IV interaction for the spatio-temporal effect.



**(a)** Random **(b)** Mean reflection **(c)** SSL4EO-RN18-MoCo$_{fc}$

**(d)** SSL4EO-RN50-MoCo$_{fc}$ **(e)** SSL4EO-RN50-MoCo$_3$ **(f)** SSL4EO-RN50-DINO$_{fc}$

**(g)** MMEarth **(h)** SatCLIP-ViT$_{L=10}$ **(i)** GeoCLIP

**Figure 12.** Cosine similarity of the embedding of Madrid and all other embedded municipalities in $\mathbf{E}_{L_{centroid}}$. Contains models not used in Figure 6